TUM School of Computation, Information and Technology
Technische Universität München

Graph Deep Learning in Medicine - Prospects, Pitfalls, and Privacy

Tamara T. Müller

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Stephan Günnemann

Prüfende der Dissertation:
1. Prof. Dr. Daniel Rückert
2. Prof. Dr. Nassir Navab
3. Prof. Dr. Ben Glocker

Die Dissertation wurde am 25.04.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 28.11.2024 angenommen.

Abstract

Over the last decades, deep learning (DL) techniques have revolutionised data processing in essentially every application domain. In medicine, DL has facilitated clinical workflows and shown highly promising results, for instance, in improving diagnoses, tumour detection, or image enhancement. A variety of DL methods has been developed for different tasks and use cases. Each method is designed to be closely linked to the data it is applied to. Convolutional neural networks (CNNs) have become very powerful DL methods for image data. They use the underlying (Euclidean) geometry of images by using local filters that aggregate information from neighbouring pixels. Graph neural networks (GNNs) are the corresponding counterparts for non-Euclidean data, such as graphs or manifolds. They utilise message-passing schemes that enable information to be propagated along the graph's underlying structure. In medicine, non-Euclidean data can be found in numerous contexts, including knowledge graphs, molecule representations, brain connectivity graphs, surface meshes, or population graphs, where a whole cohort is represented as a network.

In this dissertation, we investigate different aspects of the research question of *when and how to best use graph deep learning for medical applications and research*, specifically highlighting three main findings: (1) The impact of the graph structure on the performance of GNNs can be stronger than expected, for example in settings of medical population graphs. (2) We extend commonly used metrics for graph assessment to a wider range of applications, aligning with medically relevant scenarios such as regression tasks and the utilisation of weighted graphs. (3) We introduce a method for differentially private (DP) training of GNNs for graph-level predictions and find a correlation between the underlying graph structure and the performance of DP GNNs. Differentially private algorithms provide formal privacy guarantees and protect data owners from potential risks of privacy leakage. This is of high importance when utilising sensitive data such as medical records.

Towards a more general understanding of the utility of GNNs in medicine, we on the one hand highlight a major advantage of graph learning by studying a novel application of GNNs for the quantification of fatty tissue in the human body using whole-body meshes, which requires fewer computational resources than conventional methods, such as CNNs. On the other hand, we underline some limitations by showing that even though GNNs have achieved promising results and improved downstream performance, they only outperform graph-agnostic methods under specific circumstances. With this, we raise the need for more appropriate graph construction methods for population graph studies. With the works summarised in this dissertation, we illuminate the understanding of graph deep learning in medicine from a variety of angles, aiming to make a step towards even more effective GNNs and the most suitable application areas.

Zusammenfassung

Die Methoden des Deep Learning (DL) haben die Datenverarbeitung in jeglichen Bereichen unseres Lebens grundlegend verändert. In der Medizin haben Deep-Learning-Methoden zum Beispliel Arbeitsabläufe vereinfacht und vielversprechende Ergebnisse bei der Verbesserung von Diagnosen, Tumorerkennung und Bildverbesserung gezeigt. Für unterschiedliche Anwendungsfälle und Daten wurden jeweils unterschiedliche DL-Methoden entwickelt. Hierbei ist jede Methode eng mit den Daten verknüpft, auf denen sie angewendet wird. Faltungsneuronale Netzwerke (CNNs) sind leistungsstarke DL-Methoden, die darauf ausgelegt sind Bilddaten zu verarbeiten, indem sie lokale Filter verwenden, die Informationen aus benachbarten Pixeln aggregieren. Diese nutzen die zugrundeliegende (euklidische) Geometrie der Bilder. Graphneuronale Netzwerke (GNNs) sind das entsprechenden Gegenstücke für nicht-euklidische Daten wie Graphen oder Mannigfaltigkeiten. Sie nutzen Message-Passing-Systeme, die es ermöglichen, Informationen entlang der Kanten des Graphen weiterzugeben und daher Daten zwischen benachbarten Datenpunkten auszutauschen. In der Medizin können nicht-euklidische Daten in zahlreichen Kontexten verwendet werden, einschließlich knowledge graphs, Molekülrepräsentationen, Gehirnverbindungsgraphen, Oberflächengittern oder Populationsgraphen, bei denen eine ganze Kohorte als Netzwerk dargestellt wird.

In dieser Dissertation untersuchen wir verschiedene Aspekte der Forschungsfrage *wie und wann man am besten Graphneuronale Netze für medizinische Anwendungen verwenden sollte.* Hierbei betonen wir drei Hauptkenntnisse unserer Arbeiten: (1) Der Einfluss der Graphstruktur auf die Ergebnisse von GNNs ist teilweise höher als erwartet, was wir am Anwendungsbeispiel von Populationsgraphen zeigen. (2) Wir erweitern gängige Metriken zur Graphbewertung auf ein breiteres Anwendungsspektrum und ermöglichen somit deren Verwendung für medizinisch relevanten Anwendungsbereicht sowie Regressionsaufgaben und der Verwendung von gewichteten Graphen. (3) Wir stellen eine Methode für differentially private (DP) Training von GNNs für graph-level Vorhersagen vor und erörtern die Korrelation zwischen der zugrunde liegenden Graphstruktur und der Leistung von DP-GNNs. Differentially private Algorithmen bieten formale Datenschutzgarantien und schützen Datenbesitzer vor potenziellen

Risiken der Datenschutzverletzung. Dies ist von besonders hoher Bedeutung bei der Nutzung sensibler Daten wie im medizinischen Bereich.

Um Teile der Frage wie GNNs in der Medizin am besten verwendet werden können herauszuarbeiten, zeigen wir einerseits einen großen Vorteil von GNNs an einer neue Anwendung auf Oberflächennetzen des Körpers, die eine Quantifizierung des Fettgewebes im menschlichen Körper ermöglicht, während sie weniger Ressourcen als andere Methoden wie CNNs erfordert. Andererseits stellen wir einen großen Schwachpunkt von GNNs heraus, indem wir zeigen, dass konventionelle Methoden auf Populationsgraphen gleiche Ergebnisse erzielen wie GNNs und betonen die Notwendigkeit von besseren Graphkonstruktionsmethoden in diesem Kontext. Mit den Arbeiten dieser Dissertation beleuchten wir GNNs von unterschiedlichen Gesichtspunkten und hoffen damit einen Schritt in Richtung noch effizienterer Modelle und Anwendungsfälle zu machen.

Acknowledgements

> *"If I have seen further than others, it is by standing upon the shoulders of giants."*
>
> – Isaac Newton

Given the immense significance of those whose guidance and support have shaped my path, I want to use this opportunity to express my thankfulness to my personal giants. Foremost, I want to express my sincere gratitude to my supervisors and mentors. I want to thank Prof. Daniel Rückert for his guidance, his support and his valuable insights. The scientific freedom that I was granted during these years to follow my interests and passions exceeded everything I could have imagined. Special thanks also go to Georgios Kaissis. Thank you for guiding me throughout my PhD journey, for teaching me all the essentials, skills, and tools I needed for research – and specifically for your exceptional DP guidance. It was a pleasure learning from you. I furthermore want to thank Rickmer Braren for his support and guidance, his discussions on project ideas and medical insights, as well as for giving me the opportunity to pursue this PhD in the first place. Special thanks also go to Pietro Lio, who has been a highly supportive and encouraging mentor. All of you have enriched my PhD with meaningful and engaging discussions, guidance, support, and knowledge. Also, thank you, Deborah, for always taking weight off our shoulders. The lab would not be the same without you.

None of my work would have been possible without the wonderful group of colleagues and co-authors I was pleased to be surrounded by. I cannot name all of you here, but I want you to know that I am more than grateful for all your input and discussions. I learned plenty from all of you. Sophie, you have been a constant support in my work and it has been a great pleasure working with you. Vasiliki, without you, this time would have been significantly less joyful and thank you. It was an immense pleasure working with you and thank you for always having an open ear. Alina, the same goes for you and you are my LaTeX-hero. Thank you, Julian, for always being the supportive sunshine you are. Nil, it has been a pleasure working with you. So many of you have become more than just colleagues over these years. Thank you, Anees, Supro, Chinmay, and Johannes, for your input and passing on valuable graph learning knowledge. Also, special thanks to all medical experts: Fritzi, Johannes, Stephan, and Jessica, without you, many of our results would lack interpretation and meaning.

Throughout my PhD, I was lucky to have worked with bright and highly motivated students. Supervising all of you has been a great pleasure for me. Thank you, Maulik, Jan, Yadu, Siyu, David, Wally, Natalia, Chaima, Tim, Thomas, and Rozi for your collaboration and your efforts.

Finally, I want to thank my family and friends, who have been the best support system one could ask for. MaPa, I am deeply grateful for your unflappable safety and support net. Thank you for making all of this possible by providing me with everything I ever needed. Thank you, Tim, for always believing in me and cheering me on the whole way. Poornima, you have been an indispensable anchor and inspiration. Thank you, Nicki, Svenja, Bárbara, Uendi, Marius, and everyone else who has always been there for me. Thank you, Alex, for accompanying me throughout this journey. You have been a loyal constant at my side and an invaluable source of strength. I am immensely grateful we have met. Thank you for your support, your comfort, and your love.

Contents

List of Figures

Publication List

This dissertation is based on the following peer-reviewed publications. A * indicates shared first authorship.

[1] **T. T. Mueller**, S. Starck, A. Dima, S. Wunderlich, K.-M. Bintsi, K. Zaripova, R. Braren, D. Rueckert, A. Kazi, and G. Kaissis. "A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations." In: *Transactions on Machine Learning Research* (2024).

[2] **T. T. Mueller**, S. Starck, L. F. Feiner, K.-M. Bintsi, D. Rueckert, and G. Kaissis. "Extended Graph Assessment Metrics for Regression and Weighted Graphs." In: *Lecture Notes in Computer Science; Presented at Workshop on Graphs in Biomedical Image Analysis, held in Conjunction with MICCAI 2023, Vancouver, BC, Canada, October 8, 2023*. Springer Nature Switzerland, 2024, pp. 14–26. DOI: 10.1007/978-3-031-55088-1_2.

[3] **T. T. Mueller***, S. Zhou*, S. Starck, F. Jungmann, A. Ziller, O. Aksoy, D. Movchan, R. Braren, G. Kaissis, and D. Rueckert. "Body Fat Estimation from Surface Meshes Using Graph Neural Networks." In: *Lecture Notes in Computer Science; Presented at Shape in Medical Imaging: International Workshop, ShapeMI 2023, Held in Conjunction with MICCAI 2023, Vancouver, BC, Canada, October 8, 2023, Proceedings*. Springer Nature Switzerland, 2023, pp. 105–117. DOI: 10.1007/978-3-031-46914-5_9.

[4] **T. T. Mueller**, S. Starck, K.-M. Bintsi, A. Ziller, R. Braren, G. Kaissis, and D. Rueckert. "Are Population Graphs Really as Powerful as Believed?" In: *Transactions on Machine Learning Research* (2024).

[5] **T. T. Mueller**, D. Usynin, J. C. Paetzold, R. Braren, D. Rueckert, and G. Kaissis. "Differentially Private Guarantees for Analytics and Machine Learning on Graphs: A Survey of Results." In: *Journal of Privacy and Confidentiality* 14.1 (Feb. 2024). DOI: 10.29012/jpc.820.

[6] **T. T. Mueller**, J. C. Paetzold, C. Prabhakar, D. Usynin, D. Rueckert, and G. Kaissis. “Differentially Private Graph Neural Networks for Whole-Graph Classification.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (6 2022), pp. 7308–7318. DOI: 10.1109/TPAMI.2022.3228315.

[7] **T. T. Mueller***, M. Chevli*, A. Daigavane, D. Rueckert, and G. Kaissis. “Differentially Private Graph Neural Networks for Medical Population Graphs and the Impact of the Graph Structure.” In: *IEEE 21st International Symposium on Biomedical Imaging (ISBI)*. Accepted. 2024.

The following additional publications were further written *during the time of the doctoral thesis*. A * indicates shared first or last authorship.

[1] S. Ziegelmayer, G. Kaissis, F. Harder, F. Jungmann, **T. T. Mueller**, M. Makowski, and R. Braren. “Deep Convolutional Neural Network-Assisted Feature Extraction for Diagnostic Discrimination and Feature Visualization in Pancreatic Ductal Adenocarcinoma (PDAC) versus Autoimmune Pancreatitis (AIP).” In: *Journal of Clinical Medicine* 9.12 (2020). DOI: 10.3390/jcm9124013.

[2] G. Dehnen, M. S. Kehl, A. Darcher, **T. T. Mueller**, J. H. Macke, V. Borger, R. Surges, and F. Mormann. “Duplicate Detection of Spike Events: A Relevant Problem in Human Single-Unit Recordings.” In: *Brain Sciences* 11.6 (2021). DOI: 10.3390/brainsci11060761.

[3] **T. T. Mueller**, S. Kolek, F. Jungmann, A. Ziller, D. Usynin, M. Knolle, D. Rueckert, and G. Kaissis. *How Do Input Attributes Impact the Privacy Loss in Differential Privacy?* Accepted at The Fourth AAAI Workshop on Privacy-Preserving Artificial Intelligence. 2022. DOI: https://doi.org/10.48550/arXiv.2211.10173.

[4] A. Sathyanarayanan*, **T. T. Mueller***, M. Ali Moni, K. Schueler, B. T. Baune, P. Lio*, D. Mehta*, et al. “Multi-omics Data Integration Methods and Their Applications in Psychiatric Disorders.” In: *European Neuropsychopharmacology* 69 (2023), pp. 26–46. DOI: https://doi.org/10.1016/j.euroneuro.2023.01.001.

[5] A. Ziller, **T. T. Mueller**, R. Braren, D. Rueckert, and G. Kaissis. “Privacy: An Axiomatic Approach.” In: *Entropy* 24.5 (2022). DOI: 10.3390/e24050714.

[6] S. Starck*, Y. V. Kini*, J. J. M. Ritter, R. Braren, D. Rueckert, and **T. T. Mueller**. "Atlas-Based Interpretable Age Prediction In Whole-Body MR Images." In: *arXiv preprint arXiv:2307.07439* (2023). Accepted at iMIMIC Workshop at MICCAI 2023.

[7] **T. T. Mueller***, M. Chevli*, A. Daigavane, D. Rueckert, and G. Kaissis. "Privacy-Utility Trade-offs in Neural Networks for Medical Population Graphs: Insights from Differential Privacy and Graph Structure." In: *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*. 2023.

[8] K.-M. Bintsi, **T. T. Mueller**, S. Starck, V. Baltatzis, A. Hammers, and D. Rueckert. "A Comparative Study of Population-Graph Construction Methods and Graph Neural Networks for Brain Age Regression." In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology*. Ed. by S.-A. Ahmadi and S. Pereira. Cham: Springer Nature Switzerland, 2024, pp. 64–73.

[9] D. Bani-Harouni, **T. T. Mueller**, D. Rueckert, and G. Kaissis. "Gradient Self-alignment in Private Deep Learning." In: *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2023, pp. 89–97. DOI: 10.1007/978-3-031-47401-9_9.

[10] A. Ziller, A. Güvenir, A. C. Erdur, **T. T. Mueller**, P. Müller, F. Jungmann, J. Brandt, J. Peeken, R. Braren, D. Rueckert, et al. "Explainable 2D Vision Models for 3D Medical Data." In: *arXiv preprint arXiv:2307.06614* (2023).

[11] S. Starck*, V. Sideri-Lampretsa*, J. J. M. Ritter, V. A. Zimmer, R. Braren, **T. T. Mueller***, and D. Rueckert*. "Constructing Population-Specific Atlases from Whole Body MRI: Application to the UKBB." In: *arXiv preprint arXiv:2308.14365* (2023).

[12] A. Ziller, **T. T. Mueller**, S. Stieger, L. Feiner, J. Brandt, R. Braren, D. Rueckert, and G. Kaissis. "Reconciling AI Performance and Data Reconstruction Resilience for Medical Imaging." In: *arXiv preprint arXiv:2312.04590* (2023).

[13] A. Ziller, A. Riess, K. Schwethelm, **T. T. Mueller**, D. Rueckert, and G. Kaissis. "Bounding Reconstruction Attack Success of Adversaries Without Data Priors." In: *arXiv preprint arXiv:2402.12861* (2024).

[14] S. Starck*, V. Sideri-Lampretsa*, B. Kainz, M. Menten, **T. T. Mueller***, and D. Rueckert*. "Diff-Def: Diffusion-Generated Deformation Fields for Conditional Atlases." In: *arXiv preprint arXiv:2403.16776* (2024).

Introduction

Artificial intelligence (AI) has entered, altered, and transformed almost every domain and application area in the last decades – including medicine. Medical research has greatly benefited from the advances of AI by, for example, automating time-intensive tasks, such as semantic segmentations of medical images [1], or supporting the process of medical decision-making [2], treatment planning [3] and monitoring [4]. Neural networks are designed to learn local statistics of a dataset with the goal of generalising them to unseen data samples [5]. This concept has, for instance, been successfully employed with convolutional neural networks (CNNs) [6]–[8], which extract information from medical images. These networks operate on fundamental assumptions about the underlying geometry of the dataset, which is Euclidean [7], [8]. Pixels or voxels in images have fixed positions and all images share the same connectivity in terms of neighbouring pixels/voxels and their relation to each other. Convolutional operations use these properties by extracting local features, which are shared across the whole image domain and therefore extract global information that can be used for downstream tasks. Their efficiency and functionality have made CNNs one of the most powerful AI methods in the last decades.

However, a multitude of datasets is represented more accurately or appropriately when depicted as graph-like structures. Here, data entities can be represented as nodes and edges, which connect pairs of nodes. In contrast to images, graphs create a more flexible structure, where nodes do not necessarily have a fixed position or a constant number of neighbours. This allows for a flexible data representation for various datasets, such as social networks, routing systems, or molecules. A schematic visualisation of an example network or graph is displayed in Figure 1.1. Here, each node represents a person and a connection between two people indicates a relationship between them. In medicine, these networks of subjects, which are similar to social networks, are called population graphs and allow for a coherent representation of a whole patient cohort.

In contrast to images, the underlying space of graphs is non-Euclidean. Such non-Euclidean data does not have the same properties as images such as a common coordinate system or a fixed number of neighbours. Conventional convolutions

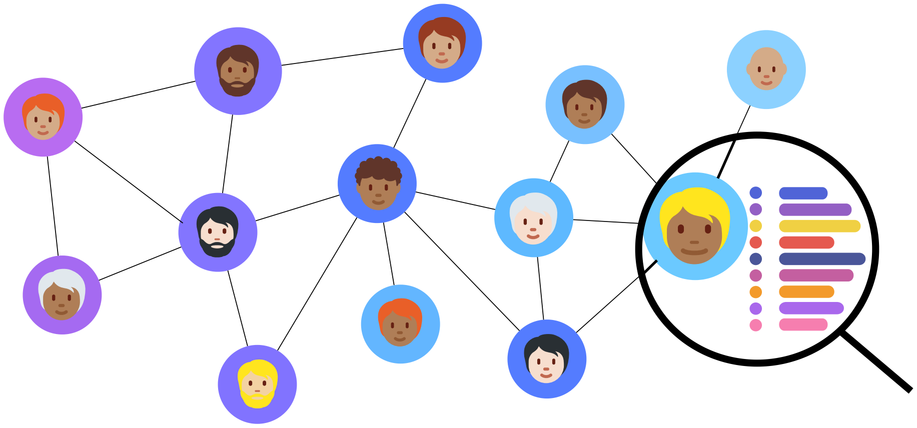


Figure 1.1: Visualisation of a network/graph, which is built by a set of nodes and edges connecting pairs of nodes. Each node can e.g. represent a person in a medical cohort and hold information about their lifestyle or medical records. Furthermore, each node can have a specific label of interest (indicated by node colour), such as a disease or age.

of CNNs are therefore not applicable to graphs. A schematic visualisation of the comparison between the grid-like structure –such as pixels in an image– and a graph without coordinates and with a more flexible structure regarding neighbourhoods and distances (indicated by curved edges) can be found in Figures 2.1a and 2.1b, respectively. Graph-like representations of data allow for high flexibility regarding the number of neighbours of nodes, where some nodes might only have a single neighbour and others multiple thousands. This flexibility comes with the drawback of requiring equally flexible learning techniques that can operate on these datasets. Deep learning (DL) techniques that can be applied to graphs or manifolds are often summarised by the term *geometric deep learning* [5] and neural networks that operate on non-Euclidean graphs, are called graph neural networks (GNNs).

GNNs have been widely applied to several different data structures and in multiple domains, including the studying of social networks [9], learning on manifolds [10], or meshes [11], [12], recommender systems [13], knowledge graphs [14], drug discovery [15], text classification [16], or disease prediction [17], [18]. In general, these graphs can become arbitrarily large and require efficient processing by appropriate deep learning methods. Another challenge in the context of medicine is that the initial data does not always provide an internal graph structure, but the connectivity of different entities needs to be established from the dataset. For example, brain connectivity graphs represent the connectivity of different brain regions. However, both the brain regions and the definition of their connectivity need to be defined and structured in order to allow for the construction of a suitable graph.

This dissertation addresses different applications and challenges of GNNs in the medical domain. For some of these applications, the graph does not come inherent with the dataset at hand but needs to be constructed first. This adds significant complexity to the learning task, which needs to be addressed with caution. We investigate the role of the underlying graph structure in different learning tasks, such as population graphs, where a cohort is represented as one interconnected network. Furthermore, medical research largely implies working with sensitive patient data, such as medical images, records or reports. This raises important privacy concerns as AI methods are vulnerable to information leakage of the data they were trained on [19]–[22]. This can be addressed by implementing privacy-preserving methods, which protect the owners of sensitive medical data. Applying privacy-preserving methods to graphs can raise novel challenges compared to tabular or image data, which need to be addressed carefully.

Objectives In this dissertation, we address parts of the very broad research question of *when and how to best use GNNs for medical applications and research*. In particular, we investigate graph construction methods that transform commonly used medical datasets into graph-like representations and explore the graph structure's impact on the success of GNNs on medical tasks. We compare the utilisation of CNNs and GNNs to investigate their differences and potential advantages. This also includes identifying settings or data representations, where GNNs might not be the ideal methodological choice and formulating fair comparisons to graph-agnostic methods. Finally, we want to discuss privacy concerns that naturally arise when working with sensitive medical data and explore privacy-preserving DL methods for GNNs. GNNs are a highly promising method that allows for the application of deep learning technologies to more diverse and flexible data structures such as graphs. We envision seeing more applications and methods in this direction in future research and hope to contribute to their development for medical applications with this work.

Contributions This dissertation discusses different aspects of graph deep learning in medicine, including different applications of graph learning in the medical domain, highlighting challenges regarding graph construction methods, and privacy-preserving deep learning in the context of graph structures. The main contributions can be summarised as follows:

- For many medical datasets, the graph structure is neither directly provided nor obvious. Here, graph learning techniques require an explicit graph construction step prior to or during training. We summarise and categorise graph construction methods for medical data and formulate recommendations for this essential step of any graph learning pipeline (Chapter 4).

- Graph properties, such as the distribution of node labels, are an important factor for the success of graph learning techniques. However, several such assessment methods can only be evaluated for a subset of problem settings. We, therefore, in Chapter 5, extend two commonly used graph assessment metrics to regression tasks and weighted graphs.

- We implement a novel application of graph deep learning to body surface meshes, which quantifies fatty tissue in the body while reducing resource requirements compared to methods using convolutional neural networks (Chapter 6).

- One application area of GNNs in medicine is in the domain of population graphs. Here a cohort of subjects is represented as a graph that can be used for medical downstream tasks such as disease prediction [17]. In Chapter 7, we show that current state-of-the-art population graphs are not as powerful as believed and do not outperform graph-agnostic baseline models on population graph settings. We highlight the graph construction as the bottleneck and discuss future directions of research in this area.

- Working with medical data naturally raises privacy concerns. Sensitive medical data of individuals is being processed and used for the training of DL models. We discuss applications of differential privacy to graph-structured data in Chapter 8, which comes with additional challenges compared to tabular datasets and images.

- Furthermore, in Chapter 9, we define a method for differentially private deep learning for whole-graph classification tasks and show its privacy-utility trade-offs on several different datasets and under varying privacy guarantees.

- Finally, we discuss the privacy-utility trade-offs for differentially private training on medical population graphs by applying a state-of-the-art method for DP graph learning for node classification to population graph datasets and linking their performance to graph properties like homophily (Chapter 10).

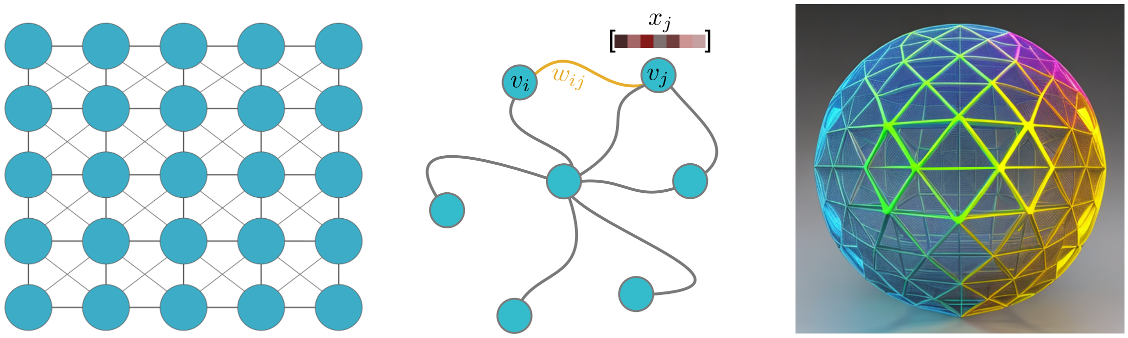
Overview of this Thesis The upcoming chapters are structured as follows: In Chapter 2, relevant background information about deep learning techniques, different data structures in medicine and their interplay with different methods, is provided. We hereby especially focus on graph structures and graph deep learning techniques and highlight their differences to image data and convolutional neural networks. Chapter 3 discusses the basics of privacy-preserving machine learning, specifically focusing on the concept of differential privacy and its application to graph-structured data. Chapter 4 provides an overview of graph construction methods in medicine, their challenges, and recommendations for how to construct graph structures for different applications in medicine and which graph deep learning techniques to use. In several settings, an assessment of a constructed graph structure is of high relevance. In Chapter 5, we formulate extended graph assessment metrics for regression tasks and weighted graphs, which allow for a broader application of two assessment methods to more advanced graph learning settings. Chapter 6 introduces a novel application of GNNs on surface meshes of the whole-body for the quantification of fatty tissue in the human body. We question the utilisation of the current population graph methods for medical downstream tasks by showing their lack of performance improvement compared to graph-agnostic methods in Chapter 7 and highlight that current graph construction methods form the bottleneck for successful graph learning on population graph datasets. Chapter 8 discusses applications of differential privacy on graph-structured data, followed by the introduction of differentially private GNNs for whole-graph classification in Chapter 9. Finally, Chapter 10 shows an application of a differentially private method for node-level predictions on population graphs and discusses the impact of the graph structure on model performance under privacy-preserving machine learning.

Geometric Deep Learning in Medicine

Deep learning (DL) techniques have been highly successful in extracting statistical properties through local statistics. For this, kernels and convolutions have been designed to harness the properties of the data and enable the abstraction of information from the data space. These kernels need to match the underlying data space and profit from using its geometric properties. Medical research and workflows have benefited greatly from DL techniques. Especially the analysis of medical images via segmentation [23], [24], object detection [25], [26], or anomaly detection [27], [28] has facilitated medical tasks and holds great potential to improve and facilitate the discovery of diagnoses or the formulation of treatment plans. The success of DL largely comes down to two aspects: (1) the quality, amount, and versatility of the data it is trained on and (2) the utilisation of appropriate methodologies. Different types of data require different methods and the quality of the training data has a crucial impact on the outcome of the method. Following these two aspects, this chapter first discusses different kinds of medical data and how they can be combined with appropriate DL methods. First, we introduce and compare different data structures, such as medical images, graphs, and triangulated surface meshes. We then link the corresponding DL techniques to them with a specific focus on graph deep learning, its application in medicine, and inherent challenges, such as graph construction and the assessment of graph structures.

2.1 Data Structures in Medicine

Medical data comes in a high variety of formats, ranging from tabular data containing blood values or demographic information, to genetic data, medical images, text of medical reports, or 3D renderings. DL techniques can be applied to the whole variety of different types of data, however, they all require slightly different methodologies. In this section, we discuss some of these structures and their appropriate combination



(a) Schematic display of pixels of an image. An image can be interpreted as a special form of a graph, where only neighbouring pixels are connected and pixels have a fixed position.

(b) Schematic display of a graph with 7 nodes and edges, one edge weight w_{ij} is highlighted in orange and an exemplary feature vector x_j of node v_j is displayed by the red squares.

(c) Schematic visualisation of a surface mesh. The surface of the object (here a sphere) is approximated by triangular faces. A mesh can be interpreted as a special type of graph[2].

Figure 2.1: Visualisation of **(a)** a grid-like image structure, **(b)** a graph with edge weights and node features, and **(c)** a triangulated surface mesh that can be used to model objects in 3D space.

with corresponding DL techniques. We specifically distinguish between data in *Euclidean space* –such as tabular data or images– and *non-Euclidean* data –such as networks or manifolds and discuss their differences and the appropriate methods to perform DL on the different types of data.

2.1.1 Data in Euclidean Space

Euclidean space is a finite-dimensional real vector space $\mathbb{R}^n$, whose properties can be described by the axioms of Euclidean geometry and holds a suitable (Cartesian) coordinate system. The axioms of Euclidean geometry, for example, include that all right angles are equal and that two points in the coordinate system define a line [29]. For more details on Euclidean spaces, we refer to [29], [30].

Data in Euclidean space includes for example 2D and 3D images, tabular data, as well as time series data. When looking through the lens of the underlying space, images can be interpreted as functions on the Euclidean space, which are sampled on a grid (representing the pixels) [5]. A schematic display of the pixels of a 2D image is visualised in Figure 2.1a. Here, the pixels have a fixed position with respect to each

[2] Image generated with *GetImg*: https://getimg.ai/

other (the image would lose its meaning if we randomly shuffle them) and the number of neighbouring pixels is defined and consistent across all images in a dataset. The pixels in the corners of the image have three neighbours each, all other border pixels have five neighbours, and the remaining pixels in the middle of the image have eight neighbours (indicated by the lines connecting pixels). This can be easily transferred to voxels in 3D images. We can say that images are stationary and stable [5]. Other similar data structures in Euclidean space include tabular data or text, where again, the overall shape of the dataset is strictly defined. These properties can be used when designing DL methods on these datasets, which we discuss in more detail in Section 2.2 for image data.

2.1.2 Data in Non-Euclidean Space

When moving away from such fundamental properties as stationarity and stability, more flexible representations can be obtained. One such example are graphs, which consist of a set of nodes and edges, connecting pairs of nodes (Figure 1.1). Two critical properties of graphs that distinguish them from images are that nodes do not have fixed positions and the number of neighbours can vary greatly –even between different nodes of a single graph. Furthermore, two graphs of the same dataset can have a largely different number of nodes or edges. An image can also be regarded as a very specific type of graph, where each pixel represents a node in the graph and both the number of neighbours and the location are fixed (see Figures 2.1a and 2.1b). Another example of non-Euclidean data is a manifold. Manifolds are locally Euclidean spaces and can be approximated via surface meshes, which can be used to discretise the surface of an object.

Formal Definition of Graphs

A graph $\mathcal{G} := \{V, E\}$ is defined as a set of n nodes/vertices V and a set of edges E, connecting pairs of nodes. A schematic display of a graph can be found in Figure 2.1b. All edges E can be summarised in an adjacency matrix $\mathbf{A}$ of size $n \times n$, where $\mathbf{A}_{ij} = 1$ if and only if there exists an edge from node v_i to node v_j, otherwise $\mathbf{A}_{ij} = 0$. Given that the nodes do not have a specific order or indexing, the adjacency matrix can look very different for the same graph, when we re-order the graph's nodes. This embodies the flexibility and yet complexity that graph-structured data carries. Each node can contain node features, usually represented by a vector or a matrix. The node features of node v_i can be represented by the vector x_i and all node features can be combined in a feature matrix $\mathbf{X}$. Furthermore, we can distinguish two types of graphs: *unweighted* and *weighted* graphs. Weighted graphs have an additional weight matrix $\mathbf{W}$ of size $n \times n$, where w_{ij} indicates the edge weight of the edge e_{ij} from node v_i to v_j. Additionally, depending on the graph learning task, a set of labels Y can be defined that either contains a label for the whole graph or a label for each node or edge. A neighbourhood $\mathcal{N}_i$ of node v_i is defined as a set of all nodes that have incoming edges to node v_i. More specifically, one can distinguish between k-hop neighbourhoods, where k defines the number of hops that are required to reach the nodes in the neighbourhood starting from the node of interest. This is displayed in Figure 2.2 and will be referred to later with the concept of graph neural networks.

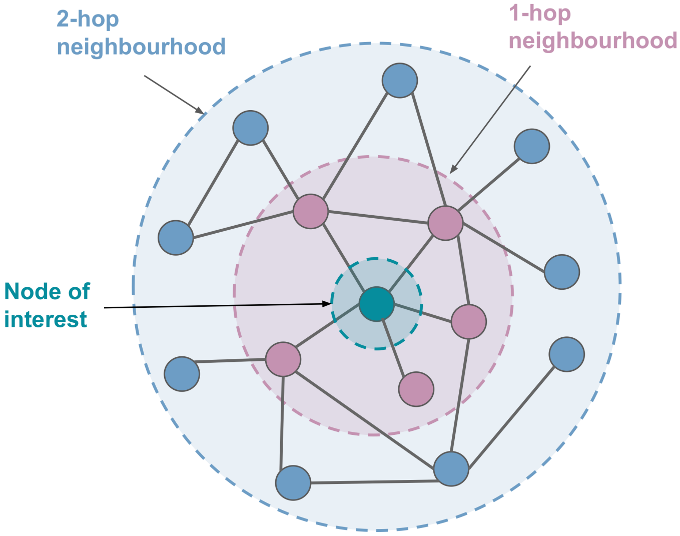


Figure 2.2: Visualisation of graph neighbourhoods. In the middle is a node of interest –also called ego-node–, all connected nodes are direct neighbours and build the 1-hop neighbourhood of this node. All nodes that are connected to nodes of the 1-hop neighbourhood build the 2-hop neighbourhood. They are accessible from the node of interest via two hops/edges.

Triangulated Surface Meshes A specific representation of a graph that we will use in this dissertation, is a 3D surface mesh. Here, each node in the graph represents a point in space that lies on the surface of an object. The node features often summarise the 3D coordinates in space as well as potential additional features of interest, such as intensity values. Neighbouring nodes/coordinates are connected such that they form triangular faces, which approximate the surface of the object. The smaller the faces, the more accurate the representation of the original object. Meshes can be seen as discrete representations of a manifold, which is why they can be processed with geometric deep learning techniques. An example visualisation of a surface mesh can be found in Figure 2.1c. Surface meshes can be used to represent any object and have been used frequently in computer vision research [31], [32]. They can also be useful for medical applications by representing medical objects of interest, such as organs, body shapes, or skin [33]–[35].

2.2 Deep Learning on Images

Convolutional neural networks (CNNs) have been designed for DL tasks on image data and are one of the most successful DL methods in computer vision. They apply a set of local convolutional filters to the input images and common tasks include semantic segmentation [1], image or object classification [36], or object tracking [37]. Their design is based on some fundamental priors about the underlying geometry of images. This, for example, allows for the utilisation of a constant number of parameters for the linear operators at each layer, which makes CNNs highly scalable. CNNs are usually constructed by composing several convolutional and optionally pooling layers, which can be used for dimensionality reduction. Both, convolutional and pooling layers operate on the prior that images hold properties such as locality and stationary and, therefore, leverage the images' underlying geometry. More details on CNNs and their success in image analyses can, for example, be found in [38]–[40].

2.3 Deep Learning on Graphs

Graph deep learning, also termed geometric deep learning [5], refers to the set of deep learning methods that can be applied to and have been designed for non-Euclidean data, such as graphs and manifolds. Neural networks that operate on these data structures are called graph neural networks (GNNs). The idea of GNNs is to use similar concepts as CNNs, such as local filters, but make them applicable to non-Euclidean data. GNNs usually follow a so-called message passing scheme [41]. This refers to the aggregation of information across neighbouring nodes in a graph or graph-like structure. One of the first GNNs that has been applied to a wide range of applications is the graph convolutional network (GCN) [42]. It uses graph convolutions to aggregate the information stored as node features across k-hop neighbourhoods (see

Figure 2.2) and learn new embeddings for the node features. These node embeddings can then be used to make edge-, node-, or graph-level predictions. In the last years, a multitude of graph convolutions have been introduced that use different notions of message passing. In general, a graph convolution can be defined as follows:

Definition 2.3.1 (Graph Convolution)**.** Let x_i be the node features of node v_i, and f a function with learnable parameters that is applied repeatedly for K steps to obtain node feature embeddings. The initial node feature representation of v_i is denoted as $h^0_{v_i} = x_i$. Then the node feature representation of x_i at step k is defined as:

$$h_{v_i}^{(k)} = f\left(h_{v_i}^{(k-1)},\ \mathsf{AGGR}\left(\{h_{v_j}^{k-1} : v_j \in \mathcal{N}_{v_i}\}\right)\right), \tag{2.1}$$

where AGGR denotes an aggregation function and $\mathcal{N}_{v_i}$ the direct neighbourhood of node v_i. Given the fact that the size of the neighbourhoods can vary greatly across the whole graph and that neighbouring nodes do not have an order, this aggregation function needs to be invariant to the number of nodes and their ordering. Examples of suitable aggregation functions are mean, min, or max operations.

2.3.1 Graph Convolutions

A wide variety of graph convolutions have been designed to address different problems and settings and several works have investigated how to make GNNs more robust, flexible, and performant. Some frequently used graph convolutions are GCNs [42], GraphSAGE [43], or graph attention networks (GAT) [44] that all implement slightly different versions of Equation 2.1. It has been shown that different graph convolutions are affected differently by the underlying graph structure. Zhu et al. [45], for example, show that a separation of ego-node features and each k-hop neighbourhood in the message-passing function can have a positive impact on GNN performance under more heterogeneous neighbourhoods. There are graph convolutions specifically designed for mesh datasets [46], or ones that have inherent interpretability via attention mechanisms [44], [47].

Neural Sheaf Diffusion Models Another slightly different graph learning technique are so-called *neural sheaf diffusion models*, introduced by Hansen and Gebhart [48] and extended by Bodnar et al. [49]. Neural sheaf models have been shown to perform better on heterogeneous graph structures (where on average neighbouring labels differ from the ego label) and operate on the topological concept of cellular sheaves. Here, a vector space is assigned to each node and the edges connecting nodes. The objective is to learn linear mappings between the vector spaces of nodes and the adjacent edges. We explore the usage of neural sheaf diffusion models in Chapter 7. Formally a sheaf convolution can be defined as follows:

Definition 2.3.2 (Sheaf Convolution [49])**.** Let $\mathcal{G}$ be a graph with feature matrix $\mathbf{X} \in \mathbb{R}^{nd \times a}$ and $\mathcal{F}$ a sheaf on $\mathcal{G}$, then the sheaf convolution is defined as

$$\mathbf{Y} = \sigma\left(\left(\mathbf{I}_{nd} - \Delta_{\mathcal{F}}\right)\left(\mathbf{I}_n \otimes \mathbf{W}_1\right)\mathbf{X}\mathbf{W}_2\right), \tag{2.2}$$

with $\otimes$ denoting the Kronecker product, $\mathbf{W}_1 \in \mathbf{R}^{d \times d}$ and $\mathbf{W}_2 \in \mathbf{R}^{a \times b}$ two trainable weight matrices, and σ a non-linearity. In this context, a and b refer to the input and output channels of the sheaf convolution and $\mathbf{I}_n$ refers to the identity matrix of shape $n \times n$.

2.3.2 Learning Tasks and Strategies on Graphs

Graph neural networks can be used in several different contexts and for solving different supervised or unsupervised learning tasks as well as for different learning strategies and setups. For this dissertation, we only explore supervised learning tasks for GNNs on medical data and summarise the main differences in application below.

Learning Tasks The main three learning tasks that can be distinguished are **(a)** *node-level predictions*, **(b)** *graph-level predictions*, and **(c)** *edge-level predictions*. Each of them can either target classification or regression tasks. As an example, we will look at molecules as the data of interest, where atoms are represented by nodes and chemical bonds are represented by edges between the atoms. Figure 2.3 visualises such node- and edge-level predictions (predictions indicated by colour). An example of node-level predictions, in this case, would be the prediction of features of atoms in a molecule. Edge-level predictions can be used to denote properties of the chemical bonds, such as different types or tightnesses of chemical bonds. In the context of molecules, graph-level prediction can be used to predict general properties of a whole molecule, such as the presence or absence of a ring. For graph-level predictions, multi-graph datasets are required, that contain several individual graphs.

Learning Strategies For supervised training, we can distinguish two types of learning strategies for graph learning: *inductive* and *transductive* learning. Inductive learning refers to a learning setup similar to classical machine learning, where separate

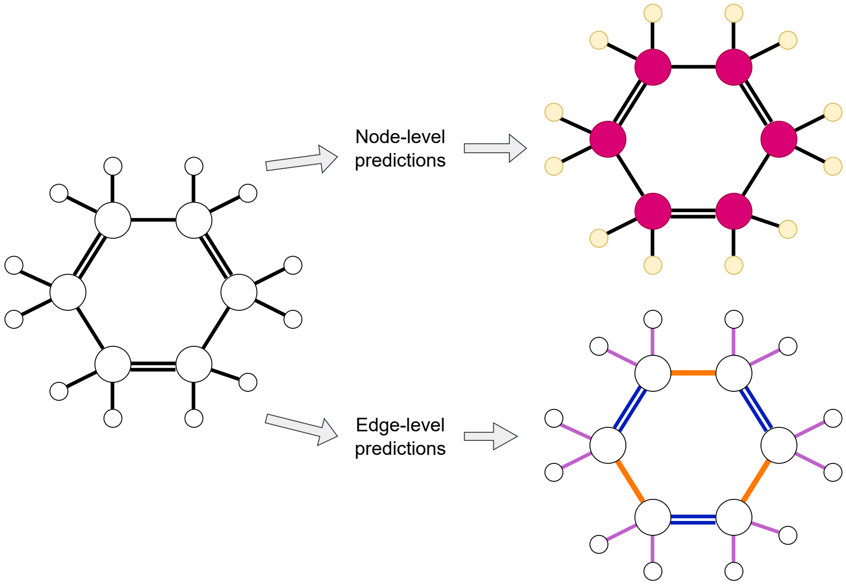


Figure 2.3: Visualisation of node- and edge-level predictions at the example of a molecule where (a) for node-level prediction a label for each node (atom) is predicted (indicated by the colour) and (b) for edge-level prediction, properties of the bindings are predicted.

training and test sets are defined. In the context of graph datasets, two separate graphs are built: a training graph and a test graph. Only the training graph is used for training and the test graph for inference. Transductive learning, on the other hand, only operates on a single graph, utilising all node features at training time but performing backpropagation only using the labels of the training nodes – excluding the test nodes' labels. The latter is specifically designed for graph learning and cannot be applied easily to other DL settings and mostly applies to node- or edge-level predictions.

Static and Adaptive Graph Learning In some works of this dissertation, we distinguish two approaches of graph learning: *static* and *adaptive* (see Figure 2.4). We consider static graph learning approaches as ones that construct a graph prior to learning, while adaptive graph learning settings adapt the graph structure during learning. The latter is specifically interesting when the graph structure is not inherent to the dataset but needs to be constructed from the data. The edges are not fixed

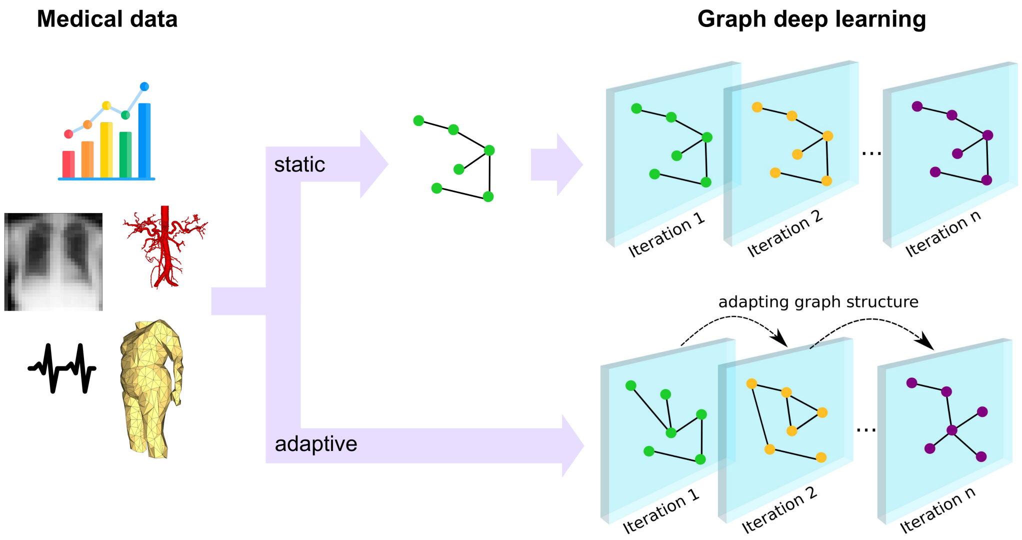
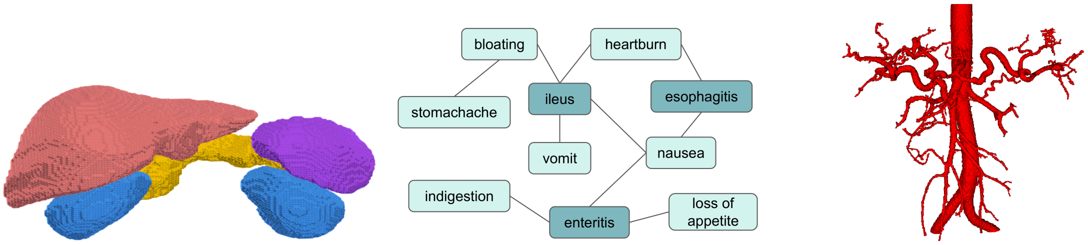


Figure 2.4: Visualisation of static and adaptive graph learning. Static graph learning builds an adjacency matrix prior to training, which is kept static during training. Adaptive graph learning adapts the graph structure during the course of learning. A similar version of this figure was first published in [50].

in these cases and can be defined differently based on the selected method. Here, weighted graphs usually come into play and a continuous adjacency matrix is used, which needs to be differentiable. This adds additional challenges to the learning task and raises interesting research questions such as which method is superior to others and under which circumstances.

2.3.3 Applications of GNNs in Medicine

In the above chapters, we have already briefly discussed some application areas of GNNs in medicine. Multiple types of medical data can benefit from representations in the form of graphs. They can be used to model brain graphs [51]–[53], curvilinear structures such as vessels or airways [54]–[56], molecules [57] for research in drug discovery, or knowledge graphs [58], which can, for instance, contain information about diseases, symptoms, and medication, as well as their interactions. They have also been used in the context of population graph studies [17], where each node represents a subject in a cohort (see Figure 1.1). Figure 2.5 shows three examples of graph-structured data frequently used in medical research. Figure 2.5a visualises triangulated surface meshes of five abdominal organs: liver (red), pancreas (yellow), spleen (purple) and both kidneys (blue). In Figure 2.5b, a schematic visualisation of a knowledge graph is visualised, and Figure 2.5c shows a vessel structure, which can be represented as a graph, connecting branching points of the vessels. In Chapter 4, more graph learning applications in medicine from the perspective of graph construction are discussed. Furthermore, we specifically investigate the usage of surface meshes and population graphs in more detail in Chapters 6, 5, and 7. The following paragraphs introduce the concepts of population graphs and triangulated surface meshes in more detail.



(a) Organ meshes of a liver (red), left and right kidney (blue), spleen (purple) and pancreas (yellow).

(b) A medical knowledge graph, where symptoms and diseases are represented in a graph.

(c) Visualisation of a vessel tree that can be represented as a graph, by using branching points as nodes.

Figure 2.5: Three examples of graph data that is used in medicine. (a) shows surface meshes of organs, **(b)** a medical knowledge graph that represents symptoms and diseases, and **(c)** a vessel tree, where branching points function as nodes and the edges follow the structure of the vessel tubes. Parts of this figure were first published in [50] and [59].

Population Graphs One application area of graph structures, and therefore GNNs in medicine, is population graphs (Figure 1.1). Here, a cohort of subjects is represented in a graph structure. Each node usually represents one subject in the form of a feature vector and "similar" subjects are connected with each other. The concept of population graphs was first introduced by Parisot et al. [17] and has since been extended to several downstream applications, such as disease prediction [60] or age regression [61]. Several works have shown that population graphs out-perform graph-agnostic techniques [17], [60]–[63]. The idea behind the usage of population graphs is medically motivated. It is assumed that subjects which share similar phenotypes (and are therefore connected in the graph) also show similar pathologies. Therefore, the exchange of information between similar subjects is believed to improve the performance of downstream tasks. However, we question this benefit in Chapter 7 and show how easy-to-use graph-agnostic methods, such as random forests or linear regressions, on the tabulated data, actually perform on par with much more complicated GNNs on population graphs.

Surface Meshes in Medicine Another example of applying GNNs to medical data is the usage of surface meshes. They can, for example, be extracted from segmentations of medical images. Figure 2.5a shows triangulated surface meshes of five abdominal organs: liver, kidneys, pancreas, and spleen. Based on the segmentation of an organ, one can extract a triangulated surface mesh of this organ, for instance, by applying the marching cubes algorithm [64]. Organ meshes have, for instance, been used for more accurate organ segmentations [35]. We propose the utilisation of surface meshes in combination with GNNs for the quantification of fatty tissue volume in the human body in Chapter 6.

2.3.4 Assessment of Graph Structures

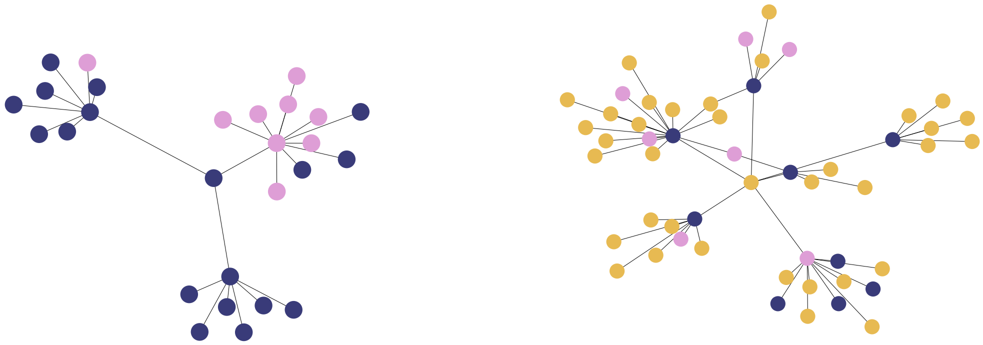
In many cases, it is important to get a better understanding of specific properties of a graph. This can include the average node degree, the density of the graph, or the label distribution across nodes. For that, several graph assessment metrics have been introduced and some of them correlate with the performance of graph neural networks. The most frequently used metric in this context is *homophily*. Homophily describes the average ratio of equally and differently labelled nodes in the 1-hop neighbourhood across all nodes in the graph. In a graph with high homophily, most of the nodes in all 1-hop neighbourhoods share the same label as the node of interest. The opposite of homophily is often called heterophily, indicating that most nodes in a neighbourhood have a different label than the node of interest. An example homophilic and heterophilic graph is displayed in Figures 2.6a and 2.6b, respectively. One can distinguish different notions of homophily: node homophily [65], edge homophily [45], [66], and class homophily [67], [68]. In this work, we utilise node homophily.

Definition 2.3.3 (Node homophily)**.** Let $\mathcal{G} := (V, E)$ be a graph and $Y := \{y_u; u \in V\}$ a set of node labels, where y_i is the label of node v_i. Furthermore, let $\mathcal{N}_{v_i}$ be the set of neighbouring nodes of v_i. Then $\mathcal{G}$ has the following node homophily:

$$h(\mathcal{G}, Y) := \frac{1}{|V|} \sum_{v \in V} \frac{|\{u | u \in \mathcal{N}_v, y_u = y_v\}|}{|\mathcal{N}_v|}, \tag{2.3}$$

where $|\cdot|$ refers to the cardinality of a set.

Homophily can have a strong impact on the performance of GNNs [66]. Most GNNs are designed based on the assumption that neighbouring nodes share similar node features, which are aggregated into new node embeddings during the forward pass. In the case of a low-homophily graph, this can lead to indistinguishable node embeddings, which hinders the performance of accurate predictions. For this reason, several graph convolutions have been introduced to specifically perform well on low-homophily graphs, such as heterogeneous graph transformers [69], H2GCN [45], HEAT convolution [70], or the previously mentioned sheaf diffusion models [48], [49]. Neighbourhoods of heterophilic graphs can still contain high information or be completely uninformative, depending on the distinguishability of nodes with different labels [71]. For example, in a binary classification task, where homophily is very low, the correct class is mostly the opposite class of the neighbouring nodes. Therefore, different notions of homophily, e.g. adjusted homophily [72], as well as additional metrics have been introduced to assess the structure of a graph and indicate different aspects of the correlation between graph structure and GNN performance.



(a) Graph with **high homophily**, most neighbours share the same label as the node of interest.

(b) Graph with **low homophily**, most neighbours have a different label than the node of interest.

Figure 2.6: Visualisation of a graph with **(a)** high homophily and **(b)** low homophily. The node colours indicate node labels. For the graph with high homophily, most neighbours share the same label, whereas the low-homophily graph shows diverse neighbourhoods. Parts of this figure have first been published in [73].

Another commonly used graph assessment metric is called *cross-class neighbourhood similarity* (CCNS) [66] (Definition 2.3.4). This metric moves away from a binary notion of neighbourhood similarities, as utilised by homophily, and quantifies how different the neighbourhoods between all pairs of node classes are. There are several more graph assessment metrics analysing different properties of the graph, such as label information [72], probabilistic Bayes error [71], negative generalised Jeffreys divergence [71], normalised total variation [74], or normalised smoothed value [74], to name only a few.

Definition 2.3.4 (Cross-class neighbourhood similarity)**.** Let $\mathcal{G} = (V, E)$, $\mathcal{N}_{v_i}$, and Y be defined as above. Furthermore, let C be the set of classes that nodes can be labelled with, and V_c the set of all nodes labelled with class c. The cross-class neighbourhood similarity of two classes c and $\hat{c}$ is defined as follows:

$$\mathrm{CCNS}(c, \hat{c}) = \frac{1}{|V_c||V_{\hat{c}}|} \sum_{u \in V_c, v \in V_{\hat{c}}} \mathrm{cossim}(d(u), d(v)), \tag{2.4}$$

where $\mathrm{cossim}(\cdot, \cdot)$ denotes the cosine similarity and $d(u)$ is the histogram of the labels of a node u's neighbours.

The CCNS of a graph is an $|C| \times |C|$ matrix, quantifying the similarity between all combinations of classes. In Chapter 5, we propose a simplified metric that reduces the CCNS matrix to a single value. Furthermore, we identified an essential limitation of most graph assessment metrics: they are limited to classification tasks and unweighted graphs. We, therefore, extend the two metrics, homophily and cross-class neighbourhood similarity, to weighted graphs and homophily to regression tasks. This is essential in order to cover a wider range of applications in medical research and make the assessment of graph structures more generally applicable.

2.3.5 Interpretablity of GNNs

Despite the huge success of DL methods, there are some strong points of criticism. One of them is that most DL models are "back box" methods that do not provide any insights into the decision-making process. This can have negative implications and affect the trustworthiness of DL techniques. This is especially critical in medical settings, where both doctors and patients need to trust these technologies and where potentially life-altering decisions could be based on recommendations of DL systems. Therefore, interpretability methods have been developed which allow one to shed light on certain aspects of how a DL model reaches a decision. Several different approaches have been developed that allow for an investigation of specific aspects of the decision-making process of DL models [75], [76]. One can, for example, distinguish between ad-hoc and post-hoc methods. The former has an inherent interpretability method integrated into the model, whereas the latter uses an additional method after training to explain the model's decisions. Furthermore, one can distinguish between perturbation-, surrogate-, gradient-, and reinforcement-based techniques. For CNNs, interpretability techniques such as Grad-CAM [77] and their variants and improvements have been used frequently, also in the domain of medical image analysis [78], [79]. Grad-CAM is a post-hoc interpretability method, that investigates the impact of input pixels/voxels in images using the model gradients. Ad-hoc methods, such as attention-based approaches allow for an immediate interpretability of the model, without requiring an extra step. For GNNs, graph attention models (GAT) [44] have, for instance, been designed with the same principle. They learn attention weights from neighbours, adapting the message-passing scheme to match the neighbourhood at hand.

In Chapter 9, we use a method called `GNNExplainer` [80], which is a perturbation-based post-hoc method that aims to identify a sub-graph $\mathcal{G}_S \subseteq \mathcal{G}$ of the input graph $\mathcal{G}$ that is considered as important for the decision-making process of the GNN model. A feature selector $F \in \{0,1\}^d$, where d denotes the number of node features, is used to identify important node features. The importance is measured by maximising the mutual information between the predicted label Y of the full model and the model with the selected sub-graph $\mathcal{G}_S$ and the feature selector F. This allows one to investigate which parts of the graph have contributed most to the decision and therefore reason about the consistency or whether a decision is medically sensible. We investigate the interpretability of GNNs in combination with privacy-preserving graph learning pipelines.

Privacy-Preserving Deep Learning

An important concern of deep learning (DL) methods is that neural networks store information about the data they are trained on, which can be leaked and raises critical privacy concerns [81]. This is especially critical in domains like medicine, where sensitive patient data is being processed. DL models are vulnerable to attacks that can retrieve information about the data they were trained on from a model and its gradients. This includes successfully distinguishing whether a specific subject was part of the training data (membership inference attack [82]) or the successful reconstructions of medical images (data reconstruction attack [19]). Both scenarios can have a severe impact on the person whose data or presence in a dataset has been identified. For example, if an adversary –who wants to retrieve information about training samples from a DL model– can determine the presence of a person's data in an oncological study dataset, they can conclude beyond reasonable doubt that this person has a tumour. This information is considered private and should not be extracted from DL models. In the case of reconstruction attacks, highly sensitive information such as the presence of medical pathologies can be inferred from a reconstructed medical image. This evokes a big challenge in maintaining the trust of patients to share their data, which is essential for gathering large datasets for successful applications of deep learning. Therefore, privacy-preserving data analysis needs to be implemented in order to protect the owners of sensitive medical data and allow for the utilisation of DL methods with all their major benefits.

Anonymisation or pseudonymisation does not hold reasonable protection from data leakage, since it is possible to match "anonymised" data points with non-anonymised databases and therefore retrieve the identity of subjects from the first [83]–[85]. DL models that are trained on sensitive data are sensitive data in themselves. The training data can be reconstructed from a trained machine learning model and its gradients [19], [21], [86]–[89], which can represent a critical privacy breach for every subject that donates their medical data for research or medical practices. Therefore, more sophisticated methods are required to preserve the privacy of individuals. The gold standard for privacy-preserving deep learning is *differential privacy* (DP) [83], which provides formal privacy guarantees.

3.1 Differential Privacy

Differential privacy (DP) [83] is a theoretical framework that allows analysts to draw conclusions from datasets while protecting the privacy of the data owners. Intuitively speaking, a DP algorithm yields approximately the same results, independent of whether a single individual's data is present or absent in the training dataset. DP is a property of the data release mechanism and holds independently of the availability or absence of additional information –like for example linking data points to other databases. DP is the only method that provides formal and mathematically provable privacy guarantees. Furthermore, it is agnostic to post-processing and subsequent computations on the private results. It is therefore future-proof: no potential additional future information or post-processing will be able to undermine the privacy guarantees provided by a DP mechanism.

Definition 3.1.1 ((ε, δ)-DP)**.** Let D be a database, which contains sensitive data. Furthermore, let D' be a neighbouring dataset which differs from D by one record. We denote neighbouring datasets by $D \simeq D'$. A randomised mechanism $\mathcal{M}$ is (ε, δ)-differentially private if for all subsets S of Range($\mathcal{M}$) and all pairs of neighbouring datasets D and D', the following relation holds:

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S) + \delta. \tag{3.1}$$

This definition is symmetric –it also holds if D and D' are swapped– and ε is also called *privacy budget*, where a low ε indicates high privacy guarantees and a high ε low guarantees.

One strong limitation of AI in medicine compared to other domains is the scarcity of medical data. AI models are reliant on large datasets, which are difficult to obtain in medical settings - also due to justified privacy concerns. We believe that with a stronger application of DP in medical research, the collection of larger datasets could be more realistic. Data owners might be more willing to donate their data for research, if they know that their privacy is being formally protected, immune to post-processing, and with clearly quantified risks. We discuss the utility of privacy budgets and define an axiomatic approach to privacy based on quantifiable information flows in [90].

Despite the strong advantages of using DP in order to provide formal privacy guarantees, there are two main drawbacks when using DP DL techniques: (a) DP algorithms require more resources for training and (b) the performance of DL models suffers under the usage of DP. The latter is referred to as the "privacy-utility trade-off". In the medical domain, this raises an important ethical question of balancing the protection of data owners and highly accurate medical algorithms that could potentially save lives or strongly impact people's health. This trade-off is especially important in domains like medicine, where both complementary goals are highly desirable and have ethical implications.

3.2 Differentially Private Deep Learning

The framework of DP is applicable to all mechanisms and has been applied frequently to DL techniques. The most commonly used approach for DP training of neural networks is differentially private stochastic gradient descent (DP-SGD), introduced by Abadi et al. [91]. DP-SGD uses the Gaussian Mechanism to privatise per-sample gradients before the update of model parameters by (a) clipping the L_2-norm of each per-sample gradient and (b) adding calibrated noise to it. We note, that this method can be applied to all first-order optimisation techniques and is not limited to SGD.

Definition 3.2.1 (Global L_2-Sensitivity)**.** Let $f : X \to Y$ be a function, X and Y metric spaces, d_X a distance metric associated with X, d_Y the L_2 metric in Y. Furthermore, let D and D' be two neighbouring datasets, defined as above. Then the global sensitivity Δ_f of f is defined as follows:

$$\Delta_f := \sup_{D,D' \in X, D \simeq D'} \frac{d_Y(f(D), f(D'))}{d_X(D, D')}. \tag{3.2}$$

Definition 3.2.2 (Gaussian Mechanism)**.** Let f and Δ_f be defined as above, then the Gaussian mechanism (GM), which is applied to the output y of f is defined as follows:

$$\mathrm{GM}(y) = y + \xi, \tag{3.3}$$

where $\xi \sim \mathcal{N}(0, \sigma^2 \mathbf{I}^n)$ and σ is calibrated to the global sensitivity of f. $\mathbf{I}^n$ denotes the identity matrix of size $n \times n$.

Sub-sampling amplification One concept that is frequently applied in DP machine learning is *privacy amplification by sub-sampling*. It ensures that a differentially private mechanism, which is executed on a random sub-sample of a population, provides better privacy guarantees than when it would be run on the whole dataset [92]. This enables the utility of DP mechanisms in practice since the privacy guarantees would otherwise be unusably low. Furthermore, it aligns well with many machine learning techniques that already involve sampling operations, such as stochastic optimisation methods.

3.3 Differential Privacy on Graph Structures

Originally, DP and DP-SGD have been designed for datasets, where the notion of an individual is clearly defined – such as image datasets or tabular data. On images, for example, DP has been studied in great detail and privacy-utility trade-offs have been reduced significantly. However, applying DP methods to models on graph-structured data raises additional challenges. DP-SGD operates on the level of per-sample gradients. However, for some graph learning tasks, they are challenging to define. When performing node- or edge-level predictions on a graph, for example, based on the message-passing methodology over neighbourhoods, individual nodes impact each other during training. Therefore, DP methods need to be adapted to graph learning purposes.

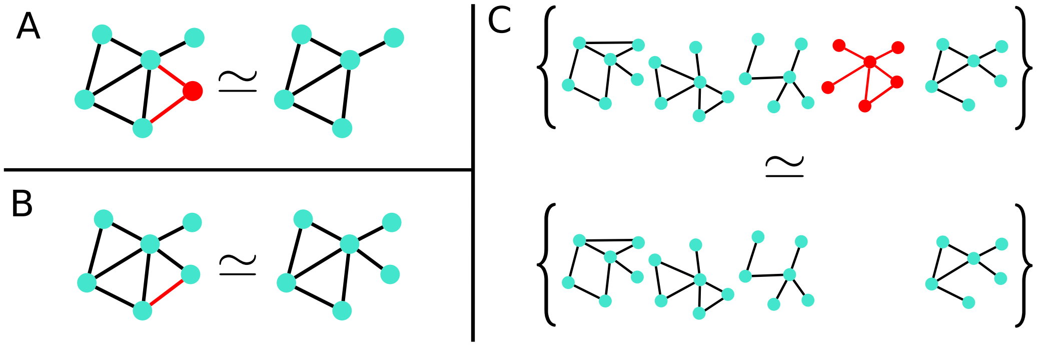


Figure 3.1: Different notions of DP on graph-structured data. **A** indicates node-level DP, where two neighbouring graphs differ in one node and its adjacent edges, **B** edge-level DP, where two neighbouring graphs differ in one edge, and **C** graph-level DP, where two datasets differ in one graph.

We can distinguish between single-graph datasets and multi-graph datasets. In single-graph datasets, it is (a) more difficult to distinguish between individual data points, since nodes are connected with each other and impact each other and (b) different parts of the graph (e.g. nodes or edges) can be considered private information and might require protection. Therefore, different notions of DP on graph-structured data can be distinguished that are based on the different definitions of neighbouring datasets. Figure 3.1 summarises the three main notions of DP on graph-structured data. Node-level DP (**A**) considers two graphs to be neighbouring if they differ in exactly one node and its adjacent edges. For edge-level DP (**B**), two datasets are neighbouring if they differ in one edge, and graph-level DP (**C**) covers multi-graph datasets, where neighbouring datasets differ in one graph. The DP notion of choice depends on the application and the data that is considered private in the respective setting. While edge-level DP considers the information of which nodes are connected to each other as private, but the node features as general knowledge, node-level DP protects both node features and edges. Node-level DP is a strictly stronger guarantee than edge-level DP. Graph-level DP is of interest when working with multi-graph datasets, where each graph as a whole is considered private. An example of this would be a dataset of molecules, which are considered sensitive information. There exist more variants of these three main notions of DP on graph-structured data, which can be found in Chapter 8.

3.4 Differential Privacy for Graph Neural Networks

There are different methods to ensure DP training of GNNs. Here, the desired notion of DP is essential for choosing an appropriate method. In Chapter 9, we introduce an extension of DP-SGD for whole-graph prediction tasks. We apply DP-SGD for whole-graph classification to several medical and non-medical datasets, evaluating privacy-utility trade-offs and the applicability of the method in different contexts.

When performing node- or edge-level predictions, the impact of one node on the whole training process can vary greatly based on its edge degree. A node with a lot of out-going edges impacts much more neighbouring nodes than a node with only one out-going edge. Therefore, several methods for node-level DP bound the maximum degree of nodes, which bounds the input of a single node on other nodes. Daigavane et al. [93] introduced a privacy amplification by sub-sampling technique for multi-layer GNNs for node-level predictions. They use DP-SGD and a sub-graphing technique that allows them to bound the influence of other nodes. We apply this method to medical population graphs and study the influence of the graph structure on performance under different privacy levels in Chapter 10. We discuss how different homophily values differently impact the performance of the GNNs on population graph datasets. We find evidence that DP has a stronger negative impact on the performance of GNNs on low-homophily graphs compared to datasets with high homophily. Developing methods to ensure DP training of GNNs for different tasks is still an active research area. Xiang et al. [94], for instance, introduce a novel Hetero Poisson sampling method to extract sub-graphs of the dataset and use symmetric multivariate Laplace noise instead of Gaussian noise for preserving node-level DP.

Alternative methods for ensuring DP for GNNs use, e.g., private aggregation of teacher ensembles (PATE) [95] approaches. Here, a collection of teacher models is trained on disjoint datasets and kept private. These teacher models are then used to train separate student models. PATE methods rely on the availability of unlabeled public datasets for training the teacher models. Works like [96] use PATE methods for differentially private GNN training. However, in medicine, large public datasets are often difficult to obtain. In the following parts of this work, we only utilise DP-SGD-based methods to ensure differentially private training of graph neural networks.

4. A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations

Tamara T. Mueller, Sophie Starck, Alina Dima, Stephan Wunderlich, Margarita Bintsi, Kamilia Mullakaeva, Rickmer Braren, and Daniel Rueckert, Anees Kazi, and Georgios Kaissis

Journal: Transactions in Machine Learning Research (TMLR), January 2024.

Synopsis: Graph neural networks are powerful tools that enable deep learning on non-Euclidean data structures like graphs, point clouds, and meshes. They leverage the connectivity of data points and can even benefit learning tasks on data, which is not naturally graph-structured –like point clouds. In these cases, the graph structure needs to be determined from the dataset, which adds a significant challenge to the learning process. This opens up a multitude of design choices for creating suitable graph structures, which have a substantial impact on the success of the graph learning task. However, so far no concrete guidance for choosing the most appropriate graph construction is available, not only due to the large variety of methods out there but also because of its strong connection to the dataset at hand. In medicine, for example, a large variety of different data types complicates the selection of graph construction methods even more. We therefore summarise the current state-of-the-art graph construction methods, especially for medical data. In this work, we introduce a categorisation scheme for graph types and graph construction methods. We identify two main strands of graph construction: static and adaptive methods, discuss their advantages and disadvantages, and formulate recommendations for choosing a suitable graph construction method. We furthermore discuss how a created graph structure can be assessed and to what degree it supports graph learning. We hope to support medical research with graph deep learning with this work by elucidating the wide variety of graph construction methods.

Contributions of thesis author: literature research, development of categorisation schemes, figure design, manuscript writing.

Copyright: Open access article.

A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations

Tamara T. Mueller[1] **Sophie Starck**[1] **Alina Dima**[1] **Stephan Wunderlich**[2,3]
Kyriaki-Margarita Bintsi[4] **Kamilia Zaripova**[5] **Rickmer Braren**[2]
Daniel Rueckert[1,4] **Anees Kazi**[6] **Georgios Kaissis**[1,7]

[1] *AI in Medicine and Healthcare, Technical University of Munich, Germany*
[2] *Department for Interventional Radiology, Technical University of Munich*
[3] *Department for Radiology, Ludwig-Maximilians-University Munich*
[4] *BioMedIA, Imperial College London, UK*
[5] *Department for Computer Aided Medical Procedures and Augmented Reality, Technical University of Munich*
[6] *Laboratories for Computational Neuroimaging, Harward Medical School*
[7] *Machine Learning in Biomedical Imaging, Helmholtz Munich*
Contact: {tamara.mueller; g.kaissis}@tum.de

Reviewed on OpenReview: *https://openreview.net/forum?id=sWlHhfijcS*

Abstract

Graph neural networks are powerful tools that enable deep learning on non-Euclidean data structures like graphs, point clouds, and meshes. They leverage the connectivity of data points and can even benefit learning tasks on data, which is not naturally graph-structured –like point clouds. In these cases, the graph structure needs to be determined from the dataset, which adds a significant challenge to the learning process. This opens up a multitude of design choices for creating suitable graph structures, which have a substantial impact on the success of the graph learning task. However, so far no concrete guidance for choosing the most appropriate graph construction is available, not only due to the large variety of methods out there but also because of its strong connection to the dataset at hand. In medicine, for example, a large variety of different data types complicates the selection of graph construction methods even more. We therefore summarise the current state-of-the-art graph construction methods, especially for medical data. In this work, we introduce a categorisation scheme for graph types and graph construction methods. We identify two main strands of graph construction: static and adaptive methods, discuss their advantages and disadvantages, and formulate recommendations for choosing a suitable graph construction method. We furthermore discuss how a created graph structure can be assessed and to what degree it supports graph learning. We hope to support medical research with graph deep learning with this work by elucidating the wide variety of graph construction methods.[8]

1 Introduction

Graphs can be used to represent several kinds of real-world datasets, such as networks, interactions, connections, or information flows. They hold information encoded in a set of nodes and edges, which connect pairs of nodes. They can add a structural component to otherwise independent data points. A wide variety of data can be structured as graphs, such as knowledge (Ruan et al., 2021), (3D) structures in space (Wolterink & Suk, 2021), brain signals (Kim et al., 2021), or maps (Yu et al., 2021c). Yet, the question of how to construct an appropriate graph structure from a given dataset can be non-trivial.

[8] The authors used ChatGPT for minor writing support.

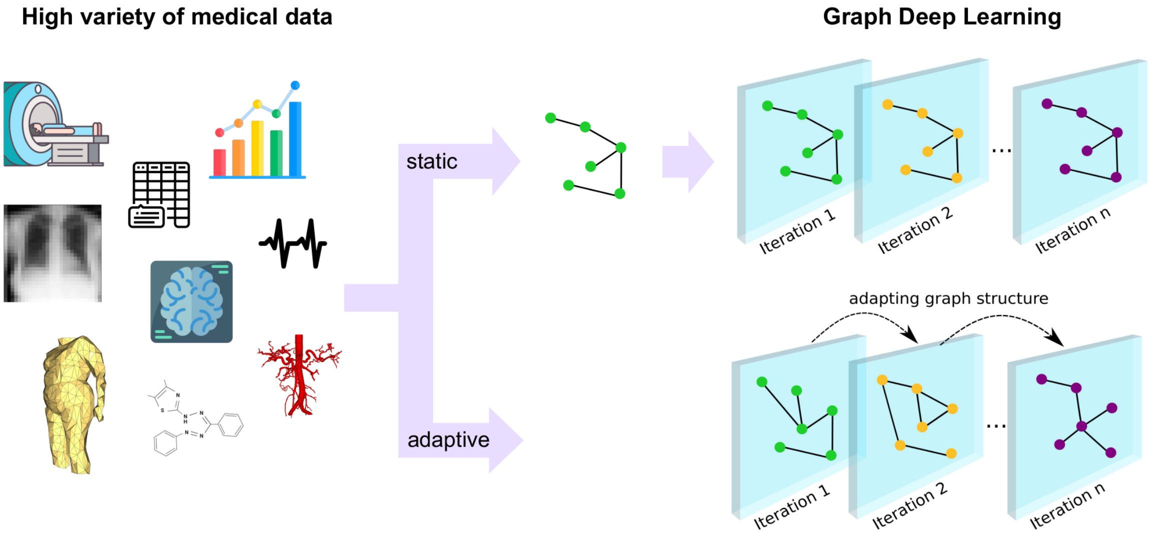


Figure 1: In this work, we summarise the state-of-the-art graph creation methods that allow one to transform a high variety of medical datasets into graph structures to perform graph deep learning. Static graph creation methods (upper row) extract the graph structure prior to learning, while adaptive methods (lower row) change the graph structure during training.

Graph learning techniques have been designed to apply deep learning (DL) methods directly to non-Euclidean datasets like graphs or meshes (Bronstein et al., 2017). These methods have since been frequently applied to data that can efficiently be structured by using a graph, for example, social networks (Fan et al., 2019) or molecules (Moreira-Filho et al., 2022), and have addressed tasks like friendship recommendations or drug discovery. Graph deep learning also naturally benefits many applications on medical datasets, as Graph Neural Networks (GNNs) have proven to be powerful algorithms for downstream tasks like medical diagnosis (Parisot et al., 2017), or image segmentation (Xie et al., 2022). They allow for straightforward integration of multi-modal data, such as image features and clinical data, into one coherent data structure, which has been explored in the context of so-called population graphs, where a medical cohort is represented by a graph structure instead of a tabular database (Kazi et al., 2022; Parisot et al., 2017).

Even in cases where a graph representation is not the default choice for a dataset, it has been shown that imposing such a graph structure by leveraging connections between data points can improve the performance of ML algorithms (Parisot et al., 2017; Ahmedt-Aristizabal et al., 2021; Bessadok et al., 2022; Pellegrini et al., 2022). This way, relations can be utilised or newly discovered, which can be beneficial for the task at hand (Cosmo et al., 2020). The application of GNNs to point cloud datasets, for example, can improve model performance compared to only using individual data points (Wang et al., 2019). This has also been shown in the medical field for tasks like disease prediction (Parisot et al., 2017), vessel segmentation (Paetzold et al., 2021), or the interaction between symptoms, diseases, and medication (Ruan et al., 2021). Here, spatial proximity (Yao et al., 2022; Hansen & Heinrich, 2021), medical knowledge (Mueller et al., 2022a), anatomical structures (Sun et al., 2021), correlations (Kim et al., 2021), or cartographic location (Yu et al., 2021c), have been used to generate a graph structure from previously disconnected data points.

This additional processing step of generating a graph structure introduces new challenges to the overall downstream task (Ahmedt-Aristizabal et al., 2021) and the definition or fine-tuning of nodes and edges hold a range of crucial design choices. This has turned out to be especially challenging in many medical settings since for medical images or health reports a graph structure is not the default choice of representation. Brain connectivity graphs have, for example, shown to be a suitable representation of the human connectome (Bessadok et al., 2022), which represents a map of neural connections in the brain. However, the construction of the brain graph holds challenges like the temporal component of functional magnetic resonance imaging (fMRI). Tube-like structures like airways (Selvan et al., 2020) and vessels (Paetzold et al., 2021) can be accurately represented by a graph that follows the anatomy of the structure at hand. Still, the concrete extraction of the graph structure requires precisely segmenting the curvilinear structures or transforming branching points into nodes.

In addition to the wide variety of graph construction methods, the strong impact of the graph structure on the success of the learning task makes this especially challenging (Luan et al., 2022). We thus conclude that the construction of a suitable graph structure is crucial to optimally leverage the connectedness inherent to the dataset.

1.1 Why the graph construction matters

Since the introduction of GNNs, many works have shown that GNNs can improve performance on non-Euclidean datasets compared to graph-agnostic DL models (Cosmo et al., 2020; Parisot et al., 2017; Ahmedt-Aristizabal et al., 2021). However, this assumption does not apply to all settings and datasets and recent works have demonstrated that GNNs outperform graph-agnostic models only under specific circumstances. This can often be attributed to the utilisation of unsuitable graph structures and can even lead to simple graph-agnostic methods outperforming GNNs (Luan et al., 2022; Zhu et al., 2020). One of the reasons for this might be an over-smoothing of node features over unideal neighbourhoods (with, for example, highly diverse labels), which complicates the establishment of suitable feature embeddings required for the downstream task. When node features of neighbours with highly different labels (and therefore different node features) get averaged during message passing, the resulting node embedding might be an over-smoothed representation that merges node features of different labels.

The interaction between the graph structure and the model performance has been investigated in several works. One question of interest here is how and under which circumstances the graph structure hinders or benefits graph deep learning. In this context, several graph metrics that assess the graph structure have been introduced that are strongly correlated with GNN performances. One such metric is *homophily* (and its counterpart: heterophily) (Luan et al., 2021; Ma et al., 2022), which quantifies the similarity of neighbouring labels. Originally, GNNs were built on the assumption that connected nodes share similar properties (they are homophilic), and GNNs perform well based on this assumption (McPherson et al., 2001). As a result, several graph DL models underperform on datasets with diverse neighbourhoods (heterophilic graphs). More metrics and their impact on GNN performance are discussed in Section 5.2.

Even though this line of research implies that the graph structure and, therefore, the graph construction method strongly impacts the performance of GNNs, the analysis of the works of GNNs in medicine shows that there is no unique, clearly defined method nor any guidelines for creating the graph structures from the wide variety of medical datasets. In this work, we, therefore, survey recent works that address graph creation methods for graph deep learning tasks with a focus on medical data. The methods summarised in this review are not limited to applications on medical datasets, and we provide links to other non-medical domains in Section 6.

1.2 Contributions and outline

This work provides an overview of graph construction methods in medicine. We performed a literature search on *Google Scholar* based on keywords like “geometric deep learning”, “graph neural network”, “medicine”, “population graph”, “disease”, “graph construction”, and combinations of them. We summarise 78 works and categorise them by their graph construction method. The outline of this work can be summarised as follows:

- We identify three types of graphs that can be distinguished: population-level graphs, subject-level graphs, and subject-independent graphs, which we use to categorise the included works (Section 3), as well as two structure types: relationship-based structures and spatially motivated graphs;
- In Section 4, we systematise existing works that utilise GNNs in medical application areas by graph construction methods with a focus on static and adaptive graph construction (see Figure 1);
- We formulate recommendations for choosing suitable graph construction methods in Section 5.3;

- We summarise existing graph assessment metrics that allow the evaluation of generated graph structures in Section 5.2;
- We embed our work into the context of related review papers in Section 6;
- We identify open challenges of graph learning in medicine (Section 7) and conclude with promising future directions of research (Section 8).

2 Background

In this section, we give an overview of graphs, GNNs, and homophily - a main graph property linked to the performance of GNNs.

2.1 Formal definition of graphs

Throughout this work, we discuss datasets involving graph structures. A graph $G := (V, E)$ is defined as a collection including a set of nodes/vertices V and a set of edges E connecting nodes. $n = |V|$ denotes the number of nodes in the graph. An edge $e_{ij} = (v_i, v_j)$ defines the connection from node v_i to node v_j. A graph G is undirected if and only if $e_{ij} \Rightarrow e_{ji}, \forall i, j \in \{1, \ldots, n\}$. All edges E can be represented in the adjacency matrix $\mathbf{A}$ of size $n \times n$, where $\mathbf{A}_{ij} = 1$ if $e_{ij} \in E$ and 0 otherwise. A weighted graph $G_w := (V, E, \mathbf{W})$ additionally requires a weight matrix $\mathbf{W}$ that assigns a weight to every edge in the adjacency matrix. The weight matrix has the same dimensions as the adjacency matrix. A neighbourhood $\mathcal{N}_v$ of a node $v \in V$ is defined by a set of all nodes that have an incoming edge to node v: $\mathcal{N}_v := \{u \in V | e_{uv} \in E\}$. A node v can be represented by a feature vector $x_v \in \mathbb{R}^m$. The features of all nodes (node features) can be summarised by the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$. In this work, we summarise, categorise, and investigate different methods to build the node feature matrix $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, and the weight matrix $\mathbf{W}$ from different datasets.

2.2 Graph neural networks

Graph neural networks (GNNs) were first introduced by Gori et al. (2005) and further extended by Scarselli et al. (2008). The term summarises a branch of research that expanded DL methods to non-Euclidean datasets, using graph convolutions. Over the last years, several different graph convolutions have been introduced. GNNs are based on a message-passing scheme, where the information stored in the nodes is propagated among neighbouring nodes, following the graph's edges. We here define the message passing at the example of graph convolutional networks (GCNs) (Kipf & Welling, 2016) but note that the principle is easily transferable to other graph convolutions.

Definition 2.1 (Graph convolutional networks (Kipf & Welling, 2016)) *Let $h_v^{(k)}$ define the feature representation of node v at layer k. For GCNs, the initial node representation for all nodes $v \in V$ is defined as following:*

$$h_v^{(0)} = x_v. \tag{1}$$

The node embedding of node v at step k is then defined as:

$$h_v^{(k)} = f^{(k)}\Big(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}_v} h_u^{k-1}}{|\mathcal{N}_v|} + B^{(k)} * h_v^{(k-1)}\Big), \tag{2}$$

where the function f, the weight matrix W, and the bias B are k-dependent learnable parameters that are shared across all nodes (Daigavane et al., 2021).

The embedding of node v of the previous step ($h_v^{(k-1)}$), as well as the sum of all neighbouring node embeddings, are combined in the new node feature representation at step k. Different graph convolutions use varied versions of this definition but also follow the message-passing scheme. For more information about GNNs, we refer to Wu et al. (2020).

2.3 Graph homophily and heterophily

A key statistical property of graphs that indicates how nodes of different labels are connected throughout the entire graph structure is homophily (Luan et al., 2021). This property has been shown to potentially have a significant impact on the performance of GNNs (Zhu et al., 2020). In general, three types of homophily can be distinguished that all focus on a different nuance of the metric: node, edge, and label homophily (Luan et al., 2021). *Edge* homophily (Zhu et al., 2020; Ma et al., 2022) in graphs is defined as the ratio of edges that connect nodes with the same label vs. different labels (see Equation 3). *Node* homophily (Pei et al., 2020) describes the average number of direct neighbours with the same label. *Class* homophily (introduced by Lim et al. (2021) and termed by Luan et al. (2021)) is an extension of edge homophily with the additional consideration for class imbalance. Formal definitions of node and class homophily can be found in Luan et al. (2021). We here define edge homophily since this is the most commonly used metric to assess graphs.

Definition 2.2 (Edge homophily) *Formally, the edge homophily of a graph $G := (V, E)$ and the set of node labels $Y := \{y_u; u \in V\}$ is defined as:*

$$h(G, Y) := \frac{1}{|E|} \sum_{e_{uv} \in E} \mathbb{I}(y_u = y_v), \tag{3}$$

where $\mathbb{I}$ is the indicator function.

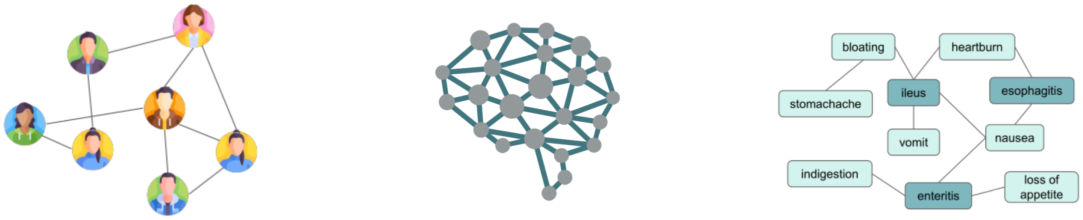
In case half of the edges in a graph connect nodes with different labels and the other half connects nodes with the same label, the graph has edge homophily of 0.5. A graph is described as *homophilous* when h is large (typically larger than 0.5) and as *heterophilous* otherwise (Kim & Oh, 2021). Homophily is only one metric to assess a graph structure and still holds some drawbacks regarding comparability between datasets and the direct impact on the performance of the downstream model (Platonov et al., 2022). More details about homophily and further graph assessment metrics can be found in Section 5.2.

3 Graph structures in medicine

Medical research and data often contain patient data that defines the structure of the dataset. We identify three distinct graph types that are used in medical applications: (1) *population-level graphs*, where typically individuals of a cohort are connected in a large graph, (2) *subject-level graphs*, where each subject is represented by an individual graph –leading to a multi-graph dataset–, and (3) *subject-independent graphs*, which represent more general structures, such as knowledge graphs, molecules or maps. Each graph type comes with individual challenges and utilises different methods for graph creation. In this section, we give an overview of those three graph types, which are visualised in Figure 2. We furthermore distinguish between two types of structures: (a) relationship-based and (b) spatial structures. Relationship-based structures use concepts and relationships to determine the graph structure and spatial structures use spatial information, for example, image key-points in Euclidean space. All graph types can be combined with all structure types. We summarise the combinations of graph types and structure types with examples in Figure 3.

3.1 Population graphs

One research area of graph learning in medicine utilises so-called population graphs (Figure 2(a)). They are generated by connecting all subjects in a cohort to a single (usually large) graph. The goal is to improve model performance by using interactions between the subjects/nodes in the graph. The most common structure of a population graph is one where every subject in the dataset is represented as a node and node connectivity is, for example, defined by some distance metric between the subjects. When using population graphs, the learning task of the GNN is usually *node prediction*. Here, a prediction (e.g. classification or regression) is made for every node. This can, e.g. be a disease prediction (Parisot et al., 2017) or age prediction (Kazi et al., 2022). Population graphs are an effective method to integrate multi-modal data and enable the usage of patient data from different data sources and modalities. There are some examples where population graphs are extended with some additional components. In Gao et al. (2021) e.g., the authors create a bipartite graph, where subjects and gene expressions are represented by node entities.



(a) Population-level graphs (b) Subject-level graphs (c) Subject-independent graphs

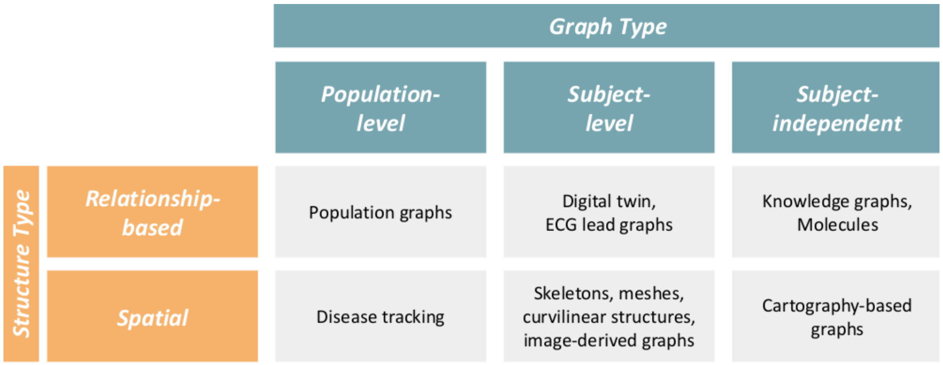
Figure 2: **Schematic display of the three graph types** utilised in medical research with GNNs: **(a)** population-level graphs, where each node represents a subject and subjects are connected based on similarity, **(b)** subject-level graphs with the example of a brain connectivity graph, where each subject is represented by a separate graph structure **(c)** subject-independent graph structures, here represented by a schematic display of a knowledge graph. The last category summarises graphs that represent more general aspects compared to subject- or population-level graphs that are not linked to medical subjects.

3.2 Subject-level graphs

Another way to represent medical data is in the form of *subject-level* graphs. This term summarises various graphs, where each subject in a dataset is represented by a single graph. The individual graphs are therefore independent of each other and together constitute a multi-graph dataset. One commonly used example for a subject-level graph is the representation of brain images as a brain connectivity graph (see Figure 2 (b)). In general, there are numerous ways to create subject-level graphs, depending on the dataset at hand and the application. They can, for instance, be used to represent structural connectivity in graph representations of arteries (Chen et al., 2020b), brain vessels (Paetzold et al., 2021), or airways (Zhao & Yin, 2021; Selvan et al., 2020), or to model a skeleton of a human in motion (He et al., 2022). When using subject-level graphs, the most common learning task is *graph prediction* (e.g. classification or regression). However, there are also applications where node-level and edge-level predictions (Chen et al., 2020b) are targeted using subject-level graphs.

3.3 Subject-independent graphs

As a third category, we summarise subject-independent graph structures that represent more general concepts and data. The graphs in this category represent structures that are independent of individuals, such as molecules or maps, that are not tailored to subjects or cohorts of patients – in contrast to subject- or population-level graphs. This includes knowledge graphs, which encode general concepts and knowledge in graph structure, highlighting relations between different entities. They are often utilised in the context of diseases, symptoms, drugs, or genes to display their correlation or interaction. They are usually not personalised but are "intended to accumulate and convey knowledge of the natural world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities" (Hogan et al., 2021). An example of a multi-modal medical knowledge graph is PrimeKG (Chandak et al., 2023), which includes information about drugs, diseases, phenotypes, exposures, and genes. Cheng et al. (2021a) construct a knowledge graph on stroke data, and Bonner et al. (2022) review different knowledge graphs on biomedical data for drug discovery. A knowledge graph differs from population graphs in the sense that here, no individual patient data is represented as a graph, but general knowledge and connections between entities are modelled in graph form. For a more detailed review of knowledge graphs, we refer to Ye et al. (2022). Knowledge graphs can be used for different applications, either on their own or as an additional source of information for other tasks, like in Pfeifer et al. (2022). Another example would be the encoding of cartographical proximity in maps like connecting hospitals in different regions of the country (Jin et al., 2021) or connecting cities based on their local proximity (Yu et al., 2021c) or molecules, where nodes represent atoms and the edges bindings between them (Bonner et al., 2022). We consider molecules as another example of subject-independent graphs. Molecule-based datasets in the medical domain are commonly used for graph-level predictions, for example investigating drug properties (Duvenaud et al., 2015; Kearnes et al., 2016) or potential interactions between different drugs (Xu et al., 2019).



		Graph Type		
		Population-level	*Subject-level*	*Subject-independent*
Structure Type	*Relationship-based*	Population graphs	Digital twin, ECG lead graphs	Knowledge graphs, Molecules
	Spatial	Disease tracking	Skeletons, meshes, curvilinear structures, image-derived graphs	Cartography-based graphs

Figure 3: **Categorisation of graph types and structure types** with examples for the different categories. We consider three graph types: *population-level*, *subject-level*, and *subject-independent* graphs, which represent overall structures that are not linked to medical subjects. Additionally, we distinguish two types of structures: ones where the edges are based on relationships between nodes and ones that follow spatial information.

4 Graph construction methods

In the following, we summarise and categorise the state-of-the-art graph construction methods for medical data, linking them to the different graph and structure types introduced in Section 3. For graph deep learning, graph structures are usually extended to contain node features. This additional knowledge is then propagated along the graph structure during the learning process.

Generally, the graph construction process consists of two aspects: (1) defining the nodes and their features and (2) defining connections between the nodes (edge construction). Each node has a feature vector of shape m and all node features of a graph $\mathcal{G}$ with n nodes are summarised by a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$. The edges can be summarised by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. The definition of the adjacency matrix and node feature matrix are in general intertwined, and both steps are necessary to extract the full graph structure.

In the following sections, we investigate the definition of nodes and node features as well as the creation of the graph structure itself (the edges). An overview of the categories for graph construction methods, with a focus on the definition of the graph's edges, is visualised in Figure 4.

4.1 Defining the graph's nodes

The extraction of the graph's nodes is highly dependent on the constructed graph type. When building **population-level graphs**, every node usually represents a subject in the dataset. Node features can, for example, contain tabular data like lab results (Parisot et al., 2017), images (Keicher et al., 2021), image-derived features (Parisot et al., 2017), or combinations of those (Keicher et al., 2021).

For **subject-level graphs**, the node feature extraction can vary greatly. We here summarise the most prominent node and node feature definition strategies for different subject-level graphs. For the creation of brain connectivity graphs from fMRI data, the most commonly used approach to define node features is to define regions of interest (ROIs). Here, the definition of nodes is often guided by a 3D atlas, which defines the ROIs from the recorded BOLD signal (Wang et al., 2022b). We, therefore, say the definition of the graph nodes relies on prior knowledge (it is prior-driven). There are some examples where slightly different approaches are utilised to define the final nodes of a graph connectivity graph. For example, Zheng et al. (2022a) identify the brain's most informative regions through sub-graph generation. Yao et al. (2021) use several templates with varying ROI parcellation scales to create coarse-to-fine brain connectivity networks for each subject instead of depending on a specific brain parcellation. Yao et al. (2022), for instance, build a graph from image data and use features extracted from a convolutional neural network (CNN) as node features of the graph. To extract a subject-level graph that represents curvilinear structures, like vessels or airways, branching points of the structure can be used to define the nodes of the graph. However, nodes

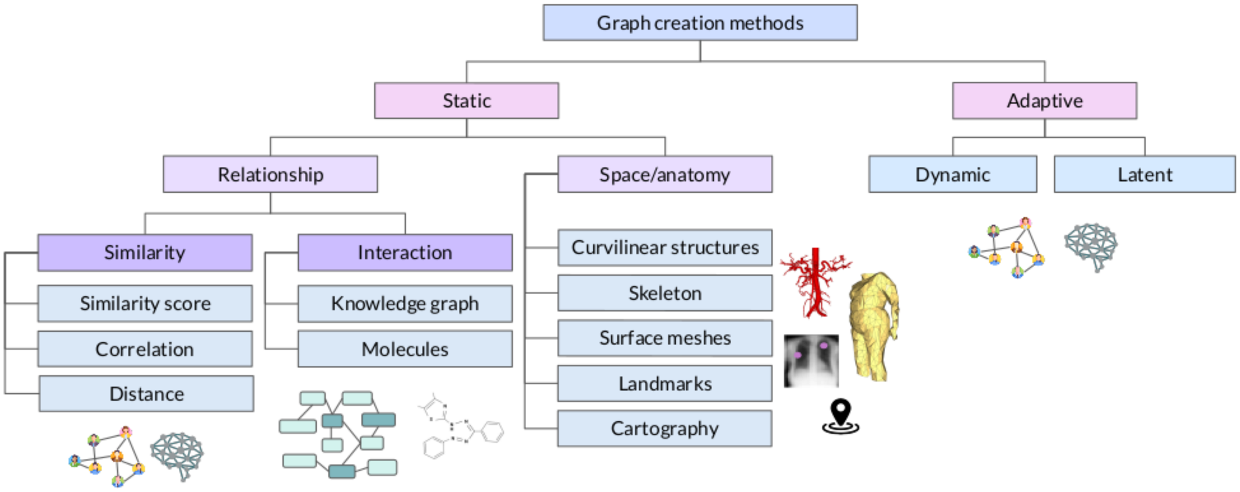


Figure 4: **Overview of graph construction methods** for GNN training in medicine categorised by static and adaptive graph construction. The icons represent examples of represented data in the respective categories. For static graph construction methods, the previously introduced separation of structure types by relationship-based and spatial interactions remains.

can also represent a section of the tube (Paetzold et al., 2021). Since in most cases, the definition of node features and edges for subject-level graphs are intertwined, we provide more information about the node features of subject-level graphs in Section 4.2.

Subject-independent graphs cover a wide range of different graph structures and datasets that all represent data that is independent of patients or subjects. This includes knowledge graphs, molecules, or cartography-based data of maps. When building knowledge graphs, the nodes typically represent entities of interest, such as diseases, symptoms, or medications. For molecules in drug research, for example, the nodes usually represent the molecules' atoms (Zheng et al., 2021; Zhao et al., 2021), and for cartography-based graphs, cities or hospitals can, for instance, be used to encode the graph's nodes (Yu et al., 2021c; Jin et al., 2021).

4.2 Defining the graph's edges

In this section, we introduce different methods for defining the structure of the graph (the edges). We hereby categorise existing works based on the following criteria:

(a) Graph type: population-level, subject-level, and subject-independent graph structures

(b) Static and adaptive graph construction mechanisms

(c) Purely data-driven and prior-driven methods

Graph creation methods can be categorised into *static* and *adaptive* approaches. We consider a graph creation method as *static* if the adjacency matrix is generated prior to training without any adaptions during the learning pipeline and as *adaptive* if the graph structure is adapted during training. The different static and adaptive graph creation methods are summarised in Table 1 and 2, respectively. Both tables also indicate the generated graph type and whether the approach is *data-* or *prior-driven*. We consider methods that only use the dataset at hand as *data-driven* and ones that also include additional prior knowledge as *prior-driven*.

4.3 Static graph construction methods

In this section, we categorise different static graph construction methods by the metric and information utilised to define the edges between the nodes. We mainly discuss graph type, connection mechanism, and the utilised data. The available methods are summarised in Table 1. In column "Category" we categorise each method by the utilised relation for edge construction. "Graph type" indicates the specific type of graph (Section 3) that is generated, "Graph construction" refers to the utilised metric or property to decide when an edge is added to the graph, the represented data is listed in the respective column, "P/D" indicates whether the graph is prior- or data-driven and the listed references give examples of works that utilise the respective methods, including the initial introduction of a specific method.

Table 1: **Summary of static graph construction methods**. We differentiate between different graph types, construction methods, and source data types. We also indicate if the method is prior-driven (P), data-driven (D) or both (P, D), as well as if the graph is weighted (**W**).

Category	**Graph type**	**Graph construction**	**Represented data**	**P/D**	**W**	**References**
Similarity and distance	Population level	Similarity score	Patient cohort	D	✗	Parisot et al. (2017); Ghorbani et al. (2022); Vivar et al. (2021) Pellegrini et al. (2022); Peng et al. (2022); Pan et al. (2021); Kazi et al. (2019); Ghorbani et al. (2021)
	Population level	Similarity score	Image features, clinical data	D	✓	Lin et al. (2023); Qiu et al. (2021)
	Population level	Mutual information	Images, clinical data	D	✗	Keicher et al. (2021)
	Population level	Euclidean distance	Image features, clinical data	D	✗	Lu et al. (2022); Yu et al. (2021b)
	Subject level	Euclidean distance	EEG signal	D	✗	Demir et al. (2021)
	Subject level	Euclidean distance	Images	D	✗	Sun et al. (2021)
	Subject level	Euclidean distance	Airways	D	✗	Tan et al. (2021)
	Subject level	Cosine similarity	Images	D	✗	Mahapatra et al. (2022)
	Subject level	Morphological similarity	fMRI data	D	✗	Mahjoub et al. (2018)
	Subject level	Pearson correlation	Brain connectivity	D	✗	Kim et al. (2021)
	Subject level	Partial correlation	Brain connectivity	D	✓	Li et al. (2021c)
	Subject-independent	Conditional probability	Lesion types	D	✗	Cheng et al. (2021b)
Relation	Population level	Medical assessments	Patient cohort	D	✓	Mao et al. (2022)
	Subject-independent	Known interactions	Disease/symptom/medication	P	✓	Ruan et al. (2021)
	Subject-independent	Interactions	Ontologies	P	✓	Hao et al. (2021)
	Subject-independent	Synergism/antagonism	Drugs	P	✓	Zheng et al. (2021)
	Subject-independent	Drug–protein pairs	Drugs	P	✓	Zhao et al. (2021)
	Subject-independent	Protein interactions	Proteins	P	✗	Schulte-Sasse et al. (2021)
	Subject-independent	Co-occurrences	(Clinical) abnormalities	P	✓	Liu et al. (2021); Zhou et al. (2021)
	Subject-independent	Co-occurrences	Medical labels	D	✓	Hou et al. (2021)
	Subject level	Medical importance	ECG leads	P	✗	Mueller et al. (2022a)
	Subject level	Protein interactions	Proteins	P	✓	Pfeifer et al. (2022)
Space, structure, or anatomy	Subject level	Local proximity	Image landmarks	D	✗	Yao et al. (2022); Hansen & Heinrich (2021); Huang et al. (2023)
	Subject level	Tree structure	Curvilinear structures	D	✓	Yu et al. (2022a)
	Subject level	Tree structure	Curvilinear structures	D	✗	Paetzold et al. (2021); Wittmann et al. (2023); Shin et al. (2019) Chen et al. (2020b); Wolterink et al. (2019); Yu et al. (2022b)
	Subject level	Local proximity	Curvilinear structures	D	✗	Xu et al. (2022); Xie et al. (2022)
	Subject level	Local proximity	Curvilinear structures	D	✓	Li et al. (2021b)
	Subject level	Anatomy	Variety of patient data	D	✗	Barbiero et al. (2021)
	Subject level	Anatomy	3D point clouds	D	✗	Yu et al. (2021a)
	Subject level	Anatomy	Skeleton	P,D	✗	He et al. (2022); Deb et al. (2022)
	Subject level	Anatomy	Curvilinear structures	D	✗	Selvan et al. (2020)
	Subject level	Mesh generation	Cortical surface meshes	D	✗	Azcona et al. (2020); Gopinath et al. (2019a); Wu et al. (2019); Gopinath et al. (2019b)
	Subject-independent	Cartography	Maps	D	✗	Yu et al. (2021c); Jin et al. (2021)
	Subject-independent	Chemical structure	Molecules	D	✗	Bonner et al. (2022); Kearnes et al. (2016); Duvenaud et al. (2015)
	Subject-independent	Text-derived	Disease/symptom/medication	P,D	✓	Vretinaris et al. (2021)
	Subject-independent	Chemical structure	Disease/symptom/medication	P,D	✓	Zhang et al. (2022b)

4.3.1 Graph construction based on relations

Following the in Section 3 introduced structure types (relationship-based and spatially motivated structures), we can differentiate these two types of static graph construction. Relationship-based graph construction methods define a relation between node features, such as similarity or interactions, and spatially motivated graph construction methods utilise spatial information to define edges.

Similarity-based graph construction

One method that is frequently used to determine whether nodes connect is based on a similarity or distance measure. Simplified speaking, in this approach, two nodes will be connected, if they are "similar" or "similar enough". The similarity/distance between nodes can be defined by different metrics, including a similarity score, the Euclidean distance, cosine similarity, or correlation-based similarity. Furthermore, methods using similarity or distance measures to construct the graph structure can be categorised into ones using the original data at hand, or a (mostly lower-dimensional) embedding of the original data (Lu et al., 2022). When using a similarity or distance measure to determine whether nodes should be connected, usually an additional step of edge selection needs to be performed to obtain the final graph (Section 4.3.4).

Similarity score The similarity between nodes in the graph can be defined with the help of a similarity score. This is one of the most commonly used methods for the creation of population graphs in order to determine how similar two subjects in a cohort are. This method was first introduced by Parisot et al. (2017) in 2017 and has since then been used in many variants and applications, e.g. in Ghorbani et al. (2022); Vivar et al. (2021); Pellegrini et al. (2022); Peng et al. (2022) and Lu et al. (2022). One advantage of this method is that the similarity score considers discrete and non-discrete features and can be adapted by adding additional weight factors. All available features are divided into two groups (e.g. imaging and demographic features), then the graph is built based on a similarity score that uses one of these subgroups of features (e.g. demographic features). The second group of features (e.g. imaging features) are used as node features. Parisot et al. (2017) originally used this approach for brain analysis in Alzheimer's disease and autism prediction. They used phenotypic information for the graph creation and imaging features as node features and show an example of multi-modal data integration.

Variations of this method have been used for bone age estimation (Du et al., 2015), brain age prediction (Stankevičiūtė et al., 2020), autism detection (Rakhimberdina et al., 2020), genome inference (Dilthey et al., 2015), disease prediction (Chen et al., 2020a; Kazi et al., 2019; Ghorbani et al., 2021), to only name a few. In each of the here mentioned works, the original approach of creating the population graph by Parisot et al. (2017) has been modified to best fit the data and the application. Some adaptations of this method have also used both imaging and non-imaging features for graph creation as well as node feature representation (Lin et al., 2023) to maximise the information about the similarity between individual subjects.

Correlation-based similarity Another option to create the graph structure is by evaluating the correlation between node features and connecting those nodes that show high correlation. Similar to the similarity-based creation, the correlation is thresholded and only nodes that are "correlated enough" will be connected by edges (Section 4.3.4). This method is used mainly for the creation of brain connectivity graphs from functional magnetic resonance imaging (fMRI) data. Here the correlation between the blood oxygen level dependency (BOLD) signal, indicating alterations in blood oxygen levels over time of all previously defined regions of interest (ROIs) –which are represented by the nodes in the graph– is first averaged within each brain ROI. Then, a correlation metric is calculated between pairs of regions, and the nodes that show a high correlation are connected leading to functional connectivity that illustrates the communication between different brain regions.

Pearson's correlation coefficient, which measures the linear correlation between nodes, has been commonly used to evaluate functional connectivity. Values range from -1 to 1 where the former shows perfect negative correlation and the latter perfect positive correlation. A value of 0 indicates no correlation. Alternatively, other correlation methods such as partial correlation, Ledoit-Wolf (LDW) regularised shrinkage estimator (Noman et al., 2021) and Spearman's rank correlation (Yu et al., 2023) are utilised to construct functional connectivity. Partial correlation is an extension of the Pearson correlation coefficient, which measures the linear relationship between two variables. By removing the effects of the controlling variables, partial correlation helps to identify the unique relationship between the two variables of interest, providing a clearer understanding of the associations between them. LDW regularised shrinkage estimator is a statistical method designed to improve the estimation of covariance matrices when the number of observations –number of scans– is small compared to the number of variables –number of ROIs. Noman et al. (2021) use this method to obtain well-conditioned functional connectivity. Spearman's rank correlation is a non-parametric rank correlation method between two nodes that can detect monotonic nonlinear relationships. Yu et al. (2023) propose a multi-graph attention network to use both Pearson's and Spearman's rank correlation measures. Li et al. (2021c) utilise ROI-aware GNN assuming more closely connected ROIs exert a greater effect on each other and apply node (ROI) pooling layer (R-pool) to retain the most representative ROIs while eliminating noisy nodes. Gharsallaoui et al. (2021) predict an affinity matrix of the target view from the source view. Klepl et al. (2022) compare different functional connectivity measures for Alzheimer's disease prediction from EEG data as well as different edge selection methods.

One potential shortcoming of the creation of brain connectivity graphs from the BOLD signal is that the signal and the correlation between ROIs change over time. Kim et al. (2021); Kong et al. (2021); Wang et al. (2022b) target this by creating several brain connectivity graphs for different time steps to allow a more dynamic graph structure over time. Kong et al. (2022), e.g., propose a multi-stage learning module to fully utilise features at different stages and use a deep auto-encoder to extract the graph structure.

Distance measures and mutual information In line with the usage of a similarity score, the *Euclidean distance* or the *cosine similarity* has also been used to determine the "distance" between nodes (Lu et al., 2022; Yu et al., 2021b). These distance measures can also be interpreted as a similarity between nodes and used as a basis for edge selection (Section 4.3.4), where only the least distant nodes are connected. Keicher et al. (2021) use images, extracted image features, and clinical data as node features and build the graph based on *mutual information*.

Interaction-based graph construction

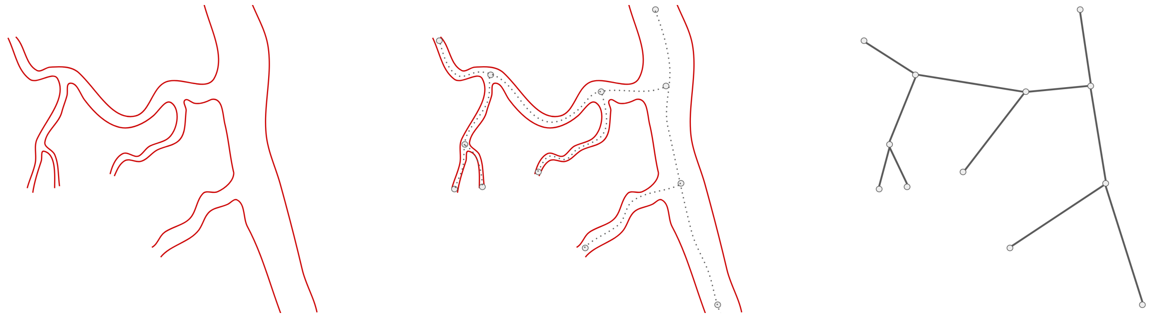
Apart from similarity- and correlation-based edge definitions, the graph structure can also be derived from (usually known) interactions between entities. This can e.g. be symptoms that are related to certain diseases, medications that are used to treat diseases, or lab results that are associated with specific doctor's visits (Mao et al., 2022).

Knowledge graphs Knowledge graphs are typically constructed based on known relations between entities that are usually extracted from large knowledge resources or clinical studies. Medical knowledge graphs can for example contain diseases, symptoms and medications, connecting co-occurring symptoms and diagnosed diseases with prescribed medication and reported symptoms. Even though the graph is constructed based on general knowledge, several design choices still remain. Zhang et al. (2020) create a knowledge graph based on clinical studies where abnormalities are connected with organs and body parts. Hou et al. (2021) follow the same graph creation approach, with an additional step of post-processing where missing links are included in a data-driven manner. Protein-protein interactions can be encoded in knowledge graphs and have, for example, been used to identify cancer genes (Schulte-Sasse et al., 2021) and drug–target interactions (Zhao et al., 2021).

The knowledge graph PrimeKG (Chandak et al., 2023) was for example built on large resources like *Drug-Bank* (Assempour et al., 2018), *Drug central* (Avram et al., 2021), and *Entrez gene* (Maglott et al., 2010) - to mention only a few. The authors highlight that PrimeKG contains edges between drugs and diseases that specify "indications", "contradictions", and "off-label use", which are often missing in other medical knowledge graphs (Chandak et al., 2023).

These constructed knowledge graphs can then be used as a basis for decision-making and graph learning. The authors in (Vretinaris et al., 2021) for example use a knowledge graph to combine generally applicable information regarding medications and symptoms with text-derived patient-specific information about their individual symptoms. Zhang et al. (2022b) utilise a similar technique and utilise an attribute graph, containing chemical structures of drugs as an additional graph to the knowledge graph.

Prior knowledge The graph structure can also be created or influenced by the inclusion of prior knowledge about the data and the importance of specific features in other areas apart from knowledge graphs. This was for example applied in a graph classification task using electrocardiogram (ECG) data to determine left bundle branch blocks (Mueller et al., 2022a), where prior medical knowledge about the importance of seeds guided the graph construction method.



(a) Vessel tree

(b) Centerline and branching points

(c) Abstract graph

Figure 5: **Example of an abstract graph representation of a vessel tree**, with nodes encoding branching points and edges encoding vessel segments. Given a vessel segmentation extracted from a 3D or 2D image (a), centerlines and branching points are automatically or semi-automatically extracted (b), from which a graph is defined (c). Although the graph is inherent to the structure of the vessel tree, the extracted graph ultimately depends on the entire processing pipeline, which is inherently noisy.

Pre-defined structural connections Some datasets come with internal structural information, from which the graph can be extracted. An example of this would be molecules, which have been used for drug discoveries (Bonner et al., 2022) or protein analysis (Pfeifer et al., 2022). Here, the graph creation usually follows the molecular bindings. For more information about these pre-defined structural connections, we refer the interested reader for example to a review on ML approaches for the design of multi-target drugs (Moreira-Filho et al., 2022).

4.3.2 Graph construction based on space, structure, or anatomy

Another graph construction technique is building spatio-structural or spatio-semantic graphs, where the nodes have associated positions in a metric space. In this case, the relations between the nodes usually correspond to distances in the respective metric space. The starting point can e.g. be a 2D or 3D image or a map, and a common choice for the metric space is the Euclidean space.

Curvilinear structures One prominent application of spatio-structural graphs in medical imaging is represented by curvilinear structures. These are long, thin, structurally constrained tubular structures, following a tree or network configuration. Examples of such structures are vessel trees, airways, or neurons. As an exemplification of their structural constraints, vessel trees follow blood flow constraints, their branching pattern is dictated by the perfusion requirements of nearby tissue, and the cross-sectional diameter of constituent vessels progressively narrows in the distal direction. The inherent graph structure of these objects makes them ideal candidates for geometric deep learning: either by achieving a more compact, abstract representation for branch classification or exploiting geometric information for an auxiliary task. The graph representation is typically only a secondary representation derived from an initial imaging representation, making graph learning highly dependent on the quality of the graph extraction process. Figure 5 shows an example of a graph representation derived from a vessel tree.

The most straightforward representation of curvilinear trees as graphs follows the structure of the trees, where branching points form the node set and individual segments are edges. Such an approach is suitable for multi-class classification tasks, such as airway or vessel labelling. Chen et al. (2020b) construct a graph of the intracranial arteries by choosing nodes as centerline points with more than two neighbours and linking them based on skeleton connectivity. Their approach to multi-class vessel labelling is formulated as simultaneous node and edge classification, where message passing occurs through both node and edge representations. Many other works opt to formulate the branch labelling problem as node classification instead since node classification is more widely spread than edge classification. Paetzold et al. (2021) introduced the VesselGraph dataset of graphs derived from mouse brains for the tasks of link prediction and vessel classification. The graph structure is extracted by first generating a segmentation of a 3D volume, then processing it via the graph extraction tool Voreen (Drees et al., 2021). Then the initial graphs with nodes as branching points and vessels as edges are subsequently converted via the line graph approach, such that nodes of the derived graph correspond to the edges of the initial graph. Xie et al. (2022) performed anatomical labelling of airways by directly constructing a graph where nodes correspond to branches and assigning edges based on the connectivity in the image segmentation map. Yu et al. (2022b) used a similar graph construction approach for airway labelling while adding additional hyperedges between the children of a parent, thereby using two information passing pathways.

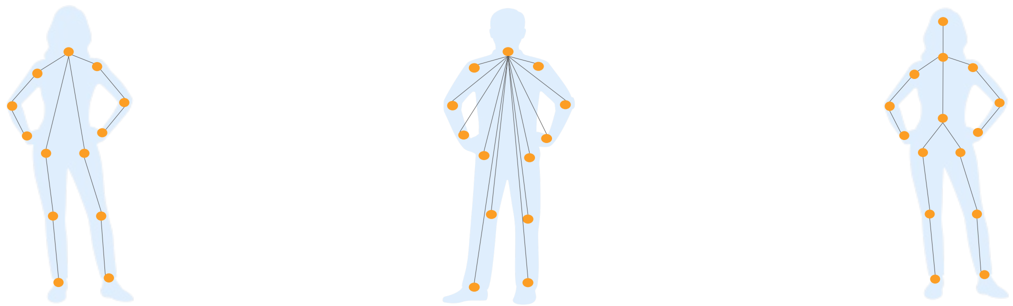
Another prominent use of curvilinear structures in graph-based deep learning is to improve image segmentation. The construction of the graph is less abstract in such scenarios, instead following the voxel structure of the data representation. Nodes are no longer well-defined anatomical structures such as branches or branching points but tend to be pixels or superpixels, while edges follow their connectivity in the image. Xu et al. (2022) first partition the input image into superpixels, which become the nodes of the vessel graph. Edges are then only defined between nodes corresponding to neighbouring superpixels in the original image. The superpixel-derived graph is used for vessel segmentation refinement, by predicting superpixel vesselness as a binary node classification problem. Similarly, Shin et al. (2019) proposed to train a node classifier for segmentation refinement also using vesselness prediction. However, in their approach nodes are represented by points sampled equidistantly along the centerlines rather than superpixels, whereas edges are based on node connectivity or geodesic distances.

Some works (Yu et al., 2022a; Li et al., 2021b; Tan et al., 2021) only use GNNs during training for improved CNN feature representation for image segmentation, rather than as a standalone task. Yu et al. (2022a) partition the image into super-voxels and sample a node per super-voxel, relying on travel distance between nodes in the ground truth for determining edges, while Li et al. (2021b) rely on corner sampling following skeletonization for node creation and a similar approach for determining edges. Tan et al. (2021) perform airway segmentation via CNNs with concatenated GNN features by combining nodes around landmarks and nodes sampled at random lung locations. The adjacency matrix is created by connecting the k nearest neighbours, in addition to connecting another node with the same predicted semantic class for each node to improve homogeneity.

Wolterink et al. (2019) use a different approach for vessel segmentation. Instead of working in the pixel space, they turn to the polar space centred around the vessel centerline and formulate the vessel segmentation problem as regression. For each cross-section of the centerline, a node for the centerline and 24 additional nodes around the centerline are created. Thus, the GNN is trained to regress the distance to the centerline. Each non-centerline node is connected to its corresponding centerline node, the corresponding nodes in adjacent cross-sections, as well as 2 neighbouring nodes. Additional connections are defined to establish a triangle mesh, whose vertex coordinates are determined by the regressed distances in the polar space.

Skeleton representation Another application that follows anatomical structures for graph creation, is the representation of the human skeleton in graph structure. Deb et al. (2022) use skeleton representation and GNNs for the assessment of physical rehabilitation exercises. Similar representations have been used for the detection of Parkinson's disease, where skeletons are extracted from video data of patients (He et al., 2022). The authors use two different methods for generating the graph structure: (1) connecting the joints of the skeleton locally, following the underlying skeleton (prior knowledge), or (2) globally, where every joint of the skeleton is connected to one node at the position of the neck. This shows that even when the data contains natural connectivity, there are multiple ways to extract a suitable graph structure for graph deep learning. The different representations are visualised in Figure 6.

Surface Meshes Meshes are a special type of graph structure, that can benefit from graph deep learning Bronstein et al. (2017). A mesh usually represents a 3D structure in space, where nodes are aligned around the surface of an object. Edges usually build triangular faces between neighbouring nodes, which is why they are often also referred to as *triangulated* meshes. Meshes have also been used for medical data representation, such as cortical structures (Azcona et al., 2020; Gopinath et al., 2019a; Wu et al., 2019; Gopinath et al., 2019b) in neuroscience or organ surfaces (Mueller et al., 2022b). Meshes can, for example, be extracted from segmentations of 3D images, using methods like marching cubes (Lorensen & Cline, 1998). Other works



(a) Local connection module: here the graph structure follows the underlying skeleton of a person.

(b) Global connection module: all relevant joints are connected to one global node at the neck of a person.

(c) Another local connection module: here the spine is also included in the skeleton representation.

Figure 6: **Schematic visualisation** of two approaches to create a graph structure for skeleton representations that are introduced in He et al. (2022) and Duan et al. (2022).

utilise finite element meshes (Salehi & Giannacopoulos, 2022), for example, using methods like TetGen (Hang, 2015). Mueller et al. (2023c) use body surface meshes for the quantification of different types of fatty tissue. Surface meshes are frequently used in neuroscience. The authors in Salehi & Giannacopoulos (2022) use cortical meshes for soft tissue prediction in image-guided neurosurgery. Salehi & Giannacopoulos (2022), e.g., use meshes for predicting soft tissue deformation in image–guided neurosurgery. Azcona et al. (2020) use surface meshes for brain morphology estimation for Alzheimer's disease classification. They analyse cortical and subcortical irregularities that have been shown to be correlated with Alzheimer's disease. A review by Zhao et al. (2023) summarises methods for cortical surface-based neuroimage analyses, including methodologies for cortical surface extraction and parcellation. Popular software tools like FreeSurfer (Fischl, 2012) have integrated methods for cortical surface extraction from MR images.

Image landmarks Graphs can also be extracted from images by identifying landmarks in the image and then connecting them based on local proximity. Hansen & Heinrich (2021) apply this method to extract lung landmarks from 3D computer tomography (CT) images in order to perform graph-based registration. They use the Foerstner interest operator (Förstner & Gülch, 1987) to identify sparse landmarks from the image, and then use a minimum spanning tree to construct the edges of the graph structure (Heinrich et al., 2015). Zhao (2021) use several graph structures in their work. One of them uses simple linear iterative clustering (SLIC) (Achanta et al., 2012) to determine the connection between super-pixels in chest X-ray images. Huang et al. (2023) use image patches as nodes, connect neighbouring nodes to construct a graph, and apply GNNs for image reconstruction.

Cartography-based graphs There are a few works, where the graph structure follows cartographic proximity, like maps or regions. Jin et al. (2021) e.g. use hospitals and regions as nodes in their graphs, connecting geologically close entities. Yu et al. (2021c) suggest representing nodes by cities in the graph structure, and the proximity of geolocation is used to connect them.

4.3.3 Fully connected graphs

Similar to pre-defined graph structures, there are some applications, where a fully connected graph is used (Chao et al., 2020). In those scenarios, no graph creation method has to be chosen. Instead, all nodes are connected to all other nodes. Chao et al. (2020), for example, use fully connected graphs for lymph node gross tumour volume detection.

4.3.4 Edge selection

In many of the above-mentioned graph creation methods, a distance/similarity/correlation/proximity between nodes is derived. However, this does not directly lead to a final graph structure. In order to create the final (sparse) graph from the determined distance between the nodes, a set of –sometimes weighted–

edges needs to be selected. This is usually either done by (a) thresholding or (b) selecting a fixed number of neighbours (k-nearest neighbours).

Thresholding edges Thresholding techniques can be further split into *absolute* thresholding and *proportional* thresholding. In *absolute* thresholding, a fixed threshold is chosen to remove weak connections which results in varying graph densities for different subjects. In *proportional* thresholding, a portion of the strongest connections is kept resulting in all graphs having the same edge density. The latter method is for instance used by (Noman et al., 2021; ElGazzar et al., 2022). In addition to those methods, Yang et al. (2021) applied the spatial-constrained sparse representation optimisation method to obtain a sparse representation matrix that captures the relationships between brain regions while considering their spatial proximity.

k-nearest neighbour selection Alternatively, the k-nearest neighbours (k-NN) of each node can be selected based on the similarity/distance/proximity between the nodes. This way, every node is connected to a fixed number (k) of neighbours. K-NN graph creation methods are applied in several application areas, e.g. for population-level graph creation and brain graph creation. In the latter, a *k-NN graph* is derived from each densely connected functional connectivity matrix. It connects each brain region of interest to the k neighbours that show the highest connectivity. This is for example used in Zhang et al. (2021b); Qu et al. (2021); El Ouahidi et al. (2022); Wang et al. (2022a); Yao et al. (2021). For population-level graphs, usually, the k-nearest neighbours are selected based on the features extracted from the dataset (Lu et al., 2022; Yu et al., 2021b). An implementation of a k-NN graph creation is for example provided by PyTorch Geometric (Fey & Lenssen, 2019).

4.4 Adaptive graph construction methods

In contrast to the previously investigated static graph creation methods, we here introduce adaptive methods, where the graph structure is adapted during GNN training. The methods using adaptive graph construction are summarised in Table 2.

Zhu et al. (2021) propose a categorisation of adaptive graph creation methods into (a) metric-based, (b) neural-based, and (c) direct approaches. Metric-based approaches (a) use kernel functions to compute a similarity between node features or embeddings and use those to determine edge weights. Neural approaches (b) use neural networks to predict the edge weights, and direct approaches (c) interpret the adjacency matrix as free variables. They refer to adaptive graph creation methods in general as “graph structure learning” (GSL). However, these terms are not common in the medical domain, where instead the terms *dynamic* and *latent* are used. *Dynamic* approaches adapt the graph structure in between training steps, but do not specifically learn the graph structure in an end-to-end manner, which *latent* graph construction methods do.

Table 2: **Summary of adaptive graph creation methods** with their corresponding graph types and examples of original data, from which the graph is extracted. D indicates, that the graph creation method is data-driven and P that it is prior-driven.

Category	**Graph type**	**Method**	**Represented data**	**P/D**	**References**
Dynamic	Subject-independent	Euclidean distance	Point clouds (non-medical)	D	Wang et al. (2019)
	Subject-independent	Euclidean distance	Proteins, molecules	D	Tran et al. (2018)
	Population level	Cosine distance	Patient cohort	D	Zheng et al. (2022b)
	Population level	Contrastive learning	Brain connectivity	D	Wang et al. (2022b)
	Subject level	Euclidean distance	Airways	D	Garcia-Uceda Juarez et al. (2019)
Latent	Population-level	End-to-end learned	Patient cohort	P,D/D	Kazi et al. (2022); Cosmo et al. (2020); Mullakaeva et al. (2022); Huang & Chung (2020); de Ocáriz Borde et al. (2023); Song et al. (2021)
	Subject level	End-to-end learned	Brain connectivity	D	Kan et al. (2022); Campbell et al. (2022); Mahmood et al. (2021); El-Gazzar et al. (2021); Zhu et al. (2022)
	Subject level	End-to-end learned	MR images	D	Mo et al. (2021a)

4.4.1 Dynamic graph construction

Dynamic graph creation methods do not construct a fixed/static graph structure before training but change the graph structure depending on the embeddings during GNN training. Wang et al. (2019) propose dynamic graph CNNs (DGCNNs) for the analysis of point clouds. They extract a k-nearest neighbour graph based on the feature space of the applied neural network, which changes in between training steps. In Tran et al. (2018), the authors extend this method with a new definition of graph convolutional filters with the goal of improving the impact of distant neighbours. They use the shortest path connections and also do not optimise the latent space with respect to the graph structure but regarding the downstream task. Their k-nearest neighbour operation is not differentiable. Tran et al. (2018) also apply their method to biological graphs like molecules and proteins, however, they have not been applied to medical datasets specifically.

Zheng et al. (2022b) introduce a framework that utilises a dynamic graph generation method for disease prediction with a focus on the integration of multi-modal data. With the addition of separate loss terms, their method optimises for the downstream task and certain properties of the adjacency matrix of the graph, using attention. Wang et al. (2022b) use contrastive learning to generate a population-level graph from fMRI data using multiple views of the fMRI scans of every subject. They use partial correlations between ROIs to learn the graph structure that represents the fMRI data. Zhao et al. (2022) dynamically generate the graph structure by aggregating calculated k-NN graphs during training. This method retains more comprehensive non-local information diffusion compared to the proximity derived from a fixed input space.

Garcia-Uceda Juarez et al. (2019) use GNNs for improved CNN feature representation, by placing a GNN as the bottleneck layer inside a U-Net (Ronneberger et al., 2015). Features from the last encoder layer corresponding to encoded super-voxels serve as nodes connected either through regular grid connectivity, or based on the nearest neighbours in the feature space.

4.4.2 Latent graph learning

Here, we summarise methods that specifically learn the graph structure in an end-to-end manner. This requires the adjacency matrix of the graph to be differentiable. When using static graph construction methods, the adjacency matrix A is often a binary discrete matrix, where an entry $a_{ij} = 1$ indicates that there is a connection from node i to node j. When using latent graph creation methods, A needs to be continuous, to allow backpropagation through the adjacency matrix (Figure 7). This is usually implemented by a *fully connected graph* with continuous edge weights that are updated during learning.

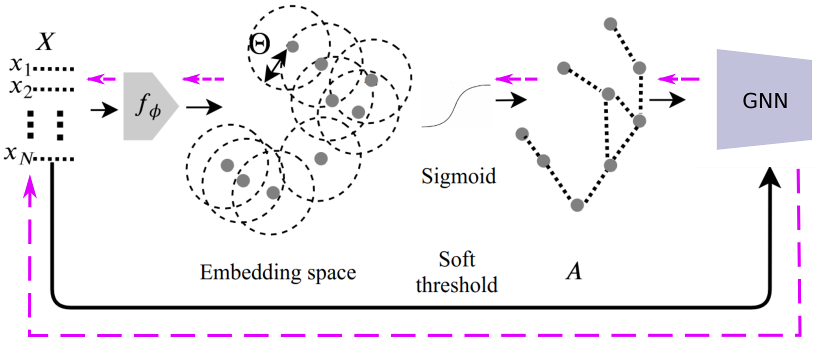


Figure 7: **Schematic overview of a latent graph learning approach**. Figure adapted from Cosmo et al. (2020); Kazi et al. (2022). The purple arrows indicate the backpropagation tracks through the networks. f_ϕ is an embedding network with learnable weights ϕ and A is the adjacency matrix, which will be updated.

The first works in this area used the spectral domain to learn the latent structure of the graph (Zhan et al., 2018; Li et al., 2018; Huang et al., 2018). These spectral methods require an initial graph which will then be changed over training and are limited to transductive learning settings since the extraction of the graph Laplacian requires the full graph including test and validation sets. Later, latent graph learning methods have also been transformed into the spatial domain, which also allows inductive training.

Cosmo et al. (2020) were the first to apply this approach in medical research for computer-aided diagnosis (CADx) and disease prediction. They could report improved performance when using a learned graph structure on two CADx problems (brain age prediction and Alzheimer's Disease prediction) compared to the approach of extracting a graph structure prior to training the GNN. In terms of graph structure learning, their method is classified as a "metric-based" GSL (Zhu et al., 2021). Mullakaeva et al. (2022) suggest extending the same graph learning method by adding additional loss to control the graph sparsity and applying it to a graph-in-graph learning task. Mo et al. (2021a) explore a similar technique for multi-modal data integration of MR images that also utilises an end-to-end trainable adaptive graph learning method. Huang & Chung (2020) also utilise an adaptive graph learning mechanism for disease prediction with the aid of population graphs. Their method includes a Monte-Carlo edge dropout for uncertainty estimation. This can be used to quantify the uncertainty of the edges learned in the graph creation method.

As an extension of Cosmo et al. (2020), Kazi et al. (2022) introduce the *differentiable graph module* (DGM). They propose two variants of their method with different sampling strategies: continuous and discrete. The latter has the advantage that a sparse graph can be sampled without the need for a fully connected adjacency matrix and requires the utilisation of the Gumbel-top-k trick (Kool et al., 2019) to ensure differentiability. In general, DGM can be applied based on an initial graph structure (prior-driven) or solely on the node features (data-driven). DGM uses Euclidean space to encode the latent features from which the latent graphs are decided. de Ocáriz Borde et al. (2023), extend the DGM to work on more complex embedding spaces by utilising Riemannian geometry.

One aspect in which static methods seem to show shortcomings in representing the full dimensionality of the data is when a temporal component is introduced. Functional brain connectivity networks e.g. fluctuate over time and can therefore benefit from an adaptive creation of the graph structure (Kim et al., 2021). We found five works that use similar approaches to learning the graph structure of functional brain connectivity networks end-to-end (Kan et al., 2022; Campbell et al., 2022; Mahmood et al., 2021; El-Gazzar et al., 2021; Zhu et al., 2022). In the method proposed by El-Gazzar et al. (2021), the adjacency matrix is randomly initialised, and its weights are adaptively learned along with the GNN weights by using gradient descent. Zhu et al. (2022) couple feature learning with dynamic graph learning into the GCN architecture. Campbell et al. (2022) and Mahmood et al. (2021) utilise self-attention in their model architecture. Kan et al. (2022) introduce loss components that enforce sparsity in the graph and maximise the similarity between adjacency matrices of the same group while minimising similarity between differently labelled subjects.

There are some methods for adaptive graph construction, that have –to the best of our knowledge– so far not been applied to medical data that we consider promising methods from which medical data analysis with GNNs could benefit. Some examples are pointer graph networks (Velickovic et al., 2020), dynamic graph message passing networks (Zhang et al., 2022a), and deep heterophily graph rewiring (Bi et al., 2022).

5 Discussion and recommendations

In this section, we give insights and recommendations for selecting suitable graph construction methods. We summarise the main advantages of static and adaptive graph creation methods (Table 3) and provide an overview of graph metrics, that can be used to assess the generated graph structure. We give recommendations regarding the selection of a graph type as well as the choice between static and adaptive graph construction and discuss some essential impacts of the graph structure on different graph convolutions, guiding a suitable choice of the latter for the downstream task.

5.1 Static vs. adaptive graph construction

Comparing Tables 1 and 2, we can see that fewer works utilise approaches that work with an adaptive graph structure creation, compared to the static approaches. In the following, we will compare both methods and discuss the shortcomings and benefits of both (Table 3).

Static graph creation methods are more computationally efficient during training since they do not require updating the graph structure. They require fewer trainable parameters and are easier to train in general. Backpropagating through the adjacency matrix and fine-tuning the adjacency matrix during training (adap-

Table 3: **Overview of advantages and disadvantages** of static and adaptive graph construction methods with respect to different aspects of the methods.

Aspect	**Static**	**Adaptive**
Training efficiency	Computationally more efficient during training	No pre-processing w.r.t graph creation required
Complexity of training	Easier to train (fewer (hyper-)parameters)	More difficult to train
Final graph structure	Generally applicable for different problems	Adjacency matrix is fine-tuned to the problem
Further utility of graph structure	Application independent graph structure	Adjacency matrix usable for interpretability
Generalisability of method	Different datasets require different methods	General method, dataset independent
Prior knowledge	Can be easily included in the graph	Some methods allow inclusion of prior knowledge

tive approach) adds additional complexity to the learning problem and can make training more difficult. On the other hand, the end-to-end learning of the graph structure eliminates critical design choices, since the adjacency matrix does not need to be defined prior to training.

Another advantage of adaptive graph creation methods is that the resulting adjacency matrix is fine-tuned to the specific task. This way, the most suitable adjacency matrix can be learned. This is not the case for static methods, where the edges are defined prior to learning. Whereas the latter makes the adjacency matrix more general and the same matrix can be used for different learning tasks on the same dataset.

Regarding the utility of the created graph structure beyond the specific downstream task, the adeptly generated adjacency matrix can also be used for interpretability purposes. Learned connections can indicate correlations between node components or highlight more important regions of the graph. Campbell et al. (2022) use their adaptive graph creation functional connectivity networks as a basis to identify sex-discriminate brain regions, which show higher importance within the decision-making progress of the algorithm.

5.2 Graph assessment

So far, we have investigated different methods to construct a graph structure from medical data and evaluated the advantages and disadvantages of different methods. Now we discuss methods that can be used to assess the constructed graph in terms of similarities in neighbourhoods. This can be done before or during model training, depending on the utilisation of static or dynamic methods. Several graph assessment metrics have been introduced to evaluate the impact of the graph structure on the learning of GNNs. Table 4 summarises the different metrics as well as a summary of their targets.

Table 4: **Summary of graph assessment metrics** that have been shown to be correlated to GNN performance and can be used to evaluate generated graph structures.

Graph Assessment Metric	**Reference**	**Details**
Edge homophily	Zhu et al. (2020)	Ratio between edges connecting same and differently labelled nodes
Node homophily	Pei et al. (2020)	Average amount of neighbours with the same label
Class homophily	Lim et al. (2021)	Edge homophily with consideration for class imbalance
Adjusted homophily Label informativeness (LI)	Platonov et al. (2022)	Extension of edge homophily satisfying maximal agreement Information quantity provided about a node's label
Probabilistic Bayes Error (PBE) Negative generalized Jeffreys divergence	Luan et al. (2023)	Probability of a node being misclassified Analytic expression for PBE
Normalized total variation (NTV) Normalized smoothness value (NSV)	Luan et al. (2022)	Variation of the graph signal w.r.t. graph filters Effect of the edge bias
Neighbourhood entropy Center-neighbour similarity	Xie et al. (2020)	Similarity of node embeddings within neighbourhoods (Dis)similarity of the neighbours of a node
Aggregation similarity score Diversification distinguishability	Luan et al. (2021)	Proportion of similarity weights after aggregation Proportion of nodes benefitting from diversification operations
Cross-class neighbourhood similarity (CCNS)	Ma et al. (2022)	Similarity measure for neighbourhoods
Label smoothness Feature smoothness	Hou et al. (2019)	Dissimilarity of neighbouring labels Similarity between connected feature vectors

Edge homophily (Zhu et al., 2020) (defined in Section 2.3) is one of the most commonly used metrics known to be correlated with the performance of GNNs. It has been shown that several graph convolutions do not perform well on heterophilic graph structures. However, homophily is not a *necessary* property for successful graph learning and heterophily affects different graph models at different rates (Zhu et al., 2020). Heterophilic graphs can have very different structures, containing either highly informative neighbourhoods or uninformative ones. These different notions of heterophily can be captured by other metrics, such as label informativeness (Platonov et al., 2022) or cross-class neighbourhood similarity (CCNS) (Ma et al., 2022). For that reason, a more general term *node distinguishability (ND)* has been used (Luan et al., 2023), which quantifies the difference in neighbourhoods more generally than heterophily/homophily.

Additional metrics have been introduced that have shown to impact the performance of GNNs (Luan et al., 2021; Ma et al., 2022; Luan et al., 2022). Luan et al. (2022) discuss in which settings GNNs are beneficial and when GNNs under-perform other ML models and introduce two metrics called normalized total variation (NTV) and normalized smoothness value (NSV) to measure the effect of edge bias. Xie et al. (2020) decide when neighbourhood aggregation may be unnecessary by evaluating the so-called neighbourhood entropy and centre-neighbour similarity in graphs. Luan et al. (2021) assess graph statistics with the aid of an aggregation similarity score and a diversification distinguishability metric, mainly capturing the linear relations of the aggregated node features. Ma et al. (2022) investigate the similarity in neighbourhoods of same-labelled nodes and call their metric cross-class neighbourhood similarity (CCNS). Mueller et al. (2023a) reduce the CCNS matrix to a single value they call CCNS distance. Label informativeness (LI) (Platonov et al., 2022) quantifies the information content a neighbour's label provides about the label of a node of interest and therefore gives more insights into different notions of heterophily. Luan et al. (2023) address the concept of node-distinguishability (ND), which is represented by several metrics (Ma et al., 2022; Luan et al., 2023). This concept is more general than homophily versus heterophily. Luan et al. (2023) also extend the idea of homophily by introducing probabilistic Bayes error (PBE) and negative generalised Jeffreys divergence as metrics. PBE describes the "probability of a node being misclassified when the true class probabilities given the predictors are known" (Luan et al., 2023) and the negative generalised Jeffreys divergence is an analytic expression for PBE. Hou et al. (2019) introduce two smoothness metrics: label and feature smoothness. Label smoothness is similar to the homophily metric and feature smoothness quantifies how similar the node features are between connected nodes.

There is no easy answer to what a "good" graph structure looks like. Some graph learning methods are highly sensitive to heterophilic graphs, while others work well on both homophilic and heterophilic graphs (Zhu et al., 2020). Different notions of heterophilic graphs (Platonov et al., 2022) show the complexity of assessing graph structures by their homophily value. We believe that the notion of node distinguishability (Luan et al., 2023) summarises the impact of graph characteristics on GNN performance best.

Given the complexity of assessing a graph structure, we advise using multiple graph assessment metrics – ideally before (and for adaptive methods also during) model training. They can shed light on the composition of the graph and can potentially reveal reasons for poor performance or guide a suitable graph convolution for performing the downstream task. The evaluation of different metrics that assess different qualities of the graph structure can be essential for understanding the complex interplay between graph structure and model performance (Platonov et al., 2022).

Graph assessment beyond supervised node classification While graph assessment metrics have mostly been applied for node-level classification tasks and under supervised training, they are not limited to these settings since they represent general graph properties and can equally be applied to graph-level or edge-level predictions. Homophily has initially been introduced for classification tasks and discrete adjacency matrices but has recently been extended to the notion of node homophily in regression tasks and continuous adjacency matrices, which are required for some adaptive graph learning methods (Mueller et al., 2023a). Furthermore, some metrics have been specifically used for link prediction and unsupervised learning tasks. Li et al. (2022) discuss unsupervised learning for node and edge classification tasks and link prediction, as well as the utilisation of edge labels for edge label assortativity. They observe that edge classification tasks rely more on features of paired nodes having signals with different frequencies. We notice a lack of more detailed and specific graph assessment metrics for tasks beyond node-level classification and identify this as an open and important research area.

5.3 Selection of graph construction methods

We here formulate specific recommendations for selecting a graph construction method for different applications and datasets. We discuss the choice of graph type (population-level, subject-level, or subject-independent graphs) as well as the choice between static and adaptive graph construction.

Choosing a graph type The choice between the different graph types is highly dependent on the dataset as well as the desired downstream task. If a graph structure can be extracted for each subject individually, *subject-level graphs* are of use. Here, no dependencies between the subjects need to exist or be derived. *Population-level graphs* on the other hand aim to represent all subjects in one data structure. Each subject is represented by a feature vector or matrix and additional information is added by connecting the subjects. There are also works studying graph-in-graph problems, where subject-level graphs can be additionally connected to form a population-level graph (Mullakaeva et al., 2022). The authors show that nested graph approaches can improve performance. *Subject-independent graphs* are of interest, when no specific relation to a medical subject is available or relevant. This is for example the case when performing link prediction in knowledge graphs, where the relationships between entities (such as diseases and symptoms) are of interest and there is no personal data of patients.

Choosing a static or adaptive graph construction method The choice between a *static* or *adaptive* graph construction method is more challenging and depends both on the application and its goal. Knowledge graphs are typically created statically, since here, the connections between entities (e.g. symptoms and diseases) are based on literature and probably do not benefit from end-to-end learning. The same holds for molecules. Here, the graph structure is intrinsic to the data, and it would be unintuitive to change the edges (i.e. chemical bonds) of molecules. On the other hand, population-level graphs and some subject-level graphs have shown to benefit from an adaptive graph construction (Kazi et al., 2022; Campbell et al., 2022; Kan et al., 2022). They can, for example, be used to incorporate a temporal component into the learning process, like with fMRI data. It can also be a suitable method to altogether avoid the necessity of picking the *right* graph creation method before training since no specific metric or feature selection is required in this case. The main advantages and shortcomings of adaptive and static graph creation methods are summarised in Section 5.1. We believe that choosing an adaptive graph construction method is especially beneficial in settings like population graphs, where neither an initial graph structure nor a ground truth is available. We believe that this is a promising area for future research. Especially, the possibility of using the learned graphs to gain insight into the dataset at hand for interpretability purposes shows high potential for further investigation.

The most commonly used method to generate population-level graphs is based on similarity (Parisot et al., 2017) or distance (e.g. Euclidean) (Lu et al., 2022). In both settings, a subset of the available features can be used to define the edges of the graph (e.g. non-imaging features), while the remaining features (e.g. imaging features) can be used as node features (Parisot et al., 2017). This –in itself– implements an opportunity for multi-modal data integration. Later, it has also been shown that using all available features for both, creating the edges and the node features might be beneficial over splitting the features up (Keicher et al., 2021). The most frequently used approach for generating brain graphs is based on correlation, and graphs from curvilinear structures are mostly extracted based on the underlying tree structure of the vessels/airways.

Regardless of the graph construction method of choice, we highly recommend evaluating graph assessment metrics to gain insights into the graph structure and potentially evaluate whether it is beneficial to the learning task or might hinder GNN performance. This is especially useful when using static graph construction methods so that the graph structure can be judged prior to training.

5.4 Selection of Graph Learning Method

The graph construction method directly impacts the graph structure and therefore the performance of the utilised GNN for the downstream task. We, therefore, here discuss the interplay between the two components. The graph construction method can be either independent of the graph learning pipeline (static graph construction) or intertwined (adaptive graph construction). In both cases, the choice of graph convolution is an important factor. Zhu et al. (2020) identify some critical design choices that can improve the performance of GNNs on heterophilous graphs. They show that a separate embedding of neighbourhood node features and node-internal features improves the performance of heterophilous graphs as well as a separate embedding of higher-order neighbourhoods and introduces a new model architecture that works well on both homophilous and heterophilous graphs. Graph convolutions that propagate information simultaneously for all neighbours and their own node features are more impacted by heterophilic graphs than convolutions that have separate message-passing schemes for the neighbourhoods and their own node features. There are several graph convolutions, that have been specifically designed for low-homophily graphs, such as H2GCN (Zhu et al., 2020), HEAT convolutions (Mo et al., 2021b), or heterogeneous graph transformers (Hu et al., 2020b). In case the constructed graph results in a low-homophily graph, it is advisable to select one of these graph convolutions for graph learning. However, the choice of such graph convolutions does not guarantee high performance. We furthermore note that graph convolutions that are highly impacted by the graph structure (e.g. GCN (Kipf & Welling, 2016)) benefit most from the utilisation of adaptive graph construction methods. In these cases, the graph structure can be optimised for the whole graph learning pipeline, including the graph convolution at hand. Different models have also been specifically designed for specific tasks. RotatE (Sun et al., 2018), TransE (Bordes et al., 2013), and PairRE (Chao et al., 2021) have, for example, been designed for link prediction tasks and applied for knowledge graph completion. We furthermore identify a systematic evaluation of the connection between model architectures and graph structures as an open research question.

6 Related work

In this section, we put our work in context with existing research. Table 5 summarises adjacent works in similar fields and more information about GNNs and their application in different areas of medicine.

Table 5: **Summary of recent survey papers** in connected areas with their specific application areas, the year of publication, and the reviewed methods. *Categorisation methodology* lists the main categorisation schemes followed in the respective works. GSP refers to graph signal processing, and AE to autoencoders.

Domain	**Reference**	**Year**	**Reviewed methods**	**Categorisation methodology**
Medical	Ahmedt-Aristizabal et al. (2021)	2021	GNNs	Model architecture, application
Medical imaging	Ding et al. (2022)	2022	GNNs	medical image type
Bioinformatics	Yi et al. (2022)	2021	Graph embedding, GNNs	Algorithm, application levels
Bioinformatics	Zhang et al. (2021a)	2020	GNNs	Application, prediction task
Neuroscience	Bessadok et al. (2022)	2020	GNNs	Application, loss function Architecture, graph creation
Biological data	Li et al. (2021a)	2022	GNNs, GSP	Application, feature extraction
Medical	Ours	2023	GNNs	Graph construction, graph type
Knowledge graphs	Ye et al. (2022)	2022	GNNs	Task, application
Recommender systems	Wu et al. (2022)	2022	GNNs	Type of information used
Fault analysis	Chen et al. (2021)	2021	GNNs	Application, data representation
Finance	Wang et al. (2021)	2021	GNNs	Graph construction, feature extraction
Text classification	Pham et al. (2022)	2022	GNNs, RNNs CNN, AE	Text to graph transformation, Model architectures
Domain invariant	Zhu et al. (2021)	2022	GNNs	Graph structure learning type

Reviews on GNNs on medical data Ahmedt-Aristizabal et al. (Ahmedt-Aristizabal et al., 2021) give a detailed overview of GNN training on medical data, organised by application type and graph architectures. The authors investigate different graph creation methods, however, putting them mostly into context with varying application areas. They specifically name graph creation as one of the open challenges in research in this area. For an overview of works covering GNNs, different graph embedding techniques (homogeneous, heterogeneous, and attributed), and generative graph models, we refer to Yi et al. (2022), who target works in bioinformatics. Zhang et al. (2021a) focus on GNNs in bioinformatics, and Bessadok et al. (2022) give an overview of GNNs in network neuroscience. The latter also categorise existing works by graph creation methods; however only refer to research in neuroscience, and summarise different loss functions for different applications. For a review of graph signal processing and graph learning on biological data, we refer to Li et al. (2021a). The authors also provide a summary of feature extraction techniques in the graph domain. Ding et al. (2022) give an overview of applications of GNN methods to multi-modal medical imaging datasets.

Graph construction in other domains Graph construction can be a challenging aspect of graph deep learning pipelines, regardless of the domain and application. Therefore, we list some works investigating graph creation methods in different domains.

For *recommender systems*, Wu et al. (2022) provide an extensive survey on GNN applications, including an elaboration of the graph creation methods in this area. However, they only focus on the post-processing of the graph structure before training since the initial graph structure is usually always provided by the bipartite user-item graph - which differs from several medical datasets. Han et al. (2022) highlight the difficulties of creating a graph for *retrosynthetic planning* with GNNs and propose a semi-dynamic approach for connecting different molecules with each other. Different graph construction methods for association graphs for *fault analysis* are discussed in Chen et al. (2021). Here, the authors categorise the graph construction methods into (a) using the k-nearest neighbour method, (b) using prior knowledge, and (c) using matrix completion. This categorisation is similar to the one we use in this review. However, we add the additional component of adaptive graph construction. For a review on GNNs in *finance*, including different graph construction methods, we refer to Wang et al. (2021). They distinguish three graph creation methods: data-based, knowledge-based, and similarity-based. This categorisation is similar to the one utilised in our work. However, we provide a more detailed methodological distinction between the graph creation methods. Pham et al. (2022) give an overview of graph creation methods for text analysis, categorising existing works based on text graphs and model architectures. In Zhu et al. (2021), the authors investigate general methods for graph structure learning, limited to the here termed "adaptive" graph creation methods.

7 Open challenges and future directions

In this section, we discuss open challenges that arise when working with GNNs in medicine that can be linked to graph construction methods and graph learning in general.

7.1 GNNs vs. graph agnostic models

As mentioned in Sections 1.1 and 5.2, recent works have shown that GNNs do not consistently outperform graph agnostic models (Luan et al., 2022). In most of the here summarised works, the original data does not come naturally in graph structure. Therefore, one could raise the question of when it is beneficial to construct a graph and perform graph learning at all. Most methodological works show that GNNs lead to improved performances compared to graph-agnostic baselines. However, sometimes GNNs show only little or no improvement over other ML methods. We see a detailed investigation of these connections to be a highly relevant area of future work.

7.2 Bias introduction

Another challenge that arises when choosing a graph creation method is the introduction of bias (Mehrabi et al., 2021). When creating a graph structure statically, every design choice (e.g. which features to select for graph creation and which to use as node features, the incorporation of prior knowledge) introduces (human) bias to the resulting graph structure. Sohn et al. (2015), for example, discuss the impact of the selection of ROIs for brain graph creation from fMRI data and argue that individual ROI selections would be more accurate. When using purely adaptive graph creation methods, no human bias will be introduced into the graph creation method. Here, the bias in the dataset will dominate the graph creation. If the available data is highly imbalanced, questions of fairness might arise. This is something we believe should be considered when selecting a graph creation method, and we believe this to be an interesting area of research.

7.3 Undirected vs. directed graphs

Most approaches discussed in this work have been applied to undirected graphs, where edges are always bi-directional and only a few works discuss methods that work on directed graphs (Keicher et al., 2021). The effects of using directed versus undirected graphs have not been studied in detail. Changing an undirected graph to a directed one could be an extension of some of the here mentioned methods for graph construction.

7.4 Availability of public datasets

One significant challenge in exploring graph construction methods on medical data is the shortage of publicly available datasets. For population graphs, two commonly used public datasets are TADPOLE (Yu et al., 2020), a subset of the ADNI dataset, and the autism brain imaging data exchange (ABIDE) (Di Martino et al., 2014), which address Alzheimer's disease and autism detection, respectively. The private dataset UK Biobank (Sudlow et al., 2015) is also frequently used. The open graph benchmark (Hu et al., 2020a) holds a set of benchmark datasets for graph learning. It also contains molecule datasets and a vessel graph dataset from Paetzold et al. (2021). For the construction of brain connectivity graphs, the human connectome project (HCP) (Elam et al., 2021) and the brain connectivity challenge dataset[1] have been used apart from several private datasets. For surface mesh representations, MedShapeNet Scharinger et al. (2023) offers a variety of medical mesh structures.

7.5 Data privacy

Medical research is usually performed on private data such as medical images or patient records, which hold highly sensitive information about patients. A well-established, formal method to allow DL while giving privacy guarantees to individuals is Differential Privacy (DP) (Dwork et al., 2014). It ensures that the output of a randomised algorithm is approximately the same, independent of a single data point being in the dataset or not. It has been shown that guaranteeing DP in GNNs is more challenging than in tabular data since the individual data points (nodes) in a graph are inter-connected (Mueller et al., 2023b). Furthermore, graph-structured data is more sensitive to privacy attacks like Membership Inference Attacks (MIAs), which aim to identify whether a certain data point was part of the training dataset. This is because the additional information lies in the graph structure itself and can be leveraged by adversaries (Olatunji et al., 2021). We see a requirement for the development of high-utility privacy-preserving techniques for graph learning in medicine.

8 Conclusion

This in-depth review of state-of-the-art works on graph creation methods for medical data shows that graph construction is challenging and requires various design choices as well as a careful consideration of the dataset and task at hand. There are numerous ways to construct a suitable graph structure from a dataset and the best method needs to be selected with caution. We categorise graph construction methods by *static* or *adaptive* approaches. Static graph construction methods generate a graph structure prior to learning, whereas adaptive methods change the graph throughout GNN training. We analyse advantages and disadvantages of both approaches and formulate recommendations about how to pick a suitable graph construction method.

So far, adaptive methods have only been applied to a small subset of graph-learning tasks in medicine. We believe that this will be explored further in the coming years. Especially, the post-hoc interpretability of the learned adjacency matrix can hold valuable information about the dataset and task, which might not be possible to extract from a statically generated graph.

Given the findings about the strong impact of the graph structure on GNN performance (Section 1.1 and 5.2), we want to raise awareness that in cases where the graph structure is not clearly defined by the dataset,

[1] miccai.brainconnectivity.net

it is important to consider different graph creation methods as well as suitable graph convolutions. One characteristic of static graph creation methods is that the graph structure is defined before training. This opens up the possibility of evaluating the graph structure based on the metrics summarised in Section 5.2. We believe this to be a valuable step in evaluating the generated graph structure. This way, for example, the homophily of the graph can be analysed and the potential need for graph convolutions that can handle heterophilous graph structures can be evaluated (Du et al., 2022; Zhu et al., 2020). This could lead to graph learning methods that are more linked to the constructed/available graph structures. These graph metrics could also potentially be incorporated into the end-to-end learning of adaptive graph structures, where they could be added to the loss function to push the generated graph in the direction of a specific metric.

Given the various options to construct graphs from medical data, we think that explicit explanations of the applied graph creation method in each work would be beneficial to the research community. Even though the current state-of-the-art results summarised in this work indicate several recommendations for graph-creation methods on medical data, we believe that there are still a lot of open questions regarding which graph-creation method works best for which application. Furthermore, we consider a more thorough investigation of why a certain method might be beneficial over others to be necessary to learn more about the fundamental methodologies of GNNs and how they can be applied in the most favourable ways. We hope to contribute to further research in this area with this survey.

References

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors*, 21(14): 4758, 2021.

N Assempour, I Iynkkaran, YF Liu, A Maciejewski, N Gale, A Wilson, L Chin, R Cummings, D Le, A Pon, et al. 5.0: a major update to the drugbank database for 2018. *Nucleic Acids Res*, 46:D1074–D1082, 2018.

Sorin Avram, Cristian G Bologa, Jayme Holmes, Giovanni Bocci, Thomas B Wilson, Dac-Trung Nguyen, Ramona Curpan, Liliana Halip, Alina Bora, Jeremy J Yang, et al. Drugcentral 2021 supports drug discovery and repositioning. *Nucleic acids research*, 49(D1):D1160–D1169, 2021.

Emanuel A Azcona, Pierre Besson, Yunan Wu, Arjun Punjabi, Adam Martersteck, Amil Dravid, Todd B Parrish, S Kathleen Bandt, and Aggelos K Katsaggelos. Interpretation of brain morphology in association to alzheimer's disease dementia classification using graph convolutional networks on triangulated meshes. In *Shape in Medical Imaging: International Workshop, ShapeMI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings*, pp. 95–107. Springer, 2020.

Pietro Barbiero, Ramon Vinas Torne, and Pietro Lió. Graph representation forecasting of patient's medical conditions: Toward a digital twin. *Frontiers in genetics*, 12:652907, 2021.

Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Graph neural networks in network neuroscience. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2022.

Wendong Bi, Lun Du, Qiang Fu, Yanlin Wang, Shi Han, and Dongmei Zhang. Make heterophily graphs better fit gnn: A graph rewiring approach. *arXiv preprint arXiv:2209.08264*, 2022.

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Andreas Bender, Charles Tapley Hoyt, and William L Hamilton. A review of biomedical datasets relating to drug discovery: a knowledge graph perspective. *Briefings in Bioinformatics*, 23(6), 2022.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2787–2795, 2013.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Alexander Campbell, Antonio Giuliano Zippo, Luca Passamonti, Nicola Toschi, and Pietro Lio. DBGSL: Dynamic brain graph structure learning. *arXiv:2209.13513*, 2022.

Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023.

Chun-Hung Chao, Zhuotun Zhu, Dazhou Guo, Ke Yan, Tsung-Ying Ho, Jinzheng Cai, Adam P Harrison, Xianghua Ye, Jing Xiao, Alan Yuille, et al. Lymph node gross tumor volume detection in oncology imaging via relationship learning using graph neural network. In *MICCAI*, pp. 772–782, 2020.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4360–4369, 2021.

Lang Chen, Yangmin Huang, Bin Liao, Kun Nie, Shoubin Dong, and Jinlong Hu. Graph learning approaches for graph with noise: application to disease prediction in population graph. In *BIBM*, pp. 2724–2729. IEEE, 2020a.

Li Chen, Thomas Hatsukami, Jenq-Neng Hwang, and Chun Yuan. Automated intracranial artery labeling using a graph neural network and hierarchical refinement. In *MICCAI*, pp. 76–85, 2020b.

Zhiwen Chen, Jiamin Xu, Cesare Alippi, Steven X Ding, Yuri Shardt, Tao Peng, and Chunhua Yang. Graph neural network-based fault diagnosis: a review. *arXiv:2111.08185*, 2021.

Binjie Cheng, Jin Zhang, Hong Liu, Meiling Cai, and Ying Wang. Research on medical knowledge graph for stroke. *Journal of Healthcare Engineering*, 2021, 2021a.

Yinlin Cheng, Mengnan Ma, Xingyu Li, and Yi Zhou. Multi-label classification of fundus images based on graph convolutional network. *BMC Medical Informatics and Decision Making*, 21(2):1–9, 2021b.

Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In *MICCAI*, pp. 643–653, 2020.

Ameya Daigavane, Balaraman Ravindran, and Gaurav Aggarwal. Understanding convolutions on graphs. *Distill*, 2021. doi: 10.23915/distill.00032.

Haitz Sáez de Ocáriz Borde, Anees Kazi, Federico Barbero, and Pietro Lio. Latent graph inference using product manifolds. In *ICLR*, 2023.

Swakshar Deb, Md Fokhrul Islam, Shafin Rahman, and Sejuti Rahman. Graph convolutional networks for assessment of physical rehabilitation exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:410–419, 2022.

Andac Demir, Toshiaki Koike-Akino, Ye Wang, Masaki Haruna, and Deniz Erdogmus. EEG-GNN: Graph neural networks for classification of electroencephalogram (eeg) signals. In *EMBC*, pp. 1061–1067, 2021.

Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.

Alexander Dilthey, Charles Cox, Zamin Iqbal, Matthew R Nelson, and Gil McVean. Improved genome inference in the mhc using a population reference graph. *Nature genetics*, 47(6):682–688, 2015.

Kexin Ding, Mu Zhou, Zichen Wang, Qiao Liu, Corey W Arnold, Shaoting Zhang, and Dimitri N Metaxas. Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications. *arXiv:2202.08916*, 2022.

Dominik Drees, Aaron Scherzinger, René Hägerling, Friedemann Kiefer, and Xiaoyi Jiang. Scalable robust graph and feature extraction for arbitrary vessel networks in large volumetric datasets. *BMC bioinformatics*, 22(1):1–28, 2021.

Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. GBK-GNN: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *WWW*, pp. 1550–1558, 2022.

Shaoyi Du, Yanrong Guo, Gerard Sanroma, Dong Ni, Guorong Wu, and Dinggang Shen. Building dynamic population graph for accurate correspondence detection. *Medical image analysis*, 26(1):256–267, 2015.

Min Duan, Tao Tan, Binzhe Zhang, and Xun Zhou. A prediction of the paths to equalization of basic public health services based on gragh neural network (GNN). 2022.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

Ahmed El-Gazzar, Rajat Mani Thomas, and Guido van Wingen. Dynamic adaptive spatio-temporal graph convolution for fMRI modelling. In *MLCN@MICCAI*, pp. 125–134, 2021.

Yassine El Ouahidi, Hugo Tessier, Giulia Lioi, Nicolas Farrugia, Bastien Pasdeloup, and Vincent Gripon. Pruning graph convolutional networks to select meaningful graph frequencies for fMRI decoding. In *EUSIPCO*, pp. 937–941. IEEE, 2022.

Jennifer Stine Elam, Matthew F Glasser, Michael P Harms, Stamatios N Sotiropoulos, Jesper LR Andersson, Gregory C Burgess, Sandra W Curtiss, Robert Oostenveld, Linda J Larson-Prior, Jan-Mathijs Schoffelen, et al. The human connectome project: a retrospective. *NeuroImage*, 244:118543, 2021.

Ahmed ElGazzar, Rajat Thomas, and Guido Van Wingen. Benchmarking graph neural networks for fmri analysis. *arXiv:2211.08927*, 2022.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *WWW*, pp. 417–426, 2019.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Bruce Fischl. Freesurfer. *NeuroImage*, 62(2):774–781, 2012. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2012.01.021. URL `https://www.sciencedirect.com/science/article/pii/S1053811912000389`. 20 YEARS OF fMRI.

Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, volume 6, pp. 281–305. Interlaken, 1987.

Jianliang Gao, Tengfei Lyu, Fan Xiong, Jianxin Wang, Weimao Ke, and Zhao Li. Predicting the survival of cancer patients with multimodal graph neural network. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 19(2):699–709, 2021.

Antonio Garcia-Uceda Juarez, Raghavendra Selvan, Zaigham Saghir, and Marleen de Bruijne. A joint 3d unet-graph neural network-based method for airway segmentation from chest cts. In *MLMI@MICCAI*, pp. 583–591, 2019.

Mohammed Amine Gharsallaoui, Furkan Tornaci, and Islem Rekik. Investigating and quantifying the reproducibility of graph neural networks in predictive medicine. In *PRIME@MICCAI*, pp. 104–116, 2021.

Mahsa Ghorbani, Mojtaba Bahrami, Anees Kazi, Mahdieh Soleymani Baghshah, Hamid R Rabiee, and Nassir Navab. Gkd: Semi-supervised graph knowledge distillation for graph-independent inference. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24*, pp. 709–718. Springer, 2021.

Mahsa Ghorbani, Anees Kazi, Mahdieh Soleymani Baghshah, Hamid R Rabiee, and Nassir Navab. RA-GCN: Graph convolutional network for disease prediction problems with imbalanced data. *Medical Image Analysis*, 75:102272, 2022.

Karthik Gopinath, Christian Desrosiers, and Herve Lombaert. Graph convolutions on spectral embeddings for cortical surface parcellation. *Medical Image Analysis*, 54:297–305, 2019a. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.03.012. URL `https://www.sciencedirect.com/science/article/pii/S1361841518305243`.

Karthik Gopinath, Christian Desrosiers, and Herve Lombaert. Adaptive graph convolution pooling for brain surface analysis. In Albert C. S. Chung, James C. Gee, Paul A. Yushkevich, and Siqi Bao (eds.), *Information Processing in Medical Imaging*, pp. 86–98, Cham, 2019b. Springer International Publishing.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proc. 2005 IEEE Int. joint conf. on neural networks*, volume 2(2005), pp. 729–734, 2005.

Peng Han, Peilin Zhao, Chan Lu, Junzhou Huang, Jiaxiang Wu, Shuo Shang, Bin Yao, and Xiangliang Zhang. Gnn-retro: Retrosynthetic planning with graph neural networks. In *AAAI*, volume 36(4), pp. 4014–4021, 2022.

Si Hang. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw*, 41(2):11, 2015.

Lasse Hansen and Mattias P Heinrich. GraphRegNet: Deep graph regularisation networks on sparse keypoints for dense registration of 3D lung CTs. *IEEE Trans. on Medical Imaging*, 40(9):2246–2257, 2021.

Junheng Hao, Chuan Lei, Vasilis Efthymiou, Abdul Quamar, Fatma Özcan, Yizhou Sun, and Wei Wang. Medto: Medical data to ontology matching using hybrid graph neural networks. In *Proc. of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining*, pp. 2946–2954, 2021.

Yefei He, Tao Yang, Cheng Yang, and Hong Zhou. Integrated equipment for parkinson's disease early detection using graph convolution network. *Electronics*, 11(7):1154, 2022.

Mattias P Heinrich, Heinz Handels, and Ivor JA Simpson. Estimating large lung motion in COPD patients by symmetric regularised correspondence fields. In *MICCAI*, pp. 338–345, 2015.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37, 2021.

Daibing Hou, Zijian Zhao, and Sanyuan Hu. Multi-label learning with visual-semantic embedded knowledge graph for diagnosis of radiology imaging. *IEEE Access*, 9:15720–15730, 2021.

Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard TB Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks. In *International Conference on Learning Representations*, 2019.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020b.

Jiahao Huang, Angelica I Aviles-Rivero, Carola-Bibiane Schönlieb, and Guang Yang. Vigu: Vision gnn u-net for fast mri. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2023.

Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. *Advances in neural information processing systems*, 31, 2018.

Yongxiang Huang and Albert CS Chung. Edge-variational graph convolutional networks for uncertainty-aware disease prediction. In *MICCAI*, pp. 562–572, 2020.

Ruidong Jin, Tianqi Xia, Xin Liu, Tsuyoshi Murata, and Kyoung-Sook Kim. Predicting emergency medical service demand with bipartite graph convolutional networks. *Ieee Access*, 9:9903–9915, 2021.

Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. FBNETGEN: Task-aware gnn-based fMRI analysis via functional brain network generation. In *MIDL*, 2022.

Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. InceptionGCN: receptive field aware graph convolutional network for disease prediction. In *Proc. IPMI 2019, June 2–7*, pp. 73–85. Springer, 2019.

Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael M Bronstein. Differentiable graph module (DGM) for graph convolutional networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 45(2):1606–1617, 2022.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30:595–608, 2016.

Matthias Keicher, Hendrik Burwinkel, David Bani-Harouni, Magdalini Paschali, Tobias Czempiel, Egon Burian, Marcus R Makowski, Rickmer Braren, Nassir Navab, and Thomas Wendler. U-gat: Multimodal graph attention network for covid-19 outcome prediction. *arXiv:2108.00860*, 2021.

Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. *Advances in Neural Information Processing Systems*, 34:4314–4327, 2021.

Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. In *ICLR*, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.

Dominik Klepl, Fei He, Min Wu, Daniel J Blackburn, and Ptolemaios Sarrigiannis. EEG-based graph neural network classification of alzheimer's disease: An empirical evaluation of functional connectivity methods. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 30:2651–2660, 2022.

Youyong Kong, Shuwen Gao, Yingying Yue, Zhenhua Hou, Huazhong Shu, Chunming Xie, Zhijun Zhang, and Yonggui Yuan. Spatio-temporal graph convolutional network for diagnosis and treatment response prediction of major depressive disorder from functional connectivity. *Human brain mapping*, 42(12):3922–3933, 2021.

Youyong Kong, Shuyi Niu, Heren Gao, Yingying Yue, Huazhong Shu, Chunming Xie, Zhijun Zhang, and Yonggui Yuan. Multi-stage graph fusion networks for major depressive disorder diagnosis. *IEEE Trans. on Affective Computing*, 13(4):1917–1928, 2022.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *ICML*, pp. 3499–3508, 2019.

Rui Li, Xin Yuan, Mohsen Radfar, Peter Marendy, Wei Ni, Terence J O'Brien, and Pablo M Casillas-Espinosa. Graph signal processing, graph neural network and graph learning on biological data: a systematic review. *IEEE Reviews in Biomedical Engineering*, 2021a.

Ruikun Li, Yi-Jie Huang, Huai Chen, Xiaoqing Liu, Yizhou Yu, Dahong Qian, and Lisheng Wang. 3d graph-connectivity constrained network for hepatic vessel segmentation. *IEEE Journal of Biomedical and Health Informatics*, 26(3):1251–1262, 2021b.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Proc. of the AAAI conference on artificial intelligence*, volume 32(1), 2018.

Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021c.

You Li, Bei Lin, Binli Luo, and Ning Gui. Graph representation learning beyond node and homophily. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4880–4893, 2022.

Derek Lim, Xiuyu Li, Felix Hohne, and Ser-Nam Lim. New benchmarks for learning on non-homophilous graphs. *arXiv:2104.01404*, 2021.

Lan Lin, Min Xiong, Ge Zhang, Wenjie Kang, Shen Sun, Shuicai Wu, and Initiative Alzheimer's Disease Neuroimaging. A convolutional neural network and graph convolutional network based framework for ad classification. *Sensors*, 23(4), 2023. ISSN 1424-8220. doi: 10.3390/s23041914. URL `https://www.mdpi.com/1424-8220/23/4/1914`.

Fenglin Liu, Chenyu You, Xian Wu, Shen Ge, Xu Sun, et al. Auto-encoding knowledge graph for unsupervised medical report generation. *Advances in Neural Information Processing Systems*, 34:16266–16279, 2021.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Siyuan Lu, Ziquan Zhu, Juan Manuel Gorriz, Shui-Hua Wang, and Yu-Dong Zhang. NAGNN: classification of covid-19 based on neighboring aware representation from deep graph neural network. *Int. Journal of Intelligent Systems*, 37(2):1572–1598, 2022.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv:2109.05641*, 2021.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, and Doina Precup. When do we need gnn for node classification? *arXiv:2210.16979*, 2022.

Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. When do graph neural networks help with node classification: Investigating the homophily principle on node distinguishability. *arXiv preprint arXiv:2304.14274*, 2023.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *ICLR*, 2022.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. Entrez gene: gene-centered information at NCBI. *Nucleic acids research*, 39(suppl_1):D52–D57, 2010.

Dwarikanath Mahapatra, Steven Korevaar, Behzad Bozorgtabar, and Ruwan Tennakoon. Unsupervised domain adaptation using feature disentanglement and gcns for medical image classification. In *ECCV Workshops*, pp. 735–748, 2022.

Ines Mahjoub, Mohamed Ali Mahjoub, and Islem Rekik. Brain multiplexes reveal morphological connectional biomarkers fingerprinting late brain dementia states. *Scientific reports*, 8(1):1–14, 2018.

Usman Mahmood, Zening Fu, Vince D Calhoun, and Sergey Plis. A deep learning model for data-driven discovery of functional connectivity. *Algorithms*, 14(3):75, 2021.

Chengsheng Mao, Liang Yao, and Yuan Luo. MedGCN: Medication recommendation and lab test imputation via graph convolutional networks. *Journal of Biomedical Informatics*, 127:104000, 2022.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi: 10.1146/annurev.soc.27.1.415. URL `https://doi.org/10.1146/annurev.soc.27.1.415`.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Shaocong Mo, Ming Cai, Lanfen Lin, Ruofeng Tong, Qingqing Chen, Fang Wang, Hongjie Hu, Yutaro Iwamoto, Xian-Hua Han, and Yen-Wei Chen. Mutual information-based graph co-attention networks for multimodal prior-guided magnetic resonance imaging segmentation. *IEEE Trans. on Circuits and Systems for Video Technology*, 32(5):2512–2526, 2021a.

Xiaoyu Mo, Yang Xing, and Chen Lv. Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction. *CoRR*, abs/2106.07161, 2021b. URL `https://arxiv.org/abs/2106.07161`.

José Teófilo Moreira-Filho, Meryck Felipe Brito da Silva, Joyce Villa Verde Bastos Borba, Arlindo Rodrigues Galvão Filho, Eugene Muratov, Carolina Horta Andrade, Rodolpho de Campos Braga, and Bruno Junior Neves. Artificial intelligence systems for the design of magic shotgun drugs. *Artificial Intelligence in the Life Sciences*, pp. 100055, 2022.

Tamara T Mueller, Johannes C Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, and Georgios Kaissis. Differentially private graph classification with gnns. *arXiv:2202.02575*, 2022a.

Tamara T. Mueller, Johannes C. Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, and Georgios Kaissis. Differentially private graph neural networks for whole-graph classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7308–7318, dec 2022b. ISSN 0162-8828. doi: 10.1109/TPAMI.2022.3228315. URL `https://doi.org/10.1109/TPAMI.2022.3228315`.

Tamara T Mueller, Sophie Starck, Leonhard F Feiner, Kyriaki-Margarita Bintsi, Daniel Rueckert, and Georgios Kaissis. Extended graph assessment metrics for graph neural networks. *arXiv preprint arXiv:2307.10112*, 2023a.

Tamara T Mueller, Dmitrii Usynin, Johannes C Paetzold, Daniel Rueckert, and Georgios Kaissis. Differential privacy guarantees for analytics and machine learning on graphs: A survey of results. *Journal of Privacy and Confidentiality (Accepted)*, 2023b.

Tamara T. Mueller, Siyu Zhou, Sophie Starck, Friederike Jungmann, Alexander Ziller, Orhun Aksoy, Danylo Movchan, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert. Body fat estimation from surface meshes using graph neural networks. In Christian Wachinger, Beatriz Paniagua, Shireen Elhabian, Jianning Li, and Jan Egger (eds.), *Shape in Medical Imaging*, pp. 105–117, Cham, 2023c. Springer Nature Switzerland. ISBN 978-3-031-46914-5.

Kamilia Mullakaeva, Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael M Bronstein. Graph-in-graph (GiG): Learning interpretable latent graphs in non-euclidean domain for biological and healthcare applications. *arXiv:2204.00323*, 2022.

Fuad Noman, Chee-Ming Ting, Hakmook Kang, Raphael C-W Phan, Brian D Boyd, Warren D Taylor, and Hernando Ombao. Graph autoencoders for embedding learning in brain networks and major depressive disorder identification. *arXiv:2107.12838*, 2021.

Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. In *TPS-ISA*, pp. 11–20, 2021.

Johannes C Paetzold, Julian McGinnis, Suprosanna Shit, Ivan Ezhov, Paul Büschl, Chinmay Prabhakar, Anjany Sekuboyina, Mihail Todorov, Georgios Kaissis, Ali Ertürk, et al. Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience. In *NeurIPS Datasets and Benchmarks*, 2021.

Li Pan, Jundong Liu, Mingqin Shi, Chi Wah Wong, and Kei Hang Katie Chan. Identifying autism spectrum disorder based on individual-aware down-sampling and multi-modal learning. *arXiv:2109.09129*, 2021.

Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. Spectral graph convolutions for population-based disease prediction. In *MICCAI*, pp. 177–185, 2017.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *ICLR*, 2020.

Chantal Pellegrini, Nassir Navab, and Anees Kazi. Unsupervised pre-training of graph transformers on patient population graphs. *arXiv:2207.10603*, 2022.

Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Trans. on Medical Imaging*, 2022.

Bastian Pfeifer, Afan Secic, Anna Saranti, and Andreas Holzinger. GNN-SubNet: disease subnetwork detection with explainable graph neural networks. *bioRxiv*, 2022.

Phu Pham, Loan TT Nguyen, Witold Pedrycz, and Bay Vo. Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artificial Intelligence Review*, pp. 1–35, 2022.

Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. Characterizing graph datasets for node classification: Beyond homophily-heterophily dichotomy. *arXiv preprint arXiv:2209.06177*, 2022.

Yali Qiu, Shuangzhi Yu, Yanhong Zhou, Dongdong Liu, Xuegang Song, Tianfu Wang, and Baiying Lei. Multi-channel sparse graph transformer network for early alzheimer's disease identification. In *ISBI*, pp. 1794–1797. IEEE, 2021.

Gang Qu, Li Xiao, Wenxing Hu, Junqi Wang, Kun Zhang, Vince D Calhoun, and Yu-Ping Wang. Ensemble manifold regularized multi-modal graph convolutional network for cognitive ability prediction. *IEEE Trans. on Biomedical Engineering*, 68(12):3564–3573, 2021.

Zarina Rakhimberdina, Xin Liu, and Tsuyoshi Murata. Population graph-based multi-model ensemble method for diagnosing autism spectrum disorder. *Sensors*, 20(21):6001, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pp. 234–241, 2015.

Chunyang Ruan, Yingpei Wu, Yun Yang, and Guangsheng Luo. Semantic-aware graph convolutional networks for clinical auxiliary diagnosis and treatment of traditional chinese medicine. *IEEE Access*, 9: 8797–8807, 2021.

Yasmin Salehi and Dennis Giannacopoulos. Physgnn: A physics–driven graph neural network based model for predicting soft tissue deformation in image-guided neurosurgery. *Advances in Neural Information Processing Systems*, 35:37282–37296, 2022.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE trans. on neural networks*, 20(1):61–80, 2008.

Bernhard Scharinger, Antonio Pepe, Yuan Jin, Christina Gsaxner, Jianning Li, and Jan Egger. Multicenter aortic vessel tree extraction using deep learning. In *Medical Imaging 2023: Biomedical Applications in Molecular, Structural, and Functional Imaging*, volume 12468, pp. 341–347. SPIE, 2023.

Roman Schulte-Sasse, Stefan Budach, Denes Hnisz, and Annalisa Marsico. Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nature Machine Intelligence*, 3(6):513–526, 2021.

Raghavendra Selvan, Thomas Kipf, Max Welling, Antonio Garcia-Uceda Juarez, Jesper H Pedersen, Jens Petersen, and Marleen de Bruijne. Graph refinement based airway extraction using mean-field networks and graph neural networks. *Medical image analysis*, 64:101751, 2020.

Seung Yeon Shin, Soochahn Lee, Il Dong Yun, and Kyoung Mu Lee. Deep vessel segmentation by learning graphical connectivity. *Medical image analysis*, 58:101556, 2019.

William Sohn, Kwangsun Yoo, Young-Beom Lee, Sang Seo, Duk Na, and Yong Jeong. Influence of ROI selection on resting state functional connectivity: an individualized approach for resting state fmri analysis. *Frontiers in Neuroscience*, 9, 2015. doi: 10.3389/fnins.2015.00280. URL `https://www.frontiersin.org/articles/10.3389/fnins.2015.00280`.

Xiaofan Song, Mingyi Mao, and Xiaohua Qian. Auto-metric graph neural network based on a meta-learning strategy for the diagnosis of alzheimer's disease. *IEEE Journal of Biomedical and Health Informatics*, 25 (8):3141–3152, 2021.

Kamilė Stankevičiūtė, Tiago Azevedo, Alexander Campbell, Richard Bethlehem, and Pietro Liò. Population graph gnns for brain age prediction. *bioRxiv*, 2020.

Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, 2015.

Li Sun, Ke Yu, and Kayhan Batmanghelich. Context matters: Graph-based self-supervised representation learning for medical images. In *Proc. AAAI*, volume 35(6), pp. 4874–4882, 2021.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018.

Zimeng Tan, Jianjiang Feng, and Jie Zhou. Sgnet: Structure-aware graph-based network for airway semantic segmentation. In *MICCAI*, pp. 153–163, 2021.

Dinh V Tran, Nicolò Navarin, and Alessandro Sperduti. On filter size in graph convolutional networks. In *SSCI*, pp. 1534–1541. IEEE, 2018.

Petar Velickovic, Lars Buesing, Matthew C Overlan, Razvan Pascanu, Oriol Vinyals, and Charles Blundell. Pointer graph networks. *stat*, 1050:11, 2020.

Gerome Vivar, Anees Kazi, Hendrik Burwinkel, Andreas Zwergal, Nassir Navab, Seyed-Ahmad Ahmadi, et al. Simultaneous imputation and classification using multigraph geometric matrix completion (MGMC): Application to neurodegenerative disease classification. *Artificial Intelligence in Medicine*, 117:102097, 2021.

Alina Vretinaris, Chuan Lei, Vasilis Efthymiou, Xiao Qin, and Fatma Özcan. Medical entity disambiguation using graph neural networks. In *SIGMOD Conf.*, pp. 2310–2318, 2021.

Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications. *arXiv:2111.15367*, 2021.

Qianqian Wang, Long Li, Lishan Qiao, and Mingxia Liu. Adaptive multimodal neuroimage integration for major depression disorder detection. *Frontiers in Neuroinformatics*, 16, 2022a.

Xuesong Wang, Lina Yao, Islem Rekik, and Yu Zhang. Contrastive functional connectivity graph learning for population-based fmri classification. In *MICCAI*, pp. 221–230, 2022b.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Trans. On Graphics (tog)*, 38(5):1–12, 2019.

Bastian Wittmann, Johannes C Paetzold, Chinmay Prabhakar, Daniel Rueckert, and Bjoern Menze. Link prediction for flow-driven spatial networks. *arXiv:2303.14501*, 2023.

Jelmer M Wolterink, Tim Leiner, and Ivana Išgum. Graph convolutional networks for coronary artery segmentation in cardiac ct angiography. In *GLMI@MICCAI*, pp. 62–69, 2019.

Jermer Wolterink and Julian Suk. Geometric deep learning for precision medicine. *Key enabling technology for scientific machine learning*, pp. 60, 2021.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

Zhengwang Wu, Fenqiang Zhao, Jing Xia, Li Wang, Weili Lin, John H. Gilmore, Gang Li, and Dinggang Shen. Intrinsic patch-based cortical anatomical parcellation using graph convolutional neural network on surface manifold. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pp. 492–500, Cham, 2019. Springer International Publishing.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Trans. on neural networks and learning systems*, 32(1):4–24, 2020.

Weiyi Xie, Colin Jacobs, Jean-Paul Charbonnier, and Bram van Ginneken. Structure and position-aware graph neural network for airway labeling. *arXiv:2201.04532*, 2022.

Yiqing Xie, Sha Li, Carl Yang, Raymond Chi-Wing Wong, and Jiawei Han. When do gnns work: Understanding and improving neighborhood aggregation. In *Proc.* {*IJCAI*}, volume 2020(1), 2020.

Nuo Xu, Pinghui Wang, Long Chen, Jing Tao, and Junzhou Zhao. MR-GNN: multi-resolution and dual graph neural network for predicting structured entity interactions. In *IJCAI*, 2019.

Xiayu Xu, Peiwei Yang, Hualin Wang, Zhanfeng Xiao, Gang Xing, Xiulan Zhang, Wei Wang, Feng Xu, Jiong Zhang, and Jianqin Lei. Av-casnet: Fully automatic arteriole-venule segmentation and differentiation in oct angiography. *IEEE Trans. on Medical Imaging*, 2022.

Chunde Yang, Panyu Wang, Jia Tan, Qingshui Liu, and Xinwei Li. Autism spectrum disorder diagnosis using graph attention network based on spatial-constrained sparse functional brain networks. *Computers in Biology and Medicine*, 139:104963, 2021.

Dongren Yao, Jing Sui, Mingliang Wang, Erkun Yang, Yeerfan Jiaerken, Na Luo, Pew-Thian Yap, Mingxia Liu, and Dinggang Shen. A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity. *IEEE transactions on medical imaging*, 40(4): 1279–1289, 2021.

Melissa Min-Szu Yao, Hao Du, Mikael Hartman, Wing P Chan, and Mengling Feng. End-to-end calcification distribution pattern recognition for mammograms: An interpretable approach with GNN. *Diagnostics*, 12 (6):1376, 2022.

Zi Ye, Yogan Jaya Kumar, Goh Ong Sing, Fengyan Song, and Junsong Wang. A comprehensive survey of graph neural networks for knowledge graphs. *IEEE Access*, 10:75729–75741, 2022.

Hai-Cheng Yi, Zhu-Hong You, De-Shuang Huang, and Chee Keong Kwoh. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1):bbab340, 2022.

Hao Yu, Jie Zhao, and Li Zhang. Vessel segmentation via link prediction of graph neural networks. In *MMMI@MICCAI*, pp. 34–43, 2022a.

Jianhui Yu, Chaoyi Zhang, Heng Wang, Dingxin Zhang, Yang Song, Tiange Xiang, Dongnan Liu, and Weidong Cai. 3D medical point transformer: Introducing convolution to attention networks for medical point cloud analysis. *arXiv:2112.04863*, 2021a.

Renping Yu, Cong Pan, Xuan Fei, Mingming Chen, and Dinggang Shen. Multi-graph attention networks with bilinear convolution for diagnosis of schizophrenia. *IEEE Journal of Biomedical and Health Informatics*, 2023.

Shuangzhi Yu, Shuqiang Wang, Xiaohua Xiao, Jiuwen Cao, Guanghui Yue, Dongdong Liu, Tianfu Wang, Yanwu Xu, and Baiying Lei. Multi-scale enhanced graph convolutional network for early mild cognitive impairment detection. In *MICCAI*, pp. 228–237, 2020.

Weihao Yu, Hao Zheng, Yun Gu, Fangfang Xie, Jie Yang, Jiayuan Sun, and Guang-Zhong Yang. Tnn: Tree neural network for airway anatomical labeling. *IEEE Trans. on Medical Imaging*, 42(1):103–118, 2022b.

Xiang Yu, Siyuan Lu, Lili Guo, Shui-Hua Wang, and Yu-Dong Zhang. ResGNet-C: A graph convolutional neural network for detection of COVID-19. *Neurocomputing*, 452:592–605, 2021b.

Zehua Yu, Xianwei Zheng, Zhulun Yang, Bowen Lu, Xutao Li, and Maxian Fu. Interaction-temporal GCN: A hybrid deep framework for covid-19 pandemic analysis. *IEEE Open Journal of Engineering in Medicine and Biology*, 2:97–103, 2021c.

Kun Zhan, Xiaojun Chang, Junpeng Guan, Ling Chen, Zhigang Ma, and Yi Yang. Adaptive structure discovery for multimedia analysis using multiple features. *IEEE transactions on cybernetics*, 49(5):1826–1834, 2018.

Li Zhang, Mohan Chen, Anurag Arnab, Xiangyang Xue, and Philip HS Torr. Dynamic graph message passing networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5712–5730, 2022a.

Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. Graph neural networks and their current applications in bioinformatics. *Frontiers in genetics*, 12:690049, 2021a.

Yingying Zhang, Xian Wu, Quan Fang, Shengsheng Qian, and Chengsheng Xu. Knowledge-enhanced attributed multi-task learning for medicine recommendation. *ACM Trans. on Information Systems (TOIS)*, 2022b.

Yixiao Zhang, Xiaosong Wang, Ziyue Xu, Qihang Yu, Alan Yuille, and Daguang Xu. When radiology report generation meets knowledge graph. In *Proc. AAAI*, volume 34(07), pp. 12910–12917, 2020.

Yu Zhang, Loïc Tetrel, Bertrand Thirion, and Pierre Bellec. Functional annotation of human cognitive states using deep graph convolution. *NeuroImage*, 231:117847, 2021b.

Fenqiang Zhao, Zhengwang Wu, and Gang Li. Deep learning in cortical surface-based neuroimage analysis: a systematic review. *Intelligent Medicine*, 3(1):46–58, 2023. ISSN 2667-1026. doi: https://doi.org/10.1016/j.imed.2022.06.002. URL https://www.sciencedirect.com/science/article/pii/S2667102622000493.

Gangming Zhao. Cross chest graph for disease diagnosis with structural relational reasoning. In *Proc. ACM Multimedia*, pp. 612–620, 2021.

Kanhao Zhao, Boris Duka, Hua Xie, Desmond J Oathes, Vince Calhoun, and Yu Zhang. A dynamic graph convolutional neural network framework reveals new insights into connectome dysfunctions in ADHD. *NeuroImage*, 246:118774, 2022.

Tianyi Zhao and Zhaozheng Yin. Airway anomaly detection by prototype-based graph neural network. In *MICCAI*, pp. 195–204, 2021.

Tianyi Zhao, Yang Hu, Linda R Valsdottir, Tianyi Zang, and Jiajie Peng. Identifying drug–target interactions based on graph convolutional network and deep neural network. *Briefings in bioinformatics*, 22(2):2141–2150, 2021.

Kaizhong Zheng, Shujian Yu, Baojuan Li, Robert Jenssen, and Badong Chen. Brainib: Interpretable brain network-based psychiatric diagnosis with graph information bottleneck. *arXiv:2205.03612*, 2022a.

Shuai Zheng, Zhenfeng Zhu, Zhizhe Liu, Zhenyu Guo, Yang Liu, Yuchen Yang, and Yao Zhao. Multi-modal graph learning for disease prediction. *IEEE Trans. on Medical Imaging*, 41(9):2207–2216, 2022b.

Zhi Zheng, Chao Wang, Tong Xu, Dazhong Shen, Penggang Qin, Baoxing Huai, Tongzhu Liu, and Enhong Chen. Drug package recommendation via interaction-aware graph induction. In *Proc. WWW*, pp. 1284–1295, 2021.

Yi Zhou, Tianfei Zhou, Tao Zhou, Huazhu Fu, Jiacheng Liu, and Ling Shao. Contrast-attentive thoracic disease recognition with dual-weighting graph reasoning. *IEEE Trans. on Medical Imaging*, 40(4):1196–1206, 2021.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. *arXiv e-prints*, pp. arXiv–2103, 2021.

Yonghua Zhu, Junbo Ma, Changan Yuan, and Xiaofeng Zhu. Interpretable learning based dynamic graph convolutional networks for alzheimer's disease analysis. *Information Fusion*, 77:53–61, 2022.

5. Extended Graph Assessment Metrics for Regression and Weighted Graphs

Tamara T. Mueller, Sophie Starck, Leonhard F. Feiner, Kyriaki-Margarita Bintsi, Daniel Rueckert, and Georgios Kaissis

Conference: 5th Workshop on Graphs in Biomedical Image Analysis (GRAIL) at International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2023. Published in Lecture Notes in Computer Science, Springer, Cham., pages 14-26, 2024.

Synopsis: When re-structuring patient cohorts into so-called population graphs, initially independent patients can be incorporated into one interconnected graph structure. This population graph can then be used for medical downstream tasks using graph neural networks (GNNs). The construction of a suitable graph structure is a challenging step in the learning pipeline that can have severe impact on model performance. To this end, different graph assessment metrics have been introduced to evaluate graph structures. However, these metrics are limited to classification tasks and discrete adjacency matrices, only covering a small subset of real-world applications. In this work, we introduce extended graph assessment metrics (GAMs) for regression tasks and weighted graphs. We focus on two GAMs in specific: homophily and cross-class neighbourhood similarity (CCNS). We extend the notion of GAMs to more than one hop, define homophily for regression tasks, as well as continuous adjacency matrices, and propose a light-weight CCNS distance for discrete and continuous adjacency matrices. We show the correlation of these metrics with model performance on different medical population graphs and under different learning settings, using the TADPOLE and UKBB datasets.

Contributions of thesis author: project planning, data processing, design and implementation of source code, planning and execution of experiments, manuscript writing.

Extended Graph Assessment Metrics for Regression and Weighted Graphs

Tamara T. Mueller[1], Sophie Starck[1], Leonhard F. Feiner[1,2], Kyriaki-Margarita Bintsi[3], Daniel Rueckert[1,3], and Georgios Kaissis[1,2,4]

[1] Institute for AI in Medicine and Healthcare, Faculty of Informatics, Technical University of Munich
[2] Department of Diagnostic and Interventional Radiology, Faculty of Medicine, Technical University of Munich
[3] BioMedIA, Department of Computing, Imperial College London
[4] Institute for Machine Learning in Biomedical Imaging, Helmholtz-Zentrum Munich
tamara.mueller@tum.de

Abstract. When re-structuring patient cohorts into so-called population graphs, initially independent patients can be incorporated into one interconnected graph structure. This population graph can then be used for medical downstream tasks using graph neural networks (GNNs). The construction of a suitable graph structure is a challenging step in the learning pipeline that can have severe impact on model performance. To this end, different graph assessment metrics have been introduced to evaluate graph structures. However, these metrics are limited to classification tasks and discrete adjacency matrices, only covering a small subset of real-world applications. In this work, we introduce extended graph assessment metrics (GAMs) for regression tasks and weighted graphs. We focus on two GAMs in specific: *homophily* and *cross-class neighbourhood similarity* (CCNS). We extend the notion of GAMs to more than one hop, define homophily for regression tasks, as well as continuous adjacency matrices, and propose a light-weight CCNS distance for discrete and continuous adjacency matrices. We show the correlation of these metrics with model performance on different medical population graphs and under different learning settings, using the TADPOLE and UKBB datasets.

1 Introduction

The performance of graph neural networks can be highly dependent on the graph structure they are trained on [16,15]. To this end, several graph assessment metrics (GAMs) have been introduced to evaluate graph structures and shown strong correlations between specific graph structures and the performance of graph neural networks (GNNs) [14,16,15]. Especially in settings, where the graph structure is not provided by the dataset but needs to be constructed from the data, GAMs are the only way to assess the quality of the constructed graph. This is for example the case when utilising so-called population graphs on medical datasets. Recent works have furthermore shown that learning the graph structure in an end-to-end manner, can improve performance on population graphs [9]. Some of these methods that learn the graph structure during model training operate with fully connected, weighted graphs, where all nodes are connected with each other and the tightness of the connection is determined by a learnable edge weight. This leads to a different representation of the graph, which does not fit the to-date formulations of GAMs. Furthermore, existing metrics are tailored to classification tasks and cannot be easily transformed for equally important regression tasks. The contributions of this work are the following: (1) We extend existing metrics to allow for an assessment of multi-hop neighbourhoods. (2) We introduce an extension of the homophily metric for regression tasks and continuous adjacency matrices and (3) define a cross-class neighbourhood similarity (CCNS) distance metric and extend CCNS to learning tasks that operate on continuous adjacency matrices. Finally, (4) we show these metrics' correlation to model performance on different medical and synthetic datasets. The metrics introduced in this work can find versatile applications in the area of graph deep learning in medical and non-medical settings, since they strongly correlate with model performance and give insights into the graph structure in various learning settings.

2 Background and Related Work

2.1 Definition of graphs

A discrete graph $G := (V, E)$ is defined by a set of n nodes V and a set of edges E, connecting pairs of nodes. The edges are unweighted and can be represented by an adjacency matrix $\mathbf{A}$ of shape $n \times n$, where $\mathbf{A}_{ij} = 1$ if and only if $e_{ij} \in E$ and 0 otherwise. A continuous/weighted graph $G_w := (V_w, E_w, \mathbf{W})$, assigns a (continuous) weight to every edge in E_w, summarised in the weight matrix $\mathbf{W}$. Continuous graphs are for example required in cases where the adjacency matrix is learned in an end-to-end manner and backpropagation through the adjacency matrix needs to be feasible. A neighbourhood $\mathcal{N}_v$ of a node v contains all direct neighbours of v and can be extended to k hops by $\mathcal{N}_v^{(k)}$. For this work, we assume familiarity with GNNs [3].

2.2 Homophily

Homophily is a frequently used metric to assess a graph structure that is correlated to GNN performance [15]. It quantifies how many neighbouring nodes share the same label [15] as the node of interest. There exist three different notions of homophily: edge homophily [10], node homophily [19], and class homophily [12,15]. Throughout this work, we use node homophily, sometimes omitting the term "node", only referring to "homophily".

Definition 1 (Node homophily). *Let $G := (V, E)$ be a graph with a set of node labels $Y := \{y_u; u \in V\}$ and $\mathcal{N}_v$ be the set of neighbouring nodes to node v. Then G has the following node homophily:*

$$h(G, Y) := \frac{1}{|V|} \sum_{v \in V} \frac{|\{u | u \in \mathcal{N}_v, Y_u = Y_v\}|}{|\mathcal{N}_v|}, \tag{1}$$

where $|\cdot|$ indicates the cardinality of a set.

A graph G with node labels Y is called *homophilous*/*homophilic* when $h(G, Y)$ is large (typically larger than 0.5) and *heterophilous*/*heterophilic* otherwise [10].

2.3 Cross-class neighbourhood similarity

Ma et al. [16] introduce a metric to assess the graph structure for graph deep learning, called cross-class neighbourhood similarity (CCNS). This metric indicates how similar the neighbourhoods of nodes with the same labels are over the whole graph – irrespective of the labels of the neighbouring nodes.

Definition 2 (Cross-class neighbourhood similarity). *Let $G = (V, E)$, $\mathcal{N}_v$, and Y be defined as above. Let C be the set of node label classes, and $\mathcal{V}_c$ the set of nodes of class c. Then the CCNS of two classes c and c' is defined as follows:*

$$\mathrm{CCNS}(c, c') = \frac{1}{|\mathcal{V}_c||\mathcal{V}_{c'}|} \sum_{u \in \mathcal{V}, v \in \mathcal{V}'} \mathrm{cossim}(d(u), d(v)). \tag{2}$$

$d(v)$ is the histogram of a node v's neighbours' labels and $\mathrm{cossim}(\cdot, \cdot)$ *the cosine similarity.*

3 Extended Graph Metrics

In this section, we introduce our main contributions by defining new extended GAMs for regression tasks and continuous adjacency matrices. We propose (1) a unidimensional version of CCNS which we call *CCNS distance*, which is easier to evaluate than the whole original CCNS matrix, (2) an extension of existing metrics to k-hops, (3) GAMs for continuous adjacency matrices, and (4) homophily for regression tasks.

3.1 CCNS distance

The CCNS of a dataset with n classes is an $n \times n$ matrix, which can be large and cumbersome to evaluate. The most desirable CCNS for graph learning has high intra-class and low inter-class values, indicating similar neighbourhoods for the same class and different neighbourhoods between classes. We propose to collapse the CCNS matrix into a single value by evaluating the L_1 distance between the CCNS and the identity matrix, which we term *CCNS distance*.

Definition 3 (CCNS distance). *Let $G = (V, E)$, C, CCNS be defined as above. Then the CCNS distance of G is defined as follows:*

$$D_{\text{CCNS}} := \frac{1}{n} \sum \|\text{CCNS} - \mathbb{I}\|_1, \tag{3}$$

where $\mathbb{I}$ indicates the identity matrix and $\|\cdot\|_1$ the L_1 norm.

We note that the *CCNS distance* is best at low values and that we do not define CCNS for regression tasks, since it requires the existence of class labels.

3.2 *K*-hop metrics

Most GAMs only evaluate direct neighbourhoods. However, GNNs can apply the message passing scheme to more hops, including more hops in the node feature embedding. We therefore propose to extend homophily and CCNS on unweighted graphs to k-hop neighbourhoods. An extension of the metrics on weighted graphs is more challenging, since the edge weights impact the k-hop metrics. The formal definitions for k-hop homophily and CCNS for unweighted graphs can be found in the Appendix. We here exchange the notion of $\mathcal{N}_v$ with the specific k-hop neighbourhood $\mathcal{N}_v^{(k)}$ of interest.

3.3 Metrics for continuous adjacency matrices

Several graph learning settings, such as [6,9], utilise a continuous graph structure. In order to allow for an evaluation of those graphs, we here define GAMs on the weight matrix $\mathbf{W}$ instead of the binary adjacency matrix $\mathbf{A}$.

Definition 4 (Homophily for continuous adjacency matrices). *Let $G_w = (V_w, E_w, \mathbf{W})$, be a weighted graph defined as above with a continuous adjacency matrix. Then the 1-hop node homophily of G_w is defined as follows:*

$$\text{HCont}(G_w, Y) := \frac{1}{|V|} \sum_{v \in V} \left(\frac{\sum_{u \in \mathcal{N}_v | y_u = y_v} w_{uv}}{\sum_{u \in \mathcal{N}_v} w_{uv}} \right), \tag{4}$$

where w_{uv} is the weight of the edge from u to v.

Definition 5 (CCNS for continuous adjacency matrices). *Let $G_w = (V_w, E_w, \mathbf{W})$, C, cossim$(\cdot, \cdot)$ be defined as above. Then, the CCNS for weighted graphs is defined as follows:*

$$\text{CCNS}_{cont}(c, c') := \frac{1}{|\mathcal{V}_c||\mathcal{V}_{c'}|} \sum_{u \in \mathcal{V}, v \in \mathcal{V}'} \text{cossim}(d_c(u), d_c(v)), \tag{5}$$

where $d_c(u)$ is the histogram considering the edge weights of the continuous adjacency matrix of the respective classes instead of the count of neighbours. The *CCNS distance* for continuous adjacency matrices can be evaluated as above.

3.4 Homophily for regression

Homophily is only defined for node classification tasks, which strictly limits its application to a subset of use cases. However, many relevant graph learning tasks perform a downstream node regression, such as age regression [21,2]. We here define homophily for node regression tasks. Since homophily is a metric ranging from 0 to 1, we contain this range for regression tasks by normalising the labels between 0 and 1 prior to metric evaluation. We subtract the average node label distance from 1 to ensure the same range as homophily for classification.

Definition 6 (Homophily for regression). *Let $G = (V, E)$ and $\mathcal{N}_v^k$ be defined as above and Y be the vector of node labels, which is normalised between 0 and 1. Then the k-hop homophily for regression is defined as follows:*

$$\mathrm{HReg}^{(k)}(G, Y) := 1 - \left(\frac{1}{|V|} \sum_{v \in V} \left(\frac{1}{|\mathcal{N}_v^{(k)}|} \sum_{n \in \mathcal{N}_v^{(k)}} \|y_v - y_n\|_1 \right) \right), \tag{6}$$

where $\|\cdot\|_1$ indicates the L_1 norm.

Definition 7 (Homophily for continuous adjacency matrices for regression). *Let $G_w = (V_w, E_w, \mathbf{W})$, Y, and N_v be defined as above and the task be a regression task, then the homophily of G is defined as follows:*

$$\mathrm{HReg}(G, Y) := 1 - \left(\frac{1}{|V|} \sum_{v \in V} \left(\frac{\sum_{n \in \mathcal{N}_v} w_{nv} \|y_v - y_n\|_1}{\sum_{n \in \mathcal{N}_v} w_{nv}} \right) \right), \tag{7}$$

where w_{nv} is the weight of the edge from n to v and $\|\cdot\|_1$ the L_1 norm.

3.5 Metric evaluation

In general, we recommend the evaluation of GAMs separately on the train, validation, and test set. We believe this to be an important evaluation step since the metrics can differ significantly between the different sub-graphs, given that the graph structure in only optimised on the training set.

4 Experiments and Results

We evaluate our metrics on several datasets with different graph learning techniques: We (1) assess benchmark classification datasets using a standard learning pipeline, and (2) medical population graphs for regression and classification that learn the adjacency matrix end-to-end. All experiments are performed in a transductive learning setting using graph convolutional networks (GCNs) [11]. In order to evaluate all introduced GAMs, we specifically perform experiments on two task settings: classification and regression, and under two graph learning settings: one using a discrete adjacency matrix and one using a continuous one.

4.1 Datasets

In order to evaluate the above defined GAMs, we perform node-level prediction experiments with GNNs on different datasets. We evaluate $\{1, 2, 3\}$-hop homophily and CCNS distance on the benchmark citation datasets Cora, CiteSeer, and PubMed [24], Computers and Photos, and Coauthors CS datasets [20]. All of these datasets are classification tasks. We use k-layer GCNs and compare performance to a multi-layer perceptron (MLP).

Furthermore, we evaluate the introduced metrics on two different medical population graph datasets, as well as two synthetic datasets. The baseline results for these datasets can be found in Appendix Table 4. We generate **synthetic datasets** for classification and regression to analyse the metrics in a controllable setting. As a real-world medical classification dataset, we use **TADPOLE** [17], a neur-imaging dataset which has been frequently used for graph learning on population graphs [18,6,9]. For a regression population graph, we perform brain age prediction on 6 406 subjects of the UK BioBank [22] (**UKBB**). We use 22 clinical and 68 imaging features extracted from the subjects' magnetic resonance imaging (MRI) brain scans, following the approach in [5]. In both medical population graphs, each subject is represented by one node and similar subjects are either connected following the k-nearest neighbours approach, like in [9] or starting without any edges.

4.2 GNN Training

Prior to this work, the homophily metric has only existed for an evaluation on discrete adjacency matrices. In this work, we extend this metric to continuous adjacency matrices. In order to evaluate the metrics for both, discrete and continuous adjacency matrices, we use two different graph learning methods: (a) *dDGM* and (b) *cDGM* from [9]. DGM stands for "differentiable graph module", referring to the fact that both methods learn the adjacency matrix in an end-to-end manner. cDGM hereby uses a continuous adjacency matrix, allowing us to evaluate the metrics introduced specifically for this setting. dDGM uses a discrete adjacency matrix by sampling the edges using the Gumbel-Top-K trick [8]. Both methods are similar in terms of model training and performance, allowing us to compare the newly introduced metrics to the existing homophily metric in the dDGM setting.

4.3 Results

(1) Benchmark classification datasets The results on the benchmark datasets are summarised in Table 1. We can see that the k-hop metric values can differ greatly between the different hops for some datasets, while staying more constant for others. This gives an interesting insight into the graph structure over several hops. We believe an evaluation of neighbourhoods in graph learning to be more insightful if the number of hops in the GNN matches the number of hops considered in the graph metric. Interestingly, performance of k-hop GCNs did not

Table 1. K-hop graph metrics of benchmark node classification datasets. Cl.: number of classes, Nodes: number of nodes

Dataset	Nodes	Cl.	Node homophily ↑			D_{CCNS} ↓		
			1-hop	2-hop	3-hop	1-hop	2-hop	3-hop
Cora	1,433	7	0.825 ± 0.29	0.775 ± 0.26	0.663 ± 0.29	0.075	0.138	0.229
CiteSeer	3,703	6	0.706 ± 0.40	0.754 ± 0.28	0.712 ± 0.29	0.124	0.166	0.196
PubMed	19,717	3	0.792 ± 0.35	0.761 ± 0.26	0.687 ± 0.26	0.173	0.281	0.363
Computers	13,752	10	0.785 ± 0.26	0.569 ± 0.27	0.303 ± 0.20	0.080	0.275	0.697
Photo	7,650	8	0.837 ± 0.25	0.660 ± 0.30	0.447 ± 0.28	0.072	0.210	0.429
Coauthor CS	18, 333	15	0.832 ± 0.24	0.698 ± 0.25	0.520 ± 0.25	0.043	0.110	0.237

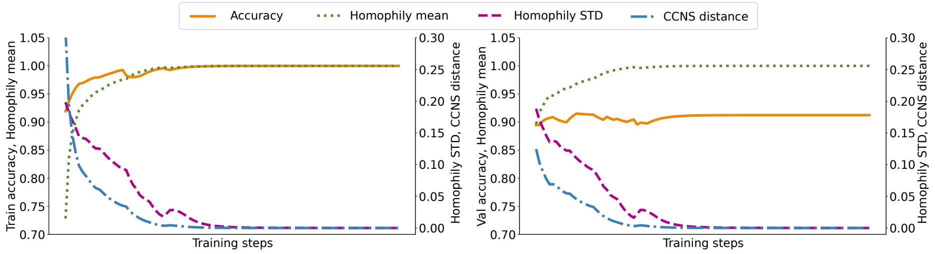


Fig. 1. Development of graph metrics on TADPOLE over training using **cDGM**; left: train set; right: validation set

align with the k-hop metric values on the specific datasets. We summarise these results in Appendix Table 3. One possible reason for this might be that, e.g., the 3-hop metrics assess the 1, 2, and 3-hop neighbourhood at once, not just the outer ring of neighbours. Another reason for this discrepancy might be that homophily and CCNS do not perfectly predict GNN performance. Furthermore, different graph convolutions have shown to be affected differently by low-homophily graphs [25]. We believe this to be an interesting direction to further investigate GAMs for GNNs.

(2) Population graph experiments Table 2 shows the dDGM and cDGM results of the population graph datasets. We can see that in some settings, such as the classification tasks on the synthetic dataset using dDGM, the homophily varies greatly between train and test set. This can be an indication for over-fitting on the training set, since the graph structure is optimised for the training nodes only and might not generalise well to the whole graph.

Since we here use graph learning methods which adapt the graph structure during model training, also the graph metrics change over training. Figure 1 shows the development of the accuracy as well as the mean and standard deviation of the 1-hop homophily and CCNS distance, evaluated on the train (left) and validation set (right). We can see that for both sets, the homophily increases with the accuracy, while the standard deviation (STD) of the homophily decreases and the CCNS distance decreases with increasing performance. However, the GAMs

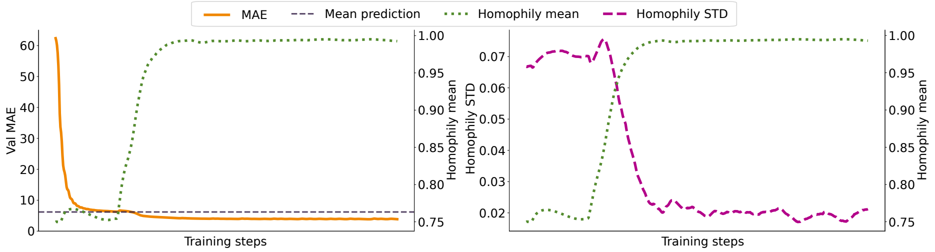


Fig. 2. Development of metrics on UKBB dataset using **cDGM** on validation set

Table 2. cDGM and dDGM results on the population graph datasets. We report the test scores averaged over 5 random seeds and 1-hop homophily and CCNS distance of one final model each. We do not report CCNS distance on regression datasets, since it is not defined for regression tasks.

Method	**Dataset**	**Task**	**Test score**	**1-hop node homophily ↑**		**1-hop D_{CCNS} ↓**	
				train	**test**	**train**	**test**
cDGM	Synthetic 1k	c	0.7900 ± 0.08	1.0000 ± 0.00	1.0000 ± 0.00	0.0000	0.0000
		r	0.0112 ± 0.01	0.9993 ± 0.00	0.9991 ± 0.00	-	-
	Synthetic 2k	c	0.8620 ± 0.03	1.0000 ± 0.00	1.0000 ± 0.00	0.0000	0.0000
		r	0.0173 ± 0.00	0.8787 ± 0.06	0.8828 ± 0.05	-	-
	Tadpole	c	0.9333 ± 0.01	1.0000 ± 0.00	0.9781 ± 0.09	0.0000	0.0314
	UKBB	r	4.0775 ± 0.23	0.8310 ± 0.06	0.8306 ± 0.07	-	-
dDGM	Synthetic 1k	c	0.8080 ± 0.04	0.6250 ± 0.42	0.1150 ± 0.32	0.4483	0.4577
		r	0.0262 ± 0.00	0.7865 ± 016	0.8472 ± 0.15	-	-
	Synthetic 2k	c	0.7170 ± 0.06	0.6884 ± 0.40	0.0950 ± 0.29	0.4115	0.4171
		r	0.0119 ± 0.00	0.8347 ± 0.13	0.8295 ± 0.13	-	-
	Tadpole	c	0.9614 ± 0.01	0.9297 ± 0.18	0.8801 ± 0.31	0.1045	0.0546
	UKBB	r	3.9067 ± 0.04	0.8941 ± 0.13	0.9114 ± 0.12	-	-

align more accurately with the training accuracy (left), showing that the method optimised the graph structure on the training set. The validation accuracy does not improve much in this example, while the validation GAMs still converge similarily to the ones evaluated on the train set (left). Figure 2 shows the mean (left) and STD (right) of the validation regression homophily HReg on the UKBB dataset with continuous adjacency matrices (using cDGM) and the corresponding change in validation mean absolute error (MAE). Again, homophily raises when the validation MAE decreases and the STD of the homophily decreases in parallel. On the left, the dotted grey line indicates the MAE of a mean prediction on the dataset. We can see that the mean regression homophily HReg raises once the validation MAE drops below the error of a mean prediction. We here only visualise a subset of all performed experiments, but we observe the same trends for all settings. From these experiments we conclude that the here introduced GAMs show strong correlation with model performance and can be used to assess generated graph structures that are used for graph deep learning.

5 Conclusion and Future Work

In this work, we extended two frequently used graph assessment metrics (GAMs) for graph deep learning, that allow to evaluate the graph structure in regression tasks and continuous adjacency matrices. For datasets that do not come with a pre-defined graph structure, like population graphs, the assessment of the graph structure is crucial for quality checks on the learning pipeline. Node homophily and cross-class neighbourhood similarity (CCNS) are commonly used GAMs that allow to evaluate how similar the neighbourhoods in a graph are. However, these metrics are only defined for discrete adjacency matrices and classification tasks. This only covers a small portion of graph deep learning tasks. Several graph learning tasks target node regression [21,2,1]. Furthermore, recent graph learning methods have shown that an end-to-end learning of the adjacency matrix is beneficial over statically creating the graph structure prior to learning [9]. These methods do not operate on a static binary adjacency matrix, but use weighted continuous graphs, which is not considered by most current GAMs. In order to overcome these limitations, we extend the definition of node homophily to regression tasks and both node homophily and CCNS to continuous adjacency matrices. We formulate these metrics and evaluate them on different synthetic and real world medical datasets and show their strong correlation with model performance. We believe these metrics to be essential tools for investigating the performance of GNNs, especially in the setting of population graphs or similar settings that require explicit graph construction.

Our definition of the CCNS distance D_{CCNS} uses the L_1-norm to determine the distance between the node labels in order to weight each inter-class-connection equally. However, the L_1-norm is only one of many norms that could be used here. Given the strong correlation of our definition of D_{CCNS}, we show the the usage of the L_1-norm is a sensible choice. We also see an extension of the metrics for weighted graphs to multiple hops as promising next steps towards better graph assessment for GNNs.

There exist additional GAMs, such as normalised total variation and normalised smoothness value [13], neighbourhood entropy and centre-neighbour similarity [23], and aggregations similarity score and diversification distinguishability [7] that have been shown to correlate with GNN performance. An extension of these metrics to regression tasks and weighted graphs would be interesting to investigate in future works. All implementations of the here introduced metrics are differentiable. This allows for a seamless integration in the learning pipeline, e.g. as loss components, which could be a highly promising application to improve GNN performance by optimising for specific graph properties.

References

1. Berrone, S., Della Santa, F., Mastropietro, A., Pieraccini, S., Vaccarino, F.: Graph-informed neural networks for regressions on graph-structured data. Mathematics **10**(5) (2022). https://doi.org/10.3390/math10050786, https://www.mdpi.com/2227-7390/10/5/786

2. Bintsi, K.M., Baltatzis, V., Potamias, R.A., Hammers, A., Rueckert, D.: Multimodal brain age estimation using interpretable adaptive population-graph learning. arXiv preprint arXiv:2307.04639 (2023)
3. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
4. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
5. Cole, J.H.: Multimodality neuroimaging brain-age in uk biobank: relationship to biomedical, lifestyle, and cognitive factors. Neurobiology of aging **92**, 34–42 (2020)
6. Cosmo, L., Kazi, A., Ahmadi, S.A., Navab, N., Bronstein, M.: Latent-graph learning for disease prediction. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 643–653. Springer (2020)
7. Elam, J.S., Glasser, M.F., Harms, M.P., Sotiropoulos, S.N., Andersson, J.L., Burgess, G.C., Curtiss, S.W., Oostenveld, R., Larson-Prior, L.J., Schoffelen, J.M., et al.: The human connectome project: a retrospective. NeuroImage **244**, 118543 (2021)
8. Jiang, B., Zhang, Z., Lin, D., Tang, J., Luo, B.: Semi-supervised learning with graph learning-convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11313–11320 (2019)
9. Kazi, A., Cosmo, L., Ahmadi, S.A., Navab, N., Bronstein, M.M.: Differentiable graph module (dgm) for graph convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 1606–1617 (2022)
10. Kim, D., Oh, A.: How to find your friendly neighborhood: Graph attention design with self-supervision. arXiv preprint arXiv:2204.04879 (2022)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Lim, D., Li, X., Hohne, F., Lim, S.N.: New benchmarks for learning on non-homophilous graphs. arXiv preprint arXiv:2104.01404 (2021)
13. Lu, S., Zhu, Z., Gorriz, J.M., Wang, S.H., Zhang, Y.D.: Nagnn: classification of covid-19 based on neighboring aware representation from deep graph neural network. International Journal of Intelligent Systems **37**(2), 1572–1598 (2022)
14. Luan, S., Hua, C., Lu, Q., Zhu, J., Chang, X.W., Precup, D.: When do we need gnn for node classification? arXiv preprint arXiv:2210.16979 (2022)
15. Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.W., Precup, D.: Is heterophily a real nightmare for graph neural networks to do node classification? arXiv preprint arXiv:2109.05641 (2021)
16. Ma, Y., Liu, X., Shah, N., Tang, J.: Is homophily a necessity for graph neural networks? arXiv preprint arXiv:2106.06134 (2021)
17. Mueller, S.G., Weiner, M.W., Thal, L.J., Petersen, R.C., Jack, C., Jagust, W., Trojanowski, J.Q., Toga, A.W., Beckett, L.: The alzheimer's disease neuroimaging initiative. Neuroimaging Clinics **15**(4), 869–877 (2005)
18. Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Moreno, R.G., Glocker, B., Rueckert, D.: Spectral graph convolutions for population-based disease prediction. In: International conference on medical image computing and computer-assisted intervention. pp. 177–185. Springer (2017)
19. Pei, H., Wei, B., Chang, K.C.C., Lei, Y., Yang, B.: Geom-gcn: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287 (2020)

20. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. arXiv preprint arXiv:1811.05868 (2018)
21. Stankeviciute, K., Azevedo, T., Campbell, A., Bethlehem, R., Lio, P.: Population graph gnns for brain age prediction. In: Proceedings of the ICML. vol. 202 (2020)
22. Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al.: Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS medicine **12**(3), e1001779 (2015)
23. Xie, Y., Li, S., Yang, C., Wong, R.C.W., Han, J.: When do gnns work: Understanding and improving neighborhood aggregation. In: IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI} 2020. vol. 2020 (2020)
24. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. pp. 40–48. PMLR (2016)
25. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. Advances in neural information processing systems **33**, 7793–7804 (2020)

6. Body Fat Estimation from Surface Meshes using Graph Neural Networks

Tamara T. Mueller*, Siyu Zhou*, Sophie Starck, Friederike Jungmann, Alexander Ziller, Orhun Aksoy, Danylo Movchan, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert

*shared first-authorship

Conference: International Workshop on Shape in Medical Imaging (ShapeMI), held in conjunction with the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2023, Published in Lecture Notes in Computer Science, Springer Nature, pages 105-117.

Synopsis: Body fat volume and distribution can be a strong indication for a person's overall health and the risk for developing diseases like type 2 diabetes and cardiovascular diseases. Frequently used measures for fat estimation are the body mass index (BMI), waist circumference, or the waist-hip-ratio. However, those are rather imprecise measures that do not allow for a discrimination between different types of fat or between fat and muscle tissue. The estimation of visceral (VAT) and abdominal subcutaneous (ASAT) adipose tissue volume has shown to be a more accurate measure for named risk factors. In this work, we show that triangulated body surface meshes can be used to accurately predict VAT and ASAT volumes using graph neural networks. Our methods achieve high performance while reducing training time and required resources compared to state-of-the-art convolutional neural networks in this area. We furthermore envision this method to be applicable to cheaper and easily accessible medical surface scans instead of expensive medical images.

Contributions of thesis author: data processing, project planning, execution of experiments, manuscript writing.

Body Fat Estimation from Surface Meshes using Graph Neural Networks

Tamara T. Mueller*[1,2], Siyu Zhou*[1], Sophie Starck[1], Friederike Jungmann[2], Alexander Ziller[1,2], Orhun Aksoy[1], Danylo Movchan[1], Rickmer Braren[2], Georgios Kaissis[1,2,4], and Daniel Rueckert[1,3]

[1] Institute for AI in Medicine and Healthcare, Faculty of Informatics, Technical University of Munich

[2] Department of Diagnostic and Interventional Radiology, Faculty of Medicine, Technical University of Munich

[3] Department of Computing, Imperial College London

[4] Institute for Machine Learning in Biomedical Imaging, Helmholtz-Zentrum Munich

tamara.mueller@tum.de

Abstract. Body fat volume and distribution can be a strong indication for a person's overall health and the risk for developing diseases like type 2 diabetes and cardiovascular diseases. Frequently used measures for fat estimation are the body mass index (BMI), waist circumference, or the waist-hip-ratio. However, those are rather imprecise measures that do not allow for a discrimination between different types of fat or between fat and muscle tissue. The estimation of visceral (VAT) and abdominal subcutaneous (ASAT) adipose tissue volume has shown to be a more accurate measure for named risk factors. In this work, we show that triangulated body surface meshes can be used to accurately predict VAT and ASAT volumes using graph neural networks. Our methods achieve high performance while reducing training time and required resources compared to state-of-the-art convolutional neural networks in this area. We furthermore envision this method to be applicable to cheaper and easily accessible medical surface scans instead of expensive medical images.

1 Introduction

The estimation of body composition measures refers to the qualification and quantification of different tissue types in the body as well as the estimation of their distribution throughout the body. These measures can function as risk factors of individuals and be an indicator for health and mortality risk [1,12]. One component of body composition analysis is the estimation of fatty tissue volume in the body. The strong correlation between body composition and disease risk has lead to a routine examination of measures indicating body composition in medical exams. The body mass index (BMI), for example, measures the ratio between a person's weight and height and has been shown to be an indicator for developing cardiovascular diseases, type 2 diabetes, as well as overall mortality

* These authors contributed equally to this work.

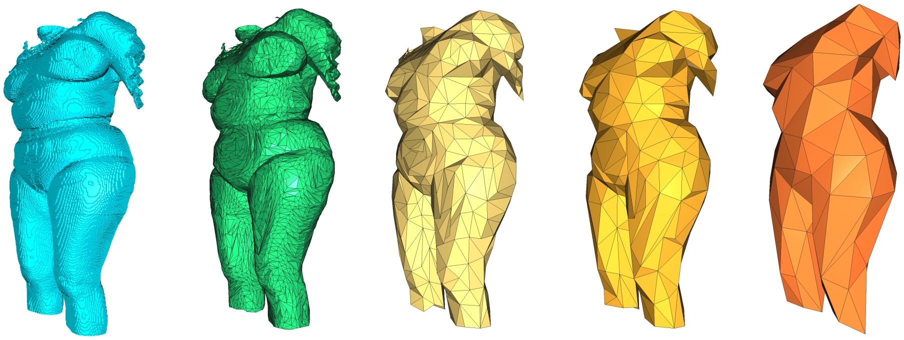


Fig. 1. Visualisation of body surface meshes at different decimation rates; The most left mesh shows the original mesh, then left to right are visualisations of decimated meshes with ten thousand, one thousand, five hundred and two hundred faces.

[28,12,3,32]. Additionally, the waist circumference and waist-hip-ratio can be used as an indication for body fat distribution [42,48,25,6]. These metrics are easy, fast, and cheap to assess. However, they have strong limitations. They are imprecise as they do not allow for a more accurate assessment of the distribution of body fat or to differentiate between weight that stems from muscle or fat tissue. Understanding the specific differences between different types of fatty tissue and their impact on health risks is crucial for accurately assessing an individual's risk factors and enabling personalised medical care. Towards this goal, several works have investigated methods to identify variations of fat distribution in the body and the quantification of fatty tissues [54,29].

Body fat can be divided into different types of fat. Two commonly investigated types are *visceral fat* (VAT), which surrounds the abdominal organs, and *abdominal subcutaneous fat* (ASAT), which is located beneath the skin. Studies have shown that especially visceral fat can have a negative impact on a person's health [40,8,47]. Therefore, a separate analysis of VAT and ASAT is an important step towards gaining accurate insights into body composition. Several works have investigated a precise estimation of VAT and ASAT volumes from medical images, like magnetic resonance (MR) [29] and computed tomography (CT) images [23], dual-energy X-ray absorptiometry (DXA) assessment [41], or ultrasound imaging [7]. Deep learning techniques have shown promising results in analysing these medical images in order to estimate body composition values [29,23,53,43].

In this work, we perform VAT and ASAT volume prediction from full body triangulated surface meshes using graph neural networks (GNNs). We show that GNNs allow to utilise the full 3D data at hand, thereby achieving better results than state-of-the-art convolutional neural networks (CNNs) on 2D silhouettes, while requiring significantly less training time and therefore resources. Both ours and related work, such as [29], use data extracted from MR images. However, MR imaging is a very expensive technique, which is highly unequally distributed around the globe. The access to MR scanners in lower income countries is much more limited [18]. Furthermore, the acquisition of MR images is time consuming and very unlikely to be used for routine exams. Given the light computational weight and fast nature of our method, we envision it to be applied to data acquired from much simpler surface scans in the future and enable an incorporation into routine medical examination.

2 Background and Related Work

In the following, we summarise related works on body fat estimation from medical (and non-medical) images, define triangulated meshes and the concept of graph neural networks and show some of their application to medical data, with a focus on surface meshes.

2.1 Body Fat Estimation from Medical Imaging

Body fat estimation has been part of routine medical assessments for decades through the analysis of simple measurements such as BMI or waist circumference [17]. However, more elaborate ways such as using proxy variables derived from medical images, like dual energy X-ray absorptiometry (DXA), CT or MR images, have achieved more accurate results. Multiple studies have successfully assessed patient body composition based upon DXA [22,15,41]. Hemke et al. [23] and Nowak et al. [43] show successful utilisation of CT images for body composition assessment. Works like [31] use segmentation algorithms to identify fatty tissue in MR scans, from which body composition values can be derived. Tian et al. [50] estimate body composition measures based on 2D photography, not even requiring medical imaging techniques. Many of these approaches focus on predicting specific types of adipose tissue [36,39,29,31]. One idea, that has been followed by several works is the utilisation of silhouettes, a binary 2D projection of the outline of the body extracted from images. Xie et al. [54] use silhouettes generated from DXA whole-body scans to estimate shape variations and Klarqvist et al. [29] use silhouettes derived from MR Images for VAT and ASAT volume estimation using CNNs. The latter use two-dimensional coronal and sagittal silhouettes of the body outline and predict VAT and ASAT volume using convolutional neural networks. The silhouettes are extracted from the full-body magnetic resonance (MR) scans of the UK Biobank dataset [49]. In our work, we propose to switch from full medical images or binary silhouettes to surface meshes for fat volume prediction, which allows to integrate the full potential of the 3D surface into deep learning methods, while using the light-weight and fast method of graph neural networks (GNNs).

2.2 Triangulated Meshes

In this work, we use triangulated surface meshes of the body outline. A mesh structure can be interpreted as a specific 3D representation of a graph. A graph

$G := (V, E)$ is defined by a set of nodes V and a set of edges E, connecting pairs of nodes. The nodes usually contain node features, which can be summarised in a node features matrix $\mathbf{X}$. A triangulated mesh M has the same structure, commonly holding the 3D coordinates of the nodes as node features. All edges form triangular faces that define the surface of the object of interest –in our case: body surfaces. A visualisation of such meshes can be found in Figure 1.

2.3 Graph Neural Networks

Graph neural networks have opened the field of deep learning to non-Euclidean data structures such as graphs and meshes [11]. Since their introduction by [20] and [46], they have been utilised in various domains, including medical research [2,14]. Graphs are, for example, frequently used for representations of brain graphs [9], research in drug discovery [10], or bioinformatics [55,56]. One native data structure that benefits from the utilisation of graph neural networks are surface meshes [11]. GNNs on mesh datasets have also advanced research in the medical domain such as brain morphology estimation [5], which can be used for Alzheimer's disease classification, or for the predicting of soft tissue deformation in image–guided neurosurgery [45].

In general, GNNs follow a so-called message passing scheme, where node features are aggregated among neighbourhoods, following the underlying graph structure [27,13,24,30]. This way, after each iteration, a new embedding for the node features is learned. In this work, we use Graph SAGE [21] convolutions, which were designed for applications on large graphs. The mean aggregator architecture for a node $v \in \mathcal{V}$ at step k is defined as follows:

$$h_v^k = \sigma\left(\mathbf{W} \cdot \mathrm{MEAN}(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in \mathcal{N}_v\})\right). \tag{1}$$

$\mathcal{N}_v$ is the neighbourhood of node v, $\mathbf{W}$ is a learnable weight matrix, and MEAN the mean aggregator, which combines the node features of v at the previous step and the node features of v's neighbours.

3 Methods

We construct three different model architectures: (a) a graph neural network, (b) a simple convolutional neural network (CNN), and (c) a DenseNet and compare their performance. All models are trained using the Adam optimiser [26] and Shrinkage loss [38] and all results reported are cross-validated based on a 5-fold data split. We use a Quadro RTX 8 000 GPU for our experiments and all models predict both targets –VAT and ASAT– with the same network, following the approach from [29].

GNN Architecutre We perform a whole-graph regression task on the input meshes. The model architecture consists of a three-layer GNN with SAGE graph convolutions [21] and batch normalisation layers, followed by a max aggregation

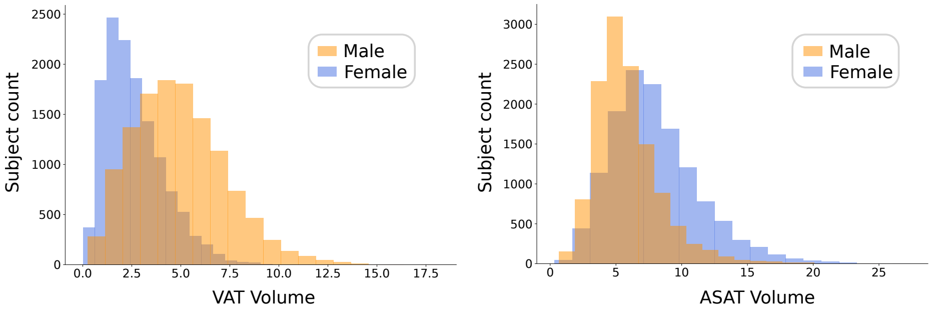


Fig. 2. Distribution of VAT (left) and ASAT (right) volume of male and female subjects in the cohort. Male subjects tend to have more VAT volume, whereas female subjects tend to have more ASAT volume.

and a three-layer multi-layer perceptron (MLP). Hyperparameters such as learning rate and GNN layers are selected by manual tuning. All GNNs are trained for 150 epochs.

CNN Architecture In order to compare our results to the work by Klarqvist et al. [29], we also train a DenseNet and a simpler CNN on the silhouette data. DenseNet is a CNN which is more densely connected, where each layer takes all previous outputs as an input. For our DenseNet implementation, we follow the architecture in [29]. We additionally construct a simpler CNN architecture that consists of three 2D convolutions, followed by a three-layer MLP, matching the design of the graph neural networks. Both convolutional networks are trained for 20 epochs on a 2D input image, that consist of a sagittal and a coronal view of the binary silhouette masks of the MR images, following the pipeline in [29].

4 Experiments and Results

We use a subset of the UK Biobank dataset [49], which is a large-scale medical database. It contains a variety of imaging data, genetics, and life-style information from almost 65 000 subjects and was acquired in the United Kingdom. In this work, we use the neck-to-knee magnetic resonance images of a subset of 25 298 subjects, for which the labels are available (12 210 male and 13 088 female). The mean age of this cohort is 62.95 years. The VAT and ASAT distributions of male and female subjects are visualised in Figure 2. We can see that female subjects tend to have a higher ASAT volume, whereas male subjects tend to have more VAT. As labels, we used the reported VAT and ASAT volumes in the UK Biobank (field IDs: 22407 and 22408).

4.1 Data Processing

The experiments in this work are performed on triangulated body surface meshes that are extracted from the neck-to-knee MR images from the UK Biobank

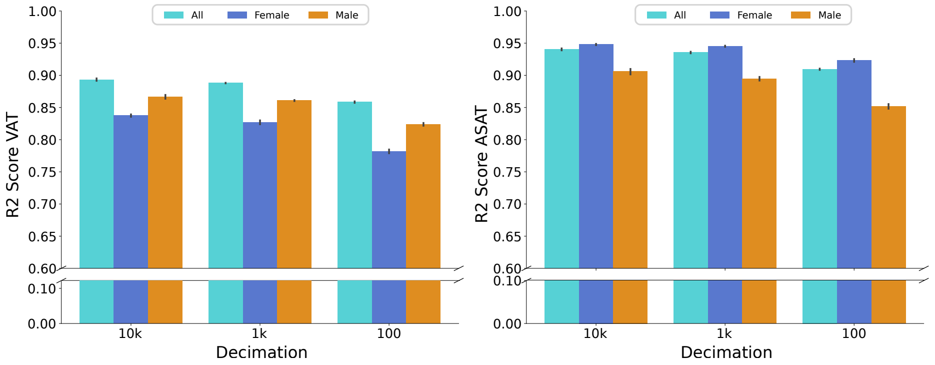


Fig. 3. R2 score results of VAT (left) and ASAT (right) predictions for all subjects, only males, and only females.

[44]. These were acquired in stations and merged through stitching [33]. In order to extract the surface meshes, we first perform an algorithmic whole-body segmentation by a succession of morphological operations on the stitched MR scans. We then convert these segmentations into surface meshes using the marching cubes algorithm [37] and the open3d library [57]. In order to investigate how much the surface meshes can be simplified, we decimate them into meshes consisting of different numbers of faces. We use meshes with 10 000, 5 000, 1 000, 500, 200, and 100 faces. The number of nodes is always half the number of faces, following Euler's formula for triangular meshes [16]. Subsequently, the meshes are registered into a common coordinate system, using the iterative closest point algorithm [4]. As a reference subject, the most average subject in the dataset was selected based on height, weight, and age. The resulting decimated and registered surface meshes are then used for graph learning. Figure 1 shows an example of a body surface mesh at different decimation rates.

4.2 Results

Table 1 summarises the results of the GNNs and CNNs for ASAT and VAT volume prediction. We report the 5-fold cross-validation results on the test set of the best performing models, evaluated on the validation loss. We compare the results of our graph neural networks (GNNs) with the results achieved by the DenseNet from [29] and the results of a simpler CNN (which we call *CNN* in the tables). We furthermore report the training times of all models, measured by the full training process for 150 and 20 epochs for GNNs and CNNs, respectively. All GNNs are trained on the body surface meshes, whereas the CNNs are trained on the silhouettes, following the approach proposed in [29]. We evaluate the GNNs on body surface meshes at different decimation rates of ten thousand, five thousand, one thousand, 500, 200, and 100 faces per mesh (see Figure 1 for a visualisation of some of these decimated meshes). The best test performances are highlighted in bold, so are the shortest training times. We can see that the

Table 1. Results for **VAT** and **ASAT** volume estimation; We report the R2 scores on the test set with standard deviations based on 5-fold cross validation, as well as the training times of the full training in minutes.

Tissue	**Model**	**Decim.**	**Test R2**	**Time (min)**
VAT	GNN (ours)	100	0.858 ± 0.001	**8.36**
		200	0.872 ± 0.001	8.63
		500	0.882 ± 0.001	9.01
		1k	0.888 ± 0.001	10.11
		5k	**0.893 ± 0.002**	22.36
		10k	0.893 ± 0.003	37.75
	CNN (ours)	-	0.874 ± 0.001	16.20
	DenseNet	-	0.878 ± 0.004	95.79
ASAT	GNN (ours)	100	0.909 ± 0.001	**8.36**
		200	0.921 ± 0.002	8.63
		500	0.931 ± 0.001	9.01
		1k	0.935 ± 0.002	10.11
		5k	0.938 ± 0.000	22.36
		10k	**0.941 ± 0.002**	37.75
	CNN (ours)	-	0.921 ± 0.002	16.20
	DenseNet	-	0.934 ± 0.002	95.79

simpler CNN architecture almost matches performance of the DenseNet proposed by [29], while requiring less training time. The GNNs outperform the CNN and the DenseNet, when the utilised meshes are not heavily decimated. But even highly decimated surface meshes with one hundred faces, only result in minor performance loss while requiring less than ten times less training time compared to the DenseNet. We envision the utilisation of the surface meshes and graph neural networks to allow for more efficient model training and the utilisation of the full 3D structure of the body, while keeping resource requirements low.

Male and female subjects show different distributions in VAT and ASAT volume. While male subjects tend to have more VAT, females tend to have more ASAT. Figure 2 shows the distributions of the fat volumes of the two sex groups. We therefore compare the results of our method for female and male subjects separately. Table 2 summarises the results of all GNNs and CNNs for VAT and ASAT volume prediction split by sex. The best performing model for each fat type and sex is highlighted in bold. We can see that the predictions of VAT volume tends to be better on male subjects whereas the prediction of ASAT volume achieves slightly higher scores for the female subject. The GNNs, however, seem to show a slightly lower gap in performance between the sex groups. We attribute the difference in performance on the different fatty tissue types to the varying distributions in fat volume between the sex groups.

Table 2. Results of **VAT** and **ASAT** volume prediction split by subject sex; all reported values are R2 scores on the test set, cross-validated across 5 folds.

Fat tissue	**Model**	**Decimation**	**Female R2**	**Male R2**
VAT	GNN (ours)	100	0.782 ± 0.004	0.824 ± 0.003
		200	0.804 ± 0.006	0.840 ± 0.003
		500	0.815 ± 0.008	0.854 ± 0.003
		1k	0.827 ± 0.004	0.861 ± 0.001
		5k	0.831 ± 0.006	**0.868 ± 0.002**
		10k	**0.837 ± 0.002**	0.867 ± 0.004
	CNN (ours)	-	0.804 ± 0.003	0.845 ± 0.002
	DenseNet	-	0.811 ± 0.006	0.849 ± 0.006
ASAT	GNN (ours)	100	0.923 ± 0.003	0.852 ± 0.004
		200	0.934 ± 0.001	0.870 ± 0.006
		500	0.940 ± 0.002	0.890 ± 0.002
		1k	0.945 ± 0.001	0.895 ± 0.004
		5k	0.945 ± 0.000	0.903 ± 0.002
		10k	**0.948 ± 0.001**	**0.906 ± 0.005**
	CNN (ours)	-	0.934 ± 0.002	0.870 ± 0.002
	DenseNet	-	0.944 ± 0.001	0.891 ± 0.003

5 Discussion and Conclusion

In this work, we introduce a graph neural network-based method that enables adipose tissue volume prediction for visceral (VAT) and abdominal subcutaneous (ASAT) fat from triangulated surface meshes. The assessment of fatty tissue has high clinical relevance, since it has been shown to be a strong risk factor for diseases like type 2 diabetes and cardiovascular diseases [28,32]. Especially a separate estimation of the two different fat tissues VAT and ASAT has shown to be a relevant medical assessment, since VAT is known to have a higher correlation with disease development compared to ASAT [40,8,47]. We here use graph neural networks and triangulated surface meshes, extracted from full-body MR scans and show that they achieve accurate VAT and ASAT volume predictions. We investigate how different decimation rates impact model performance and training times. Figure 4 visualises this correlation. The bars in the left figure show the average ASAT volume prediction R2 scores on the test set of the GNNs trained on the differently decimated meshes. The overlaid line plot notes the corresponding training times. We can see that at one thousand faces, we reach an optimal trade-off between training time and performance. Training the GNN on the meshes with one thousand faces only takes about 10 minutes and achieves high results of 0.893 R2 on VAT and 0.935 on ASAT volume prediction. On the right in Figure 4, we visualise the linear relation between the training time and the number of faces in the meshes. Training time also corresponds linearly to energy consumption in kWh. We attribute the comparably high performance of the

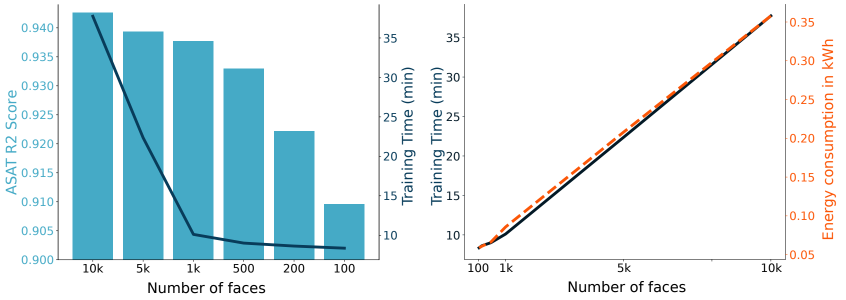


Fig. 4. Relationship between training time and decimation rate of the meshes; The left plot shows the ASAT R2 scores (bars) and the corresponding training time, the right plot shows the linear relation between the training time or the energy consumption in kWh and the number of faces of the meshes.

strongly decimated meshes to the fact that the most outer coordinates/nodes still remain in the meshes, which carry a lot of information about the outline of a body.

The light-weight nature of GNNs allows for the usage of the full 3D data, while significantly reducing resource requirements and run time compared to 3D image-based methods. This shows great promise in the effort of bridging the gap between cheap, fast, but imprecise measures –such as BMI and waist circumference– and time-consuming, costly, but accurate methods such as medical imaging (CT, MR, or DXA).

6 Limitations and Future Work

We see high potential in the utilisation of surface meshes and graph neural networks, given that the full 3D data can be utilised compared to only using binary silhouette projections like in [29]. The low training times as well as the high scores of the GNNs show the successful application to fat volume prediction. We note that we compare the run time of the training loops only. This does not include any pre-processing that is required for both silhouette-based and surface mesh-based approaches. The GNN architecture is based on SAGE graph convolutions [21], because they achieved the best results in our experiments, compared to graph attention networks [51] and graph convolutional networks [27]. A potential improvement of our method would be the utilisation of other mesh-specific convolutions such as adaptive graph convolution pooling [19] or FeaStNet [52]. Another interesting direction to explore is the utilisation of deeper GNNs. Li et al. [34], for example, introduce a method that enables the utilisation of deeper GNNs without over-smoothing –a commonly known problem with GNNs. Over-smoothing refers to the issue that deep GNNs do not achieve high performance because all node embeddings in the graph converge to the same value [35].

Our experiments are performed on surface meshes, that were extracted from MR images. However, we envision this method to work equally well on designated surface scans, without requiring expensive and time-consuming MR scans. We intend to investigate this in future work and apply our method to surface scans, which are for example acquired for dermatological examinations. This would eliminate the need for expensive MR scans and could lead to an embedding of this technique into routine medical examination.

Acknowledgements TM and SS were supported by the ERC (Deep4MI - 884622). This work has been conducted under the UK Biobank application 87802. SS has furthermore been supported by BMBF and the NextGenerationEU of the European Union.

References

1. Afshin, A., Reitsma, M.B., Murray, C.J.: Health effects of overweight and obesity in 195 countries. The New England journal of medicine **377**(15), 1496–1497 (2017)
2. Ahmedt-Aristizabal, D., Armin, M.A., Denman, S., Fookes, C., Petersson, L.: Graph-based deep learning for medical diagnosis and analysis: past, present and future. Sensors **21**(14), 4758 (2021)
3. Anderson, M.R., Geleris, J., Anderson, D.R., Zucker, J., Nobel, Y.R., Freedberg, D., Small-Saunders, J., Rajagopalan, K.N., Greendyk, R., Chae, S.R., et al.: Body mass index and risk for intubation or death in sars-cov-2 infection: a retrospective cohort study. Annals of internal medicine **173**(10), 782–790 (2020)
4. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-9**(5), 698–700 (1987). https://doi.org/10.1109/TPAMI.1987.4767965
5. Azcona, E.A., Besson, P., Wu, Y., Punjabi, A., Martersteck, A., Dravid, A., Parrish, T.B., Bandt, S.K., Katsaggelos, A.K.: Interpretation of brain morphology in association to alzheimer's disease dementia classification using graph convolutional networks on triangulated meshes. In: Shape in Medical Imaging: International Workshop, ShapeMI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings. pp. 95–107. Springer (2020)
6. Baioumi, A.Y.A.A.: Comparing measures of obesity: waist circumference, waist-hip, and waist-height ratios. In: Nutrition in the Prevention and Treatment of Abdominal Obesity, pp. 29–40. Elsevier (2019)
7. Bazzocchi, A., Filonzi, G., Ponti, F., Albisinni, U., Guglielmi, G., Battista, G.: Ultrasound: Which role in body composition? European Journal of Radiology **85**(8), 1469–1480 (2016)
8. Bergman, R.N., Kim, S.P., Catalano, K.J., Hsu, I.R., Chiu, J.D., Kabir, M., Hucking, K., Ader, M.: Why visceral fat is bad: mechanisms of the metabolic syndrome. Obesity **14**(2S), 16S (2006)
9. Bessadok, A., Mahjoub, M.A., Rekik, I.: Graph neural networks in network neuroscience. IEEE Trans. on Pattern Analysis and Machine Intelligence (2022)
10. Bonner, S., Barrett, I.P., Ye, C., Swiers, R., Engkvist, O., Bender, A., Hoyt, C.T., Hamilton, W.L.: A review of biomedical datasets relating to drug discovery: a knowledge graph perspective. Briefings in Bioinformatics **23**(6) (2022)

11. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
12. Calle, E.E., Rodriguez, C., Walker-Thurmond, K., Thun, M.J.: Overweight, obesity, and mortality from cancer in a prospectively studied cohort of us adults. New England Journal of Medicine **348**(17), 1625–1638 (2003)
13. Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J.: Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 257–266 (2019)
14. Ding, K., Zhou, M., Wang, Z., Liu, Q., Arnold, C.W., Zhang, S., Metaxas, D.N.: Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications. arXiv:2202.08916 (2022)
15. Direk, K., Cecelja, M., Astle, W., Chowienczyk, P., Spector, T.D., Falchi, M., Andrew, T.: The relationship between dxa-based and anthropometric measures of visceral fat and morbidity in women. BMC cardiovascular disorders **13**, 1–13 (2013)
16. Euler, L.: De summis serierum reciprocarum. Commentarii academiae scientiarum Petropolitanae pp. 123–134 (1740)
17. Fan, Z., Chiong, R., Hu, Z., Keivanian, F., Chiong, F.: Body fat prediction through feature extraction based on anthropometric and laboratory measurements. Plos one **17**(2), e0263333 (2022)
18. Geethanath, S., Vaughan Jr, J.T.: Accessible magnetic resonance imaging: a review. Journal of Magnetic Resonance Imaging **49**(7), e65–e77 (2019)
19. Gopinath, K., Desrosiers, C., Lombaert, H.: Adaptive graph convolution pooling for brain surface analysis. In: Chung, A.C.S., Gee, J.C., Yushkevich, P.A., Bao, S. (eds.) Information Processing in Medical Imaging. pp. 86–98. Springer International Publishing, Cham (2019)
20. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proc. 2005 IEEE Int. joint conf. on neural networks. vol. 2(2005), pp. 729–734 (2005)
21. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
22. Harty, P.S., Sieglinger, B., Heymsfield, S.B., Shepherd, J.A., Bruner, D., Stratton, M.T., Tinsley, G.M.: Novel body fat estimation using machine learning and 3-dimensional optical imaging. European journal of clinical nutrition **74**(5), 842–845 (2020)
23. Hemke, R., Buckless, C.G., Tsao, A., Wang, B., Torriani, M.: Deep learning for automated segmentation of pelvic muscles, fat, and bone from ct studies for body composition assessment. Skeletal radiology **49**, 387–395 (2020)
24. Huang, Q., He, H., Singh, A., Lim, S.N., Benson, A.R.: Combining label propagation and simple models out-performs graph neural networks. arXiv preprint arXiv:2010.13993 (2020)
25. Jacobs, E.J., Newton, C.C., Wang, Y., Patel, A.V., McCullough, M.L., Campbell, P.T., Thun, M.J., Gapstur, S.M.: Waist circumference and all-cause mortality in a large us cohort. Archives of internal medicine **170**(15), 1293–1301 (2010)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
27. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv:1609.02907 (2016)

28. Kivimäki, M., Kuosma, E., Ferrie, J.E., Luukkonen, R., Nyberg, S.T., Alfredsson, L., Batty, G.D., Brunner, E.J., Fransson, E., Goldberg, M., et al.: Overweight, obesity, and risk of cardiometabolic multimorbidity: pooled analysis of individual-level data for 120 813 adults from 16 cohort studies from the usa and europe. The Lancet Public Health **2**(6), e277–e285 (2017)
29. Klarqvist, M.D., Agrawal, S., Diamant, N., Ellinor, P.T., Philippakis, A., Ng, K., Batra, P., Khera, A.V.: Silhouette images enable estimation of body fat distribution and associated cardiometabolic risk. npj Digital Medicine **5**(1), 105 (2022)
30. Kong, K., Li, G., Ding, M., Wu, Z., Zhu, C., Ghanem, B., Taylor, G., Goldstein, T.: Flag: Adversarial data augmentation for graph neural networks. arXiv preprint arXiv:2010.09891 (2020)
31. Küstner, T., Hepp, T., Fischer, M., Schwartz, M., Fritsche, A., Häring, H.U., Nikolaou, K., Bamberg, F., Yang, B., Schick, F., Gatidis, S., Machann, J.: Fully automated and standardized segmentation of adipose tissue compartments via deep learning in 3d whole-body mri of epidemiologic cohort studies. Radiology. Artificial intelligence **2 6**, e200010 (2020)
32. Larsson, S.C., Bäck, M., Rees, J.M., Mason, A.M., Burgess, S.: Body mass index and body composition in relation to 14 cardiovascular conditions in uk biobank: a mendelian randomization study. European heart journal **41**(2), 221–226 (2020)
33. Lavdas, I., Glocker, B., Rueckert, D., Taylor, S., Aboagye, E., Rockall, A.: Machine learning in whole-body mri: experiences and challenges from an applied study using multicentre data. Clinical radiology **74**(5), 346–356 (2019)
34. Li, G., Muller, M., Thabet, A., Ghanem, B.: Deepgcns: Can gcns go as deep as cnns? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9267–9276 (2019)
35. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
36. Linder, N., Michel, S., Eggebrecht, T., Schaudinn, A., Blüher, M., Dietrich, A., Denecke, T., Busse, H.: Estimation of abdominal subcutaneous fat volume of obese adults from single-slice mri data – regression coefficients and agreement. European Journal of Radiology **130**, 109184 (2020). https://doi.org/https://doi.org/10.1016/j.ejrad.2020.109184, https://www.sciencedirect.com/science/article/pii/S0720048X20303739
37. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
38. Lu, X., Ma, C., Ni, B., Yang, X., Reid, I., Yang, M.H.: Deep regression tracking with shrinkage loss. In: Proceedings of the European conference on computer vision (ECCV). pp. 353–369 (2018)
39. Lu, Y., Shan, Y., Dai, L., Jiang, X., Song, C., Chen, B., Zhang, J., Li, J., Zhang, Y., Xu, J., Li, T., Xiong, Z., Bai, Y., Huang, X.: Sex-specific equations to estimate body composition: Derivation and validation of diagnostic prediction models using uk biobank. Clinical Nutrition **42**(4), 511–518 (2023). https://doi.org/https://doi.org/10.1016/j.clnu.2023.02.005, https://www.sciencedirect.com/science/article/pii/S0261561423000341
40. Matsuzawa, Y., Nakamura, T., Shimomura, I., Kotani, K.: Visceral fat accumulation and cardiovascular disease. Obesity research **3**(S5), 645S–647S (1995)
41. Messina, C., Albano, D., Gitto, S., Tofanelli, L., Bazzocchi, A., Ulivieri, F.M., Guglielmi, G., Sconfienza, L.M.: Body composition with dual energy x-ray absorp-

tiometry: from basics to new tools. Quantitative imaging in medicine and surgery **10**(8), 1687 (2020)

42. Neeland, I.J., Ross, R., Després, J.P., Matsuzawa, Y., Yamashita, S., Shai, I., Seidell, J., Magni, P., Santos, R.D., Arsenault, B., et al.: Visceral and ectopic fat, atherosclerosis, and cardiometabolic disease: a position statement. The lancet Diabetes & endocrinology **7**(9), 715–725 (2019)
43. Nowak, S., Faron, A., Luetkens, J.A., Geißler, H.L., Praktiknjo, M., Block, W., Thomas, D., Sprinkart, A.M.: Fully automated segmentation of connective tissue compartments for ct-based body composition analysis: a deep learning approach. Investigative radiology **55**(6), 357–366 (2020)
44. Petersen, S.E., Matthews, P.M., Bamberg, F., Bluemke, D.A., Francis, J.M., Friedrich, M.G., Leeson, P., Nagel, E., Plein, S., Rademakers, F.E., et al.: Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of uk biobank-rationale, challenges and approaches. Journal of Cardiovascular Magnetic Resonance **15**(1), 1–10 (2013)
45. Salehi, Y., Giannacopoulos, D.: Physgnn: A physics–driven graph neural network based model for predicting soft tissue deformation in image–guided neurosurgery. Advances in Neural Information Processing Systems **35**, 37282–37296 (2022)
46. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE trans. on neural networks **20**(1), 61–80 (2008)
47. Shuster, A., Patlas, M., Pinthus, J., Mourtzakis, M.: The clinical importance of visceral adiposity: a critical review of methods for visceral adipose tissue analysis. The British journal of radiology **85**(1009), 1–10 (2012)
48. Song, X., Jousilahti, P., Stehouwer, C., Söderberg, S., Onat, A., Laatikainen, T., Yudkin, J., Dankner, R., Morris, R., Tuomilehto, J., et al.: Comparison of various surrogate obesity indicators as predictors of cardiovascular mortality in four european populations. European Journal of Clinical Nutrition **67**(12), 1298–1302 (2013)
49. Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al.: Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS medicine **12**(3), e1001779 (2015)
50. Tian, I.Y., Ng, B.K., Wong, M.C., Kennedy, S., Hwaung, P., Kelly, N., Liu, E., Garber, A.K., Curless, B., Heymsfield, S.B., et al.: Predicting 3d body shape and body composition from conventional 2d photography. Medical Physics **47**(12), 6232–6245 (2020)
51. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
52. Verma, N., Boyer, E., Verbeek, J.: Feastnet: Feature-steered graph convolutions for 3d shape analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2598–2606 (2018)
53. Wang, B., Torriani, M.: Artificial intelligence in the evaluation of body composition. In: Seminars in Musculoskeletal Radiology. vol. 24, pp. 030–037. Thieme Medical Publishers (2020)
54. Xie, B., Avila, J.I., Ng, B.K., Fan, B., Loo, V., Gilsanz, V., Hangartner, T., Kalkwarf, H.J., Lappe, J., Oberfield, S., et al.: Accurate body composition measures from whole-body silhouettes. Medical physics **42**(8), 4668–4677 (2015)
55. Yi, H.C., You, Z.H., Huang, D.S., Kwoh, C.K.: Graph representation learning in bioinformatics: trends, methods and applications. Briefings in Bioinformatics **23**(1), bbab340 (2022)

56. Zhang, X.M., Liang, L., Liu, L., Tang, M.J.: Graph neural networks and their current applications in bioinformatics. Frontiers in genetics **12**, 690049 (2021)
57. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)

7. Are Population Graphs Really as Powerful as Believed?

Tamara T. Mueller, Sophie Starck, Kyriaki-Margarita Bintsi, Alexander Ziller, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert

Journal: Transactions on Machine Learning Research (TMLR), February 2024.

Synopsis: Population graphs and their use in combination with graph neural networks (GNNs) have demonstrated promising results for multi-modal medical data integration and improving disease diagnosis and prognosis. Several different methods for constructing these graphs and advanced graph learning techniques have been established to maximise the predictive power of GNNs on population graphs. However, in this work, we raise the question of whether existing methods are really strong enough by showing that simple baseline methods –such as random forests or linear regressions–, perform on par with advanced graph learning models on several population graph datasets for a variety of different clinical applications. We use the commonly used public population graph datasets TADPOLE and ABIDE, a brain age estimation and a cardiac dataset from the UK Biobank, and a real-world in-house COVID dataset. We (a) investigate the impact of different graph construction methods, graph convolutions, and dataset size and complexity on GNN performance and (b) discuss the utility of GNNs for multi-modal data integration in the context of population graphs. Based on our results, we argue towards the need for "better" graph construction methods or innovative applications for population graphs to render them beneficial.

Contributions of thesis author: data processing, design and implementation of source code, planning and execution of experiments, manuscript writing.

Are Population Graphs Really as Powerful as Believed?

Tamara T. Mueller[1] **Sophie Starck**[1] **Kyriaki-Margarita Bintsi**[2]
Alexander Ziller[1] **Rickmer Braren**[3] **Georgios Kaissis**[1,4] **Daniel Rueckert**[1,2]

[1]*AI in Medicine and Healthcare, Technical University of Munich, Germany*
[2]*BioMedIA, Imperial College London, UK*
[3]*Department for Interventional Radiology, Technical University of Munich*
[4]*Machine Learning in Biomedical Imaging, Helmholtz Munich*
Contact: tamara.mueller@tum.de

Reviewed on OpenReview: *https://openreview.net/forum?id=TTRDCVnbjI*

Abstract

Population graphs and their use in combination with graph neural networks (GNNs) have demonstrated promising results for multi-modal medical data integration and improving disease diagnosis and prognosis. Several different methods for constructing these graphs and advanced graph learning techniques have been established to maximise the predictive power of GNNs on population graphs. However, in this work, we raise the question of whether existing methods are really strong enough by showing that simple baseline methods –such as random forests or linear regressions–, perform on par with advanced graph learning models on several population graph datasets for a variety of different clinical applications. We use the commonly used public population graph datasets TADPOLE and ABIDE, a brain age estimation and a cardiac dataset from the UK Biobank, and a real-world in-house COVID dataset. We (a) investigate the impact of different graph construction methods, graph convolutions, and dataset size and complexity on GNN performance and (b) discuss the utility of GNNs for multi-modal data integration in the context of population graphs. Based on our results, we argue towards the need for "better" graph construction methods or innovative applications for population graphs to render them beneficial[1].

1 Introduction

Graphs can be used to model and represent various types of data. They allow for a suitable representation of interconnected structures, such as social networks (Fan et al., 2019), molecules (Moreira-Filho et al., 2022), or surface meshes (Mueller et al., 2023b). In order to perform deep learning on graph-like data structures, graph neural networks (GNNs) have been introduced (Gori et al., 2005; Scarselli et al., 2008). GNNs follow a message-passing scheme and collect information that is stored in nodes across a graph structure (Bronstein et al., 2017) and have shown improved performance of various deep learning tasks (Parisot et al., 2017; Ahmedt-Aristizabal et al., 2021; Bessadok et al., 2022; Pellegrini et al., 2022). Most of these tasks rely on datasets that inherently provide a graph structure, such as social networks, or provide well-established methods to construct the graph, such as point clouds (Wang et al., 2019).

In the medical domain, GNNs have been applied to improve disease diagnostics (Parisot et al., 2017; Cosmo et al., 2020; Kazi et al., 2022), model biological structures (Chen et al., 2020), or temporal components of data (Kim et al., 2021). They can be used to perform deep learning on surface meshes for fatty tissue quantification (Mueller et al., 2023b), vessel structures (Paetzold et al., 2021) for vessel segmentation, or molecules for drug discovery (Bonner et al., 2022). The respective datasets provide an inherent graph structure in the form of a mesh, a vessel tree, or chemical bindings. In contrast to datasets that provide a clear graph structure,

[1] The source code for this work can be found at: `https://github.com/tamaramueller/population_graphs`

one research area in medicine studies so-called *population graphs*. A population graph refers to a network of inter-connected subjects encoding the medical information of all subjects in graph form. Usually, the subjects' medical data, such as imaging or clinical features, is used as node features in the graph. The edges are constructed so that similar subjects are connected. Figure 1 shows a schematic of a typical population graph. Each subject (node) is represented by a data vector often extracted from medical images. Additionally, non-imaging clinical data, such as demographics or lab results, can be used to define the edges between subjects, where similar non-imaging features lead to a connection between two subjects.

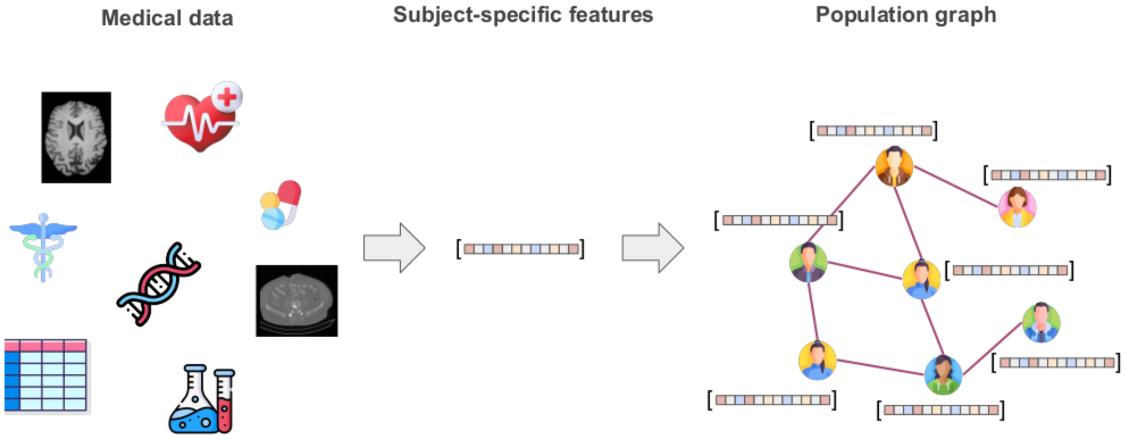


Figure 1: **Overview of a typical population graph construction**. Subject-specific medical data is represented as a feature vector and used as node features in the population graph. The most frequently used setup uses imaging features as node features and non-imaging features for edge construction.

Several works have shown that population graphs for medical applications can improve downstream tasks compared to graph-agnostic methods (Parisot et al., 2017; Kazi et al., 2019; Cosmo et al., 2020; Kazi et al., 2022). Parisot et al. (2017) first introduced the concept of population graphs for the detection of Alzheimer's disease and autism. Later works (Kazi et al., 2019; Cosmo et al., 2020; Kazi et al., 2022; Bintsi et al., 2023a;b) used the method of population graphs under different settings, developing new graph construction methods and for different tasks, such as age prediction. The motivation for using population graphs is the hypothesis that subjects that share similar phenotypes tend to have similar pathologies and, therefore, benefit from sharing information. The goal is to facilitate personalised medicine by utilising the shared information across similar subjects. However, population graphs come with a significant limitation: the graph structure needs to be constructed from the dataset. This has led to different graph construction methods. Two branches of graph construction have been established: static and dynamic graph construction. Static graph construction refers to creating the graph structure prior to graph learning, while dynamic graph construction methods adapt the graph structure during training (Cosmo et al., 2020). To date, both methods are used frequently. For an overview of graph construction methods for GNNs in medicine, we refer to Mueller et al. (2024). What makes the choice of graph construction method so crucial is the impact of the resulting graph structure on the downstream performance of the GNN. It has been shown that a "poor" graph structure can lead to GNNs under-performing graph-agnostic models (Luan et al., 2022; Zhu et al., 2020). Some methods have been specifically designed to work on such challenging graph structures, one of them being neural sheaf diffusion models (Hansen & Gebhart, 2020). We investigate their potential on population graph datasets, which tend to have challenging graph structures.

So far, there are two commonly used arguments for using medical population graphs compared to graph-agnostic models: (1) GNNs allow for meaningful multi-modal data integration, and (2) the message passing across neighbourhoods improves model performance. In this work, we investigate how firm those claims are and contradict them on several datasets. Our contributions can be summarised as follows:

- We compare static and dynamic state-of-the-art graph construction methods with GNNs, as well as the usage of neural sheaf diffusion models for population graphs and show how simple graph-agnostic baselines perform on par with them on several population graph datasets.
- We show that GNNs can be superior to graph-agnostic models if the graph structure is provided with the dataset but do not achieve performance boosts on any medical population graph dataset used in this work. We hypothesise that in the latter case, the graph structure does not add additional valuable information.
- We evaluate the impact of the graph structure on several different types of graph convolution using two different graph assessment metrics: homophily and cross-class neighbourhood similarity (CCNS) distance.
- We highlight that the graph construction methods for population graphs are not sufficient and discuss potential future directions for population graph studies.

Our results lead us to conclude that we need a discussion about whether population graphs are beneficial over graph-agnostic methods and that the currently available graph construction methods are the performance bottleneck of GNNs on population graphs. We see a requirement for "better" graph construction methods if we want to improve the performance of GNNs on population graphs.

2 Background

In this section, we discuss some background on graphs, graph neural networks, neural sheaf diffusion models, and two graph assessment metrics, namely homophily and cross-class neighbourhood similarity.

2.1 Graph Structures

A graph $G := (V, E)$ is defined as a set of n vertices/nodes V and a set of edges E, where $e_{ij} = 1$ and $e_{ij} \in E$ if there exists an edge from node i to node j. All edges can be summarised in an $n \times n$ adjacency matrix $\mathbf{A}$, where $a_{ij} = 1$ if $e_{ij} \in E$ and 0 otherwise. In the context of graph deep learning, the graph's nodes usually hold node features of dimension r that can be summarised in the node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times r}$. A neighbourhood of a node i, $\mathcal{N}_i$ is the set of all nodes j, for which an edge e_{ji} from j to i exists. Furthermore, in the setting of node classification, each node i usually holds a label y_i, and all labels can be summarised in the label vector Y.

2.2 Graph Assessment Metrics

Several works have shown that the graph structure can have a significant impact on the performance of GNNs (Luan et al., 2022; Zhu et al., 2020). In this line, different metrics have been introduced that assess graph structures and have been shown to correlate with GNN performance. The metric most commonly used is *homophily*. One can distinguish between three different types of homophily: class homophily (Lim et al., 2021; Luan et al., 2021), edge homophily (Kim & Oh, 2022), and node homophily (Pei et al., 2020), which all highlight slightly different aspects of the graph structure. They all evaluate the ratio between edges that connect nodes with the same label and edges that connect nodes with different labels. The idea is that since GNNs propagate node features across edges, the less similar the neighbours are, the less likely it is for the GNN to learn representative node feature embeddings for this node, which can impact the network's performance. In the remaining parts of this work, we will use node homophily.

Definition 2.1 (Node homophily (Pei et al., 2020)) *A graph $G := (V, E)$ with node labels $Y := \{y_u; u \in V\}$ has the following node homophily:*

$$\mathrm{hom}(G, Y) := \frac{1}{|V|} \sum_{v \in V} \frac{|\{u | u \in \mathcal{N}_v, Y_u = Y_v\}|}{|\mathcal{N}_v|}, \tag{1}$$

where $\mathcal{N}_v$ is the set of neighbouring nodes of v and $|\cdot|$ the cardinality of a set.

We speak of "high homophily" or a "homophilic" graph, when $\hom(G, Y) \to 1$ and of "low homophily" or a "heterophilic" graph, when $\hom(G, Y) \to 0$. A graph's homophily can also be defined for regression tasks by taking the distance between node feature labels among neighbourhoods into account (Mueller et al., 2023a):

Definition 2.2 (Homophily for regression (Mueller et al., 2023a)) *The node homophily of a graph G with labels Y (defined as above) that indicate a regression task is defined as follows:*

$$\hom_{\text{reg}}(G, Y) := 1 - \left(\frac{1}{|V|} \sum_{v \in V} \left(\frac{1}{|\mathcal{N}_v|} \sum_{n \in \mathcal{N}_v} \|y_v - y_n\|_1 \right) \right), \tag{2}$$

where $\|\cdot\|_1$ indicates the L_1 norm.

Another metric that does not only focus on the ratio of edges connecting same-labelled or differently-labelled nodes is cross-class neighbourhood similarity (CCNS) (Ma et al., 2021). Here, the overall similarity of neighbourhoods of nodes with the same label is evaluated, irrespective of whether the neighbours share the same label as the node of interest.

Definition 2.3 (Cross-class neighbourhood similarity (Ma et al., 2021)) *Let $G := (V, E)$, Y, and $\mathcal{N}_v$ be defined as above. In addition, let C be the set of all possible classes of node labels, and V_c the set of vertices of a specific class c. Then the CCNS of two classes c_r and c_s can be derived as follows:*

$$\text{CCNS}(c_r, c_s) = \frac{1}{|V_{c_r}||V_{c_s}|} \sum_{u,v \in V} \text{cossim}(d(u), d(v)), \tag{3}$$

where $d(v)$ indicates the histogram of a node v's neighbours' labels and $\text{cossim}(\cdot, \cdot)$ *the cosine similarity.*

Mueller et al. (2023a) introduce a reduction of CCNS to a single-valued parameter, they call *CCNS distance*, which defines the L_1 distance between the CCNS matrix and the identity matrix:

Definition 2.4 (CCNS distance (Mueller et al., 2023a)) *Let $G := (V, E)$, C, and* CCNS *be defined as above. Then the CCNS distance of the whole graph G is:*

$$D_{\text{CCNS}} := \frac{1}{n} \sum \|\text{CCNS} - \mathbb{I}\|_1, \tag{4}$$

where $\|\cdot\|_1$ is the L_1 norm and $\mathbb{I}$ the identity matrix.

2.3 Graph Neural Networks

GNNs have been introduced with the aim of enabling deep learning on non-Euclidean spaces, such as graphs, manifolds, or meshes (Bronstein et al., 2017). They all follow a so-called message-passing scheme, which propagates the information that is stored in the node features of the graph (or mesh or manifold) to its neighbouring nodes. The GNN then learns a node feature embedding based on the original node features as well as the propagated node features of the neighbouring nodes. GNNs make use of graph convolutions, which specify the concrete message-passing scheme that is applied during training and inference. There exist several different types of graph convolution, all varying slightly in their methodology. Here, we summarise the definitions of four commonly used graph convolutions.

Definition 2.5 (Graph Convolutional Networks (GCN) (Kipf & Welling, 2016)) *Graph convolutional networks (GCNs) were one of the first GNNs introduced by Kipf & Welling (2016). They were originally defined in a spectral manner, using the graph Laplacian. The* `PyTorch Geometric` *implementation follows the following definition:*

$$x_i' = \Theta^T \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} x_j, \tag{5}$$

where $\hat{d}_i = 1 + \sum_{j \in \mathcal{N}_i} 1$, Θ learnable weights, and $\mathcal{N}_i$ the neighbourhood of node i.

Definition 2.6 (Graph SAGE (Hamilton et al., 2017)) *In 2017, Hamilton et al. (2017) introduced a novel graph convolution that was originally designed for large graphs and inductive training, which is called GraphSAGE. Here, the new feature representation of a node i is defined as follows:*

$$x_i' = W_1 x_i + W_2 \cdot \mathbb{E}_{j \in \mathcal{N}_i}, \tag{6}$$

where W_1 and W_2 denote learnable weights and $\mathbb{E}_{j \in \mathcal{N}_i}$ the expectation over all node features in the neighbourhood of j.

Definition 2.7 (Higher-order Graph Neural Networks (GraphCONV) (Morris et al., 2019)) *Morris et al. (2019) introduced so-called higher-order GNNs, where the node feature embedding x_i' of node i is defined as follows:*

$$x_i' = W_1 x_i + W_2 \sum_{j \in \mathcal{N}_i} x_j, \tag{7}$$

where W_1 and W_2 are learable weights and $\mathcal{N}_i$ denotes the neighbourhood of node i.

Definition 2.8 (Graph Attention Networks (GAT) (Veličković et al., 2017)) *Veličković et al. (2017) introduced a graph neural network, that learns attention weights for edges in the graph. The new node feature embedding of a node i is defined as:*

$$x_i' = \alpha_{ii} \Theta x_i + \sum_{j \in \mathcal{N}_i} \alpha_{ij} \Theta x_j, \tag{8}$$

where Θ are learable parameters and α_{ij} is the attention coefficient between two nodes i and j and is defined as follows:

$$\alpha_{ij} = \frac{\exp\left(\phi\left(a^T\left(\Theta x_i \parallel \Theta x_j\right)\right)\right)}{\sum_{k \in \mathcal{N}_i \cup i} \exp\left(\phi\left(a^T\left(\Theta x_i \parallel \Theta x_k\right)\right)\right)}, \tag{9}$$

where ϕ is commonly the `LeakyReLU` function and $\parallel$ indicates a concatenation of the values.

2.4 Neural Sheaf Diffusion Models

With a rising discussion on how GNNs perform on low-homophily graph structures, different approaches to graph learning have been established that target these more challenging settings for graph learning. One of these methods is neural sheaf diffusion models, originally introduced by Hansen & Gebhart (2020) and extended by Bodnar et al. (2022). They use the topological concept of cellular sheaves, which assign vector spaces to all nodes and edges and linear mappings between them for all node-edge connections. Traditional GNNs are designed in a way that they assume a graph structure with a trivial underlying sheaf. Hansen & Gebhart (2020) and Bodnar et al. (2022) introduce an alternative approach to graph deep learning that is based on the concept of cellular sheaves, where different sheaf representations are learned for nodes and edges of the graph. They show that with this method, they can provide a graph learning technique that is less impacted by heterophilic graphs and over-smoothing - two commonly known limitations of GNNs. Sheaf neural networks (Hansen & Gebhart, 2020; Bodnar et al., 2022) are a generalisation of GCNs (Kipf & Welling, 2016) and leverage the sheaf Laplacian (Hansen & Ghrist, 2019), an extension of the graph Laplacian. This allows for an expression of more complex relationships between nodes rather than "similarity". Bodnar et al. (2022) furthermore show how these sheaves can be learned from the data at hand, using neural networks.

Definition 2.9 (Sheaf Convolution) *Let $\mathcal{F}$ be a sheaf on a graph G with feature matrix $X \in \mathbb{R}^{nd \times a}$ and sheaf Lapacian $\Delta_{\mathcal{F}}$. A sheaf convolutional model is then defined as follows:*

$$Y = \sigma\left(\left(I_{nd} - \Delta_{\mathcal{F}}\right)\left(I_n \otimes W_1\right) X W_2\right), \tag{10}$$

where σ is a non-linearity, $\otimes$ denotes the Kronecker product, $W_1 \in \mathbb{R}^{d \times d}$ and $W_2 \in \mathbb{R}^{a \times b}$ are two weight matrices, and a and b define the number of input and output channels, respectively.

The authors introduce different versions of neural sheaf networks, such as *GeneralSheaf*, *BundleSheaf*, and *DiagSheaf*. For more details about sheaf networks, we refer to Hansen & Gebhart (2020) and Bodnar et al. (2022). In this work, we utilise neural sheaf diffusion models on all classification datasets in order to investigate their potential on potentially low-homophily graph structures of medical population graphs.

3 Related Work

Medical population graphs have been used for several different downstream tasks, such as disease prediction (Parisot et al., 2017; Kazi et al., 2019; 2022) or age prediction (Bintsi et al., 2023a;b). Given that the construction of the graph itself is a major challenge when working with population graphs, several methods for graph construction have been established, which we utilise and compare in this work. For example, dynamic graph learning (Cosmo et al., 2020; Kazi et al., 2022) has been established to allow for end-to-end learning of the graph structure, so the graph does not have to be defined manually. There is little work investigating the impact of different graph construction methods and different graph learning schemes on the performance of population graphs. Bintsi et al. (2023b), for instance, evaluate different static graph construction methods on an age regression dataset but do not evaluate dynamic graph construction methods. To the best of our knowledge, this is the first work specifically addressing the challenge of graph construction in population graph studies in combination with different graph learning methods and with a detailed comparison to baseline models.

In general GNN research, several works have investigated the impact of the graph structure on model performance. Zhu et al. (2020) address the issue of the impact of the graph structure, measured by homophily (see Section 2.2), on different graph convolutional networks on citation networks. Several metrics have been established that allow for an assessment of the graph structure and show a correlation with the performance of GNNs. Luan et al. (2022) introduce two metrics, normalised total variation and normalised smoothness value, that measure the effect of edge bias. Xie et al. (2020) measure the graph structure with two metrics called neighbourhood entropy and centre-neighbourhood similarity. Ma et al. (2021) utilise the above-mentioned metric called cross-class neighbourhood similarity, which assesses how similar all neighbourhoods of all nodes with the same label are and show their correlation with GNN performance. Most of these works assess their metrics on benchmark datasets, such as citation networks, that come with a ground truth graph structure. In this work, we want to take these experiments one step further and investigate the impact of graph construction methods on population graph studies with GNNs and investigate the benefit of using GNNs over baseline methods.

4 Methods and Training Setup

In this section, we provide an overview of the utilised methods in this work. We introduce the different static and dynamic graph construction methods, summarise the utilised GNN models and the training setup, and introduce the datasets that were used to perform the experiments. A summary of the different learning and graph construction pipelines is visualised in Figure 2.

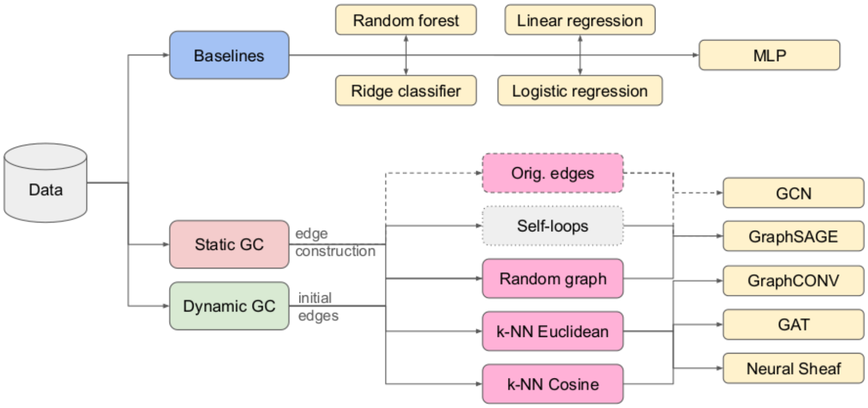


Figure 2: **Overview of the conducted experiments.** We tune different baselines and compare their performance to GNNs on population graphs. We perform static and dynamic graph construction (GC) and use four graph convolutions: GCN, GraphSAGE, GraphConv, and GAT, and Neural Sheaf Models. The original edges are only used if available, and self-loops mimic a transductive learning setting (see appendix).

4.1 Datasets

We perform our experiments on five medical population graph datasets, which are summarised in Table 1. First, we use the commonly used subset of the **TADPOLE** dataset (Yu et al., 2020) that is, for example, used in Kazi et al. (2022). The task of this dataset is to distinguish between patients with Alzheimer's disease (AD), ones with mild cognitive impairment (MCI), and healthy control groups (NC). The dataset consists of 30 imaging features of 564 subjects. A second public and frequently used dataset for population graph studies is the Autism Brain Imaging Data Exchange (**ABIDE**) dataset (Di Martino et al., 2014). It contains brain imaging features and clinical features such as age of 871 subjects and has been used in the context of population graphs in several works (Parisot et al., 2017; Kazi et al., 2019; 2022). The task of this dataset is a binary classification task, discriminating between autism patients and healthy controls. Furthermore, we use a small real-world medical dataset of **COVID** patients that has also been used before in population graph settings (Keicher et al., 2021); however, in a slightly different version of the dataset. The task is a binary classification of whether a subject is predicted to require intensive care or not. The dataset consists of image-derived features and clinical features of 65 subjects. Additionally, we use a larger population graph dataset from the UK Biobank (UKBB) (Sudlow et al., 2015) that consists of features extracted from brain magnetic resonance (MR) images (**UKBB brain age**). To extract the features, we follow the approach from Cole (2020), resulting in 68 imaging features and 20 non-imaging features for each subject. We use a set of 6406 subjects and perform a regression task for age prediction on this dataset. The mean age of this dataset is 62.86 years. We use this dataset to explore the difference in model performance when only using the imaging features compared to using all features. If not specifically specified, we only use the 68 imaging features. We extract another dataset from the UKBB (Sudlow et al., 2015) containing imaging features from cardiac MRIs as well as clinical features, on which we perform a binary classification of whether a subject suffers from cardiovascular diseases or not (**UKBB cardiac**). We extract 6 non-imaging features and 86 imaging features using the pipeline from Bai et al. (2020) and create a population graph with 2900 subjects.

Table 1: **Overview of all utilised population graph datasets** with the respective number of nodes, number of samples/nodes in the train, test, and validation sets, the number of node features (Nr. features), and the number of classes.

Dataset	Nr. nodes	Train samples	Val. samples	Test samples	Nr. features	Nr. classes
TADPOLE	564	468	48	57	30	3
ABIDE	871	609	41	221	6105	2
UKBB cardiac	2900	2320	58	522	89	2
COVID	65	45	4	16	29	2
UKBB brain age	6406	4811	1276	319	88	Regression

In order to evaluate the impact of the graph construction method and the resulting graph structure on the performance of the GNN, we also utilise three benchmark citation datasets: **CORA**, **CITESEER**, and **PUBMED** (Yang et al., 2016). These datasets come with a pre-defined graph structure, which we can use as the ground truth graph and compare performance to our generated graph structures. In Section 5.4, we also evaluate the impact of scale with a **synthetically** generated classification dataset with 4 classes and between 5 000 and 30 000 nodes.

4.2 Graph Construction Methods

We use distinct graph construction methods for population graphs and compare their impact on the performance of different GNNs. We note that the utilised methods are not extensive, but we picked the most representative, most frequently used, and well-established methods for static and dynamic graph construction. For more details on graph construction methods for GNNs in medicine, we refer to Mueller et al. (2024).

4.2.1 Static Graph Construction

Static graph construction methods refer to the construction of a graph structure that stays constant throughout GNN training. There are several methods to construct a static population graph structure, while the most common one utilises a k-nearest neighbour approach (Cunningham & Delany, 2021).

Self-loops Only To get an intuition about the impact of the graph structure on the GNN, we evaluate a GNN on a graph that is not really a graph but only contains self-loops. The adjacency matrix of a graph that only contains self-loops is equivalent to the identity matrix. In this setting, no message passing among nodes is performed since there are no connections between nodes. We use this setting to simulate a transductive learning setting without using a graph structure.

Random Graph Secondly, we construct a random graph structure by generating an Erdos-Rényi Graph with an edge probability of 0.001. We choose to evaluate all methods applied to a graph with a random graph structure in order to investigate the impact of the graph structure on model performance.

k-Nearest Neighbour Graph The most frequently used approach of graph construction for population graphs is the k-Nearest Neighbour (k-NN) approach. Here, k is a hyperparameter and defines the number of neighbours each node has. For this approach, different distance measures can be used, for example, the Euclidean distance or the cosine similarity. We use the implementation of `knn_graph` from Pytorch Geometric (Fey & Lenssen, 2019) and refer to the usage of the Euclidean distance as "k-NN Eucl." and the usage of the cosine similarity as "k-NN Cosine" in the tables below.

4.2.2 Dynamic Graph Construction

Dynamic graph construction refers to the learning of the graph structure in an end-to-end manner in parallel to the model training. There exist a few dynamic graph construction methods; however, for population graphs, mostly the approach from Kazi et al. (2022) is used. Here, we use the dDGM method, a differentiable graph construction method that allows for end-to-end learning of the graph structure during GNN training. In their work, Kazi et al. (2022) propose two differentiable graph learning modules: cDGM and dDGM. We here only use the dDGM implementation since both in their work and in our preliminary results and related works like (Mueller et al., 2023a), dDGM resulted in better performance. The dDGM module can be applied to arbitrary initial graph structures. We evaluate the impact of the initial graph structure on the model performance by using different graphs as a starting point. For the CORA dataset, we evaluate dDGM starting with **(a)** no edges, **(b)** only self-loops, **(c)** a random graph structure, **(d)** a k-NN graph, and **(e)** the original edges of the dataset, in Section 5.4.

4.3 Graph Assessment

In order to gain insights into the constructed graph structures and investigate their "quality", we evaluate two graph assessment metrics: node homophily (Pei et al., 2020) and cross-class neighbourhood similarity (CCNS) (Ma et al., 2021). We follow the approach from Mueller et al. (2023a) and evaluate the *CCNS distance*, the there-defined homophily for regression tasks, and split the evaluation of all metrics into train and test nodes. The latter can be useful to investigate how differently the graph structure impacts training and test nodes.

4.4 Model Architectures and Training

We use two different model architectures in our experiments. For all dynamic graph construction experiments, we use the architecture proposed by Kazi et al. (2022), which consists of two graph convolutional networks: a graph embedding function f and a diffusion function g. Following the results from the original paper (Kazi et al., 2022), we use the respective graph convolutions for both modules. For the static graph construction experiments, we use a GNN with 1, 2, or 3 graph convolutional layers (e.g. GCN or GraphSAGE), followed by an MLP. We use two sets of hyperparameters regarding the layers of these networks that can be found in the Appendix. During preliminary experiments, we noticed that using the same architecture for static graph construction results in strong over-fitting of the models to the training sets. We, therefore, use a different architecture for the static graph construction experiments than for the dynamic ones. More details about all architectures can be found in the appendix. In all architectures, we utilise four different frequently used graph convolutions, namely graph convolutional networks (GCNs) (Kipf & Welling, 2016), graph SAGE networks (Hamilton et al., 2017), higher-order GNNs (GraphConv) (Morris et al., 2019), and graph attention networks (GATs) (Veličković et al., 2017). They all differ in the methodology of how the message-passing scheme is performed and their formal definitions can be found in Section 2.3. For the neural sheaf diffusion models, we utilise the setup of the original work, varying between the following sheaf models: *BundleSheaf*, *DiagSheaf*, and *GeneralSheaf*.

All models are trained in a transductive setting, where all nodes are available during training. We define a fixed set of hyperparameters for all experiments and run a hyperparameter search for at least 200 runs using sweeps from *Weights and Biases* (Biewald, 2020). We then pick the run with the best validation accuracy/MAE, evaluate its performance over 5 random seeds, and report the mean test accuracy with the standard deviation. All trainings are performed on an Nvidia Quadro RTX 8000 GPU, using `Pytorch lightning` and `Pytorch Geometric` (Fey & Lenssen, 2019). The hyperparameters can be found in the appendix.

5 Experiments and Results

In this section, we summarise our experiments with different graph construction methods, including static and dynamic graph construction and Neural Sheaf Diffusion models. We (1) summarise the overall best-performing GNNs for all datasets and compare them to three different baselines and discuss more detailed results on two of the medical population graph datasets, (2) compare our results to different state-of-the-art (SOTA) population graph studies, (3) evaluate the method of population graphs for multi-modal data integration, and (4) evaluate the impact of the different components –such as graph structure, dataset complexity and size– on the performance of GNNs for population graphs.

The most noteworthy finding of our work is possibly the fact that simple baseline methods outperform more complex graph learning techniques on all tested population graph datasets.

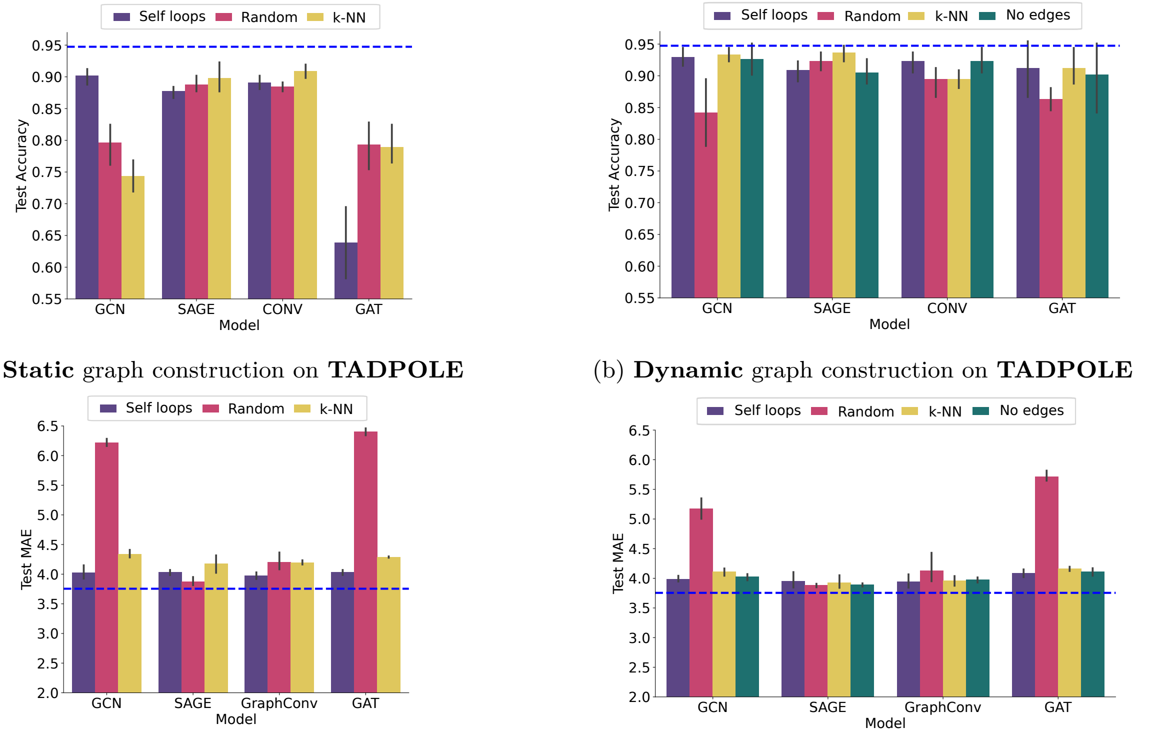
5.1 Baselines Achieving Comparable Performance to GNNs

During an extensive evaluation of the performance of GNNs on medical population graphs, we found that when optimally tuning baseline models (random forest, linear/logistic regression, and ridge classifier/regression) they perform competitively on all datasets. We summarise these results, the best GNN as well as a Neural Sheaf Diffusion model in Table 2, where the best model for each dataset is highlighted in bold.

Table 2: **Summary of results** of different baseline methods and the best GNNs and Neural Sheaf Models, either from our training evaluated on 5 random seeds or from literature ([1]: Parisot et al. (2017)). For classification datasets, we report the test accuracy; for regression tasks, the test MAE.

Method	TADPOLE	UKBB Brain Age	UKBB Cardiac	COVID	ABIDE
Random forest	**0.9474 ± 0.00**	3.7913 ± 0.01	**0.7061 ± 0.01**	0.8250 ± 0.02	**0.7046 ± 0.01**
Ridge	0.7368 ± 0.00	3.4185 ± 0.00	0.6935 ± 0.00	**0.8750 ± 0.00**	0.7014 ± 0.00
Linear/Logistic	0.8421 ± 0.00	3.4287 ± 0.00	0.6858 ± 0.00	0.8125 ± 0.00	0.6290 ± 0.00
GNN k-NN	0.9404 ± 0.02	**3.3524 ± 0.06**	0.6970 ± 0.02	0.7875 ± 0.03	0.695 [1]
Neural Sheaf	0.9368 ± 0.02	-	0.6904 ± 0.01	0.8000 ± 0.03	0.5448 ± 0.01

It is noteworthy that for all population graph datasets apart from the UKBB brain age dataset, at least one of the baseline methods outperforms the best GNN model. On the UKBB brain age dataset, the GNN slightly outperforms the ridge regression (best baseline) by an MAE of 0.066. However, a two-sided t-test between the results of the best GNN and the strongest baseline (ridge regression) did not show a significant difference in performance with a p-value of 0.06. These results raise the main question of this work: *"Are population graphs really as powerful as believed?"* Our results indicate the contrary, and we investigate the discrepancy between our work and related works in the following sections, discussing potential reasons for this gap.



(a) **Static** graph construction on **TADPOLE**

(b) **Dynamic** graph construction on **TADPOLE**

(c) **Static** graph construction on **UKBB brain age**

(d) **Dynamic** graph construction on **UKBB brain age**

Figure 3: **Results on two datasets** with static graph construction (left column) and dynamic graph construction (right column). First row: **TADPOLE** reporting the test accuracy (higher better), second row: **UKBB brain age**, reporting the test MAE (lower better). The mean performance of the baseline is indicated by the dashed blue lines.

In the following, we evaluate the experiments summarised in Figure 2 on the two population graph datasets TADPOLE and UKBB brain age in more detail. The results are visualised in Figure 3, where the first row shows the TADPOLE dataset and the second row the UKBB brain age dataset. The results are also listed in Tables 3 and 4, respectively. For the TADPOLE dataset, none of the GNNs outperform the best baseline method, which in this case is a random forest. This is even the case in settings where the homophily of the test set is very high, for example, for the static k-NN graph construction and the GAT convolution. We observe similar results on the UKBB brain age dataset, where we perform age regression on the imaging features only and report the MAE as model performances. We do not report the CCNS values for this dataset since CCNS is not defined for regression tasks. GraphSAGE and GraphConv networks do not seem to be influenced by the randomness of the graph structure and are still able to learn meaningful representations of the node features and make accurate predictions. The homophily of the k-NN graphs generated for the UKBB dataset is also quite high, similar to the TADPOLE dataset. The same holds for its low CCNS distance score. Furthermore, we observe that GCN models tend to perform better at a lower number of neighbours. Interestingly, the best-performing GNN on the TADPOLE dataset is trained on a random graph structure, using GraphSAGE convolutions and dynamic graph construction. We also cannot see a clear benefit of using dynamic graph construction methods on all datasets. While the best dynamic result outperforms the best static result on the TADPOLE dataset, static methods achieve higher results on the UKBB brain age dataset. We observe the same behaviours on all other datasets Their results can be found in the appendix.

Table 3: Results of the experiments on the **TADPOLE** dataset. GC: graph construction, BL: baselines, k: number of neighbours. The best performance for each method is bold.

	Initial edges	Model	k	Test acc ↑	Test homophily ↑	Test CCNS distance ↓
BL	-	Majority vote	-	0.5674 ± 0.00	-	-
	-	Random forest	-	**0.9474 ± 0.00**	-	-
	-	Logistic regression	-	0.8597 ± 0.00	-	-
Static GC	Random	GCN	-	0.7965 ± 0.04	0.426 ± 0.49	0.348
		SAGE	-	0.8877 ± 0.01	0.426 ± 0.49	0.348
		GraphConv	-	0.8842 ± 0.01	0.426 ± 0.49	0.348
		GAT	-	0.7930 ± 0.04	0.426 ± 0.49	0.348
	k-NN Euclidean	GCN	5	0.7439 ± 0.03	0.775 ± 0.24	0.213
		SAGE	5	0.8982 ± 0.03	0.775 ± 0.24	0.213
		GraphConv	5	**0.9088 ± 0.01**	0.775 ± 0.24	0.213
		GAT	2	0.7895 ± 0.04	**0.904 ± 0.20**	**0.094**
Dynamic GC	No edges	GCN	20	0.9263 ± 0.03	**0.919 ± 0.19**	0.073
		SAGE	20	0.9053 ± 0.02	0.806 ± 0.21	0.183
		GraphConv	2	0.9228 ± 0.02	0.798 ± 0.34	0.190
		GAT	20	0.9018 ± 0.06	0.908 ± 0.15	**0.101**
	Random	GCN	2	0.8421 ± 0.06	0.851 ± 0.27	0.177
		SAGE	10	0.9228 ± 0.02	0.423 ± 0.22	0.616
		GraphConv	5	0.8947 ± 0.03	0.411 ± 0.25	0.594
		GAT	5	0.8632 ± 0.02	0.895 ± 0.20	0.119
	k-NN Euclidean	GCN	2	0.9333 ± 0.01	0.793 ± 0.28	0.204
		SAGE	20	**0.9368 ± 0.01**	0.461 ± 0.63	0.632
		GraphConv	10	0.8947 ± 0.02	0.777 ± 0.29	0.219
		GAT	10	0.9123 ± 0.03	0.775 ± 0.29	0.206

Table 4: Results of the experiments on the **UKBB brain age** imaging dataset. BL: baselines, k: number of neighbours, GC: graph construction. The best performance for static and dynamic graph construction and the highest homophily is **bold**.

	Initial edges	Model	k	Test MAE ↓	Test homophily ↑
BL	-	Mean prediction	-	6.4090 ± 0.00	-
	-	Random Forest	-	4.1424 ± 0.01	-
	-	Linear Regression	-	3.7545 ± 0.00	-
Static GC	Random	GCN	-	6.2158 ± 0.07	0.742 ± 0.10
		SAGE	-	**3.8764 ± 0.08**	0.742 ± 0.10
		GraphConv	-	4.2029 ± 0.16	0.742 ± 0.10
		GAT	-	6.4034 ± 0.07	0.742 ± 0.10
	k-NN Euclidean	GCN	2	4.3351 ± 0.07	**0.916 ± 0.07**
		SAGE	10	4.1780 ± 0.17	0.844 ± 0.06
		GraphConv	2	4.1979 ± 0.04	**0.916 ± 0.07**
		GAT	20	4.2888 ± 0.01	0.834 ± 0.06
Dynamic GC	No edges	GCN	2	4.0257 ± 0.06	**0.865 ± 0.10**
		SAGE	5	3.8882 ± 0.03	0.754 ± 0.10
		GraphConv	5	3.9741 ± 0.05	0.840 ± 0.08
		GAT	2	4.1071 ± 0.07	0.843 ± 0.11
	Random	GCN	2	5.1712 ± 0.20	0.834 ± 0.13
		SAGE	10	**3.8811 ± 0.04**	0.780 ± 0.09
		GraphConv	10	4.1248 ± 0.30	0.768 ± 0.09
		GAT	2	5.7138 ± 0.10	0.831 ± 0.14
	k-NN Euclidean	GCN	2	4.1109 ± 0.07	0.849 ± 0.11
		SAGE	20	3.9226 ± 0.13	0.842 ± 0.07
		GraphConv	2	3.9560 ± 0.09	0.831 ± 0.11
		GAT	2	4.1603 ± 0.04	0.837 ± 0.11

5.2 Comparison to Other Published Results

With these results, the question arises as to why population graphs have been believed to improve the performance of medical downstream tasks. We compare our results to published results in the most closely related works, investigating the different performances of baseline models and GNNs on different datasets. We compare all datasets that have been used in related works: TADPOLE, ABIDE, and UKBB brain age datasets. The related works we pick for comparison are works introducing the concept of population graphs (Parisot et al., 2017), as well as new graph learning techniques that have been applied to or designed for population graph studies (Kazi et al., 2019; 2022; Bintsi et al., 2023a). The results are summarised in Table 5. All our baselines outperform the published baselines in the related works, while our GNN implementations match the performances reported in the respective works. This corroborates our hypothesis that our implementation is on par with previously reported works, while these works seem to underestimate the baseline performance.

Table 5: Comparison of our results to results from related works: Parisot et al. (2017) [1], Kazi et al. (2022) [2], Kazi et al. (2019) [3], and Bintsi et al. (2023a) [4]. The overall best result for each dataset is underlined. The baseline for the UKBB brain age dataset is a ridge regression for our work and a linear regression for the results from Bintsi et al. (2023a); for the TADPOLE dataset: Linear classifier for results from Kazi et al. (2022), random forest for our results; for ABIDE: Ridge regression for results from Parisot et al. (2017), random forest for our results. All our baselines outperform reported baselines in other works, while our GNN implementations match performance.

Dataset	Score	Method	Convolution	Other reported results	Our results
TADPOLE	Accuracy ↑	Baseline	-	0.7022 ± 0.06 [2]	**0.9474 ± 0.00**
		dDGM [2]	GCN	**0.9414 ± 0.02** [2]	0.9333 ± 0.01
		InceptionGCN [3]	InceptionGCN	0.8435 ± 0.07 [3]	-
UKBB Brain Age	MAE ↓	Baseline	-	3.82 [4]	**3.5063 ± 0.00**
		dDGM [2]	GCN	**3.72** [4]	3.8287 ± 0.03
		dDGM [2]	SAGE	-	**3.5034 ± 0.06**
		adaptive [4]	GCN	3.62 [4]	-
ABIDE	Accuracy ↑	Baseline	-	0.668 [1]	**0.7040 ± 0.01**
		Similarity Score [1]	GCN	0.695 [1]	-
		InceptionGCN [3]	InceptionGCN	0.6923 ± 0.07 [3]	-

The discrepancy in baseline performance can partially be due to different models, different hyperparameters, or the utilisation of only a subset of the features for the evaluation of the baselines. Some works, for example, only use the node features of the GNN as input for the baseline, while using additional features for the edge construction of the population graph. We deem this to be an unfair comparison and always use all features that we use for graph construction and as node features as input for the baseline. For the evaluation of the baseline methods on the benchmark citation network datasets, we use only the node features of the graphs since the edges cannot be incorporated in the same feature vector in a straightforward way. Some works do not specify on which features the baseline is evaluated (Parisot et al., 2017).

5.3 Population Graphs for Multi-Modal Data Integration

One highly emphasised advantage of population graphs is their utilisation for multi-modal data integration (Parisot et al., 2017; Zheng et al., 2022; Keicher et al., 2021). In one of the first utilisations of population graphs (Parisot et al., 2017), for instance, a graph construction method is introduced that uses clinical features to generate the edges between subjects, while image-derived features are used as node features in the graph. In later approaches, especially for dynamic graph construction, methods moved away from a clear separation between clinical and image-derived features (Kazi et al., 2022). In this so far typical setting of population graphs, we scrutinise this claimed advantage and argue that all available features can easily be appended and, therefore, incorporated into the node features. However, we see exceptions when the information used for edge construction cannot be used as node features. This is the case when high dimensional data is used as node features –e.g. text, audio data or images. However, this setup comes with large memory requirements and has not been studied in detail. We encourage a more critical assessment of the utilisation of GNNs for multi-modal data integration in conventional configurations of population graphs and advocate a shift towards more advanced settings and a more suitable usage of multi-modal data integration for cases where it is indeed beneficial.

We perform several experiments investigating whether GNNs are useful for multi-modal data integration for population graphs. We take the two UKBB datasets and evaluate the performance of GNNs with different combinations of imaging and non-imaging features for graph construction and as node features. The results are summarised in Table 6. Given that the convolutions GraphSAGE and GraphConv performed best in our previous experiments on population graphs, we limit these results to those two convolutions. The best performing GNN is highlighted in **bold**, the second best in purple, and the third best in green. The corresponding homophily values for each graph structure for both datasets are summarised in Table 7. For these experiments with static graph construction, we experiment with a different model architecture consisting of only one graph convolutional layer, followed by an MLP.

We observe that for the brain age dataset, the best GNN is the one that uses all available features as node features and for edge construction. The second and third-best GNNs also use all features as node features. For the cardiac dataset, the best and second-best models also use all features as node features. However, the third-best model uses only the imaging features as node features and the non-imaging features for edge construction. Furthermore, on the UKBB brain age dataset, some GNNs outperform the respective baseline (which only uses the node features) by small margins. This is not the case for the cardiac dataset. Here none of the GNNs outperform the respective baselines. Interestingly, on the UKBB brain age dataset, the static graph construction results in better performance than dynamic graph construction, which is the opposite for the cardiac dataset. We can also see that the node features slightly dominate the prediction, such that the performance of the GNN somewhat matches the performance of the baseline that uses the node features only. This is reasonable since the specific features used for edge construction are reduced into a simple "measure of similarity". However, overall, the baselines perform on par with the GNNs.

The graph metrics for the experiments are summarised in Tables 3 and 4. The homophily values of the graph structures constructed from different combinations of image and non-image features for the UKBB brain age and cardiac dataset are summarised in Table 7. We can see that for both datasets, all graph structures have similar homophily values, which might be why the performance of all graph structures is very similar when using all node features.

Table 6: Results of different combinations of image-derived and non-imaging features as node features and for graph construction on the UKBB brain age and cardiac datasets. For the age prediction dataset, the baseline is a ridge regression, and for the cardiac dataset, a random forest. GNN outperforms their corresponding node-feature-baseline are underlined. Best GNN: **bold**, second best GNN: purple, third best GNN: green. All scores are evaluated on the test set.

	Features		Model	UKBB Brain Age Test MAE ↓		UKBB Cardiac Test accuracy ↑	
Baseline	-		Naive baseline	6.4090		0.5000	
	Non-imaging		Best baseline	4.6509 ± 0.00		0.6678 ± 0.00	
	Imaging			3.5063 ± 0.00		0.6969 ± 0.01	
	All			3.4185 ± 0.00		**0.7046 ± 0.01**	
	Node Feat.	**(Initial) Edges**	**Model**	**dDGM MAE ↓**	**Static MAE ↓**	**dDGM acc. ↑**	**Static acc. ↑**
Graph Neural Networks	All	All	GraphSAGE	3.5034 ± 0.06	3.4351 ± 0.00	0.6816 ± 0.01	0.6609 ± 0.02
			GraphConv	3.5407 ± 0.04	**3.3524 ± 0.06**	0.6785 ± 0.01	0.6705 ± 0.01
	All	Imaging	GraphSAGE	3.5471 ± 0.02	3.4249 ± 0.00	**0.6839 ± 0.01**	0.6739 ± 0.01
			GraphConv	3.5221 ± 0.03	3.3758 ± 0.05	0.6690 ± 0.01	0.6743 ± 0.01
	All	Non-imaging	GraphSAGE	3.5317 ± 0.04	3.4175 ± 0.00	0.6724 ± 0.01	0.6632 ± 0.01
			GraphConv	3.6792 ± 0.25	3.4330 ± 0.01	0.6751 ± 0.01	0.6644 ± 0.02
	Imaging	Imaging	GraphSAGE	3.9226 ± 0.13	3.7716 ± 0.04	0.6743 ± 0.01	0.6705 ± 0.00
			GraphConv	3.9560 ± 0.09	3.8368 ± 0.00	0.6632 ± 0.01	0.6628 ± 0.01
	Imaging	Non-imaging	GraphSAGE	3.9130 ± 0.05	3.6791 ± 0.01	0.6567 ± 0.01	0.6785 ± 0.00
			GraphConv	3.9835 ± 0.01	3.7099 ± 0.04	0.6805 ± 0.01	0.6483 ± 0.01
	Non-imaging	Imaging	GraphSAGE	4.6767 ± 0.06	4.9382 ± 0.00	0.6755 ± 0.01	0.6521 ± 0.01
			GraphConv	4.0376 ± 0.12	5.0410 ± 0.02	0.6579 ± 0.01	0.6452 ± 0.01

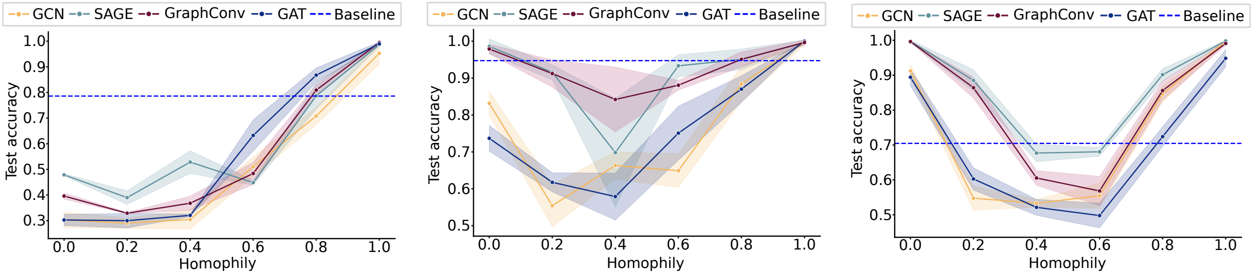
5.4 Further Components of Impact on Model Performance

In this section, we investigate the impact of the graph structure on model performance from three further viewpoints. (1) The experiments above have indicated that the graph structure has a different impact on different graph convolutions, (2) the complexity of the dataset plays an important role in the performance of GNNs on low-homophily graphs, and (3) if a meaningful graph structure is available, GNNs out-perform

Table 7: Homophily values of the **UKBB brain age** and **cardiac** datasets with $k = 5$ and k-NN graph construction, when using all features, only imaging, or only non-imaging features for graph construction.

Dataset	Features	Homophily
Brain Age	All	0.8571 ± 0.07
	Imaging	**0.8619 ± 0.07**
	Non-imaging	0.8237 ± 0.08
Cardiac	All	0.6404 ± 0.22
	Imaging	0.6396 ± 0.22
	Non-imaging	**0.6649 ± 0.23**

graph-agnostic models. Therefore, we perform additional experiments on synthetically generated graph structures at different homophily values. Here, the graph is constructed statically and to specifically match a certain homophily value by using the labels and connecting each node to a specific number of same and differently labelled neighbours. The results for three datasets are visualised in Figure 4, and more visualisations can be found in the appendix in Figure 6.



(a) **CORA** dataset

(b) **TADPOLE** dataset

(c) **ABIDE** dataset

Figure 4: Performance of different graph convolutions on synthetic graph structures with different homophily values on (a) the **CORA** dataset, (b) the **TADPOLE** dataset, and (c) the ABIDE dataset. The dashed blue line indicates the mean performance of the best baseline for each dataset.

Different Types of Graph Convolution Zhu et al. (2020) have shown interesting correlations between homophily and different graph convolutions. They showed that the separate handling of node features of the node of interest (x_i) and its neighbouring nodes ($\mathcal{N}_i$) improves the performance of GNNs on heterophilic graphs. The same accounts for networks that evaluate the k-hop neighbourhoods separately. Graph convolutional networks (GCNs) (Kipf & Welling, 2016) do not separate node features of i and $\mathcal{N}_i$, but average the message passing over both in one step (Equation 5). GraphSAGE and GraphConv, on the other hand, distinguish between x_i and $x_j, j \in \mathcal{N}_i$ (Equations 6 and 7). GAT (Equation 8) learns different attention coefficients for x_i and $x_j, j \in \mathcal{N}_i$. However, the network weights are shared for both, which might negatively impact performance on graphs with low homophily. Our experiments support these findings. We observe that the graph structure strongly affects GCN and GAT, whereas GraphSAGE or GraphConv networks perform more consistently across different graph structures.

Impact of Dataset Complexity The impact of the homophily on the model performance is not only dependent on the graph convolution but also varies depending on the dataset, probably related to the number of classes in the dataset as well as class imbalance. In order to investigate this, we perform experiments with synthetic graph structures on the TADPOLE dataset (3 classes), the CORA dataset (7 classes), the UKBB cardiac dataset (2 classes), and the ABIDE dataset (2 classes) at different synthetically generated homophily values. Figure 4 shows the performance of different graph convolutions on 3-layer GNNs using synthetically generated graphs for the different datasets. For the CORA dataset (Figure 4a), all models perform worse than the baseline with homophily values lower than 0.8. While all graph convolutions are impacted similarly and perform worse than the baseline for low-homophily graphs, SAGE and GraphConv perform better than GAT and GCN. The low-homophily graphs do not allow the model to learn meaningful node feature embeddings since, during the course of training, node features of differently labelled nodes are averaged and shared, interfering with the model's goal to distinguish different classes. Interestingly, the performance for the TADPOLE dataset (Figure 4b) looks different. We observe similar differences between the graph convolutions. However, we also observe that only at very high and very low homophily values can the GNN outperform the baseline. Everything in between either matches the performance of the baseline or reaches a worse performance. When we now compare the homophily values of the generated graph structures in our experiments on the TADPOLE dataset above, we can see that most of them have a homophily of around 0.7 or 0.8. The ABIDE dataset requires a graph structure with lower homophily to outperform the baseline. However, the same pattern holds that all population graphs constructed in our experiments reached homophily values in the range where the GNNs under-perform or perform on par with the baselines. This potentially explains why the population graphs do not outperform the graph-agnostic baseline models.

Furthermore, the high performance of the GNNs at low homophily values for the population graphs is highly different from that on the CORA dataset. We attribute this to the capability of the GNNs to learn the opposite labels from the majority of the neighbour labels, which we deem impossible for datasets with more classes. To investigate this further, we evaluate the attention of GAT models trained on graphs with different homophily values on the TADPOLE dataset and observe that low-homophily graphs (homophily=0.1) attribute high attention from differently labelled nodes and low attention to same-labelled nodes. The opposite is the case for high-homophily graphs. This allows the model to also perform well on low-homophily graphs on datasets with only a few classes. More details about these experiments can be found in appendix, Section C.3.

Impact of Dataset Size We investigate the impact of the graph size on model performance with two additional experiments: **(a)** We partition the largest population graph dataset –UKBB brain age– into smaller subsets (25%, 50%, and 75% of the original dataset) and **(b)** generate a synthetic dataset at different sizes and compare GNN performances to baselines. The results of the partitioned UKBB brain age dataset **(a)** and the synthetic dataset **(b)** are summarised in Table 8. For the GNN, we use the DGM adaptive graph construction with the k-NN initial graph structure and GraphSAGE convolutions. For the baseline, we use a linear regression for the UKBB brain age dataset and a 4-layer MLP for the synthetic dataset. We do not observe a tendency for the dataset size to have an impact on the difference in performance between the GNN and the baseline on the partitioned UKBB dataset. The same holds for the synthetic dataset, even for very large graphs with 30 000 nodes.

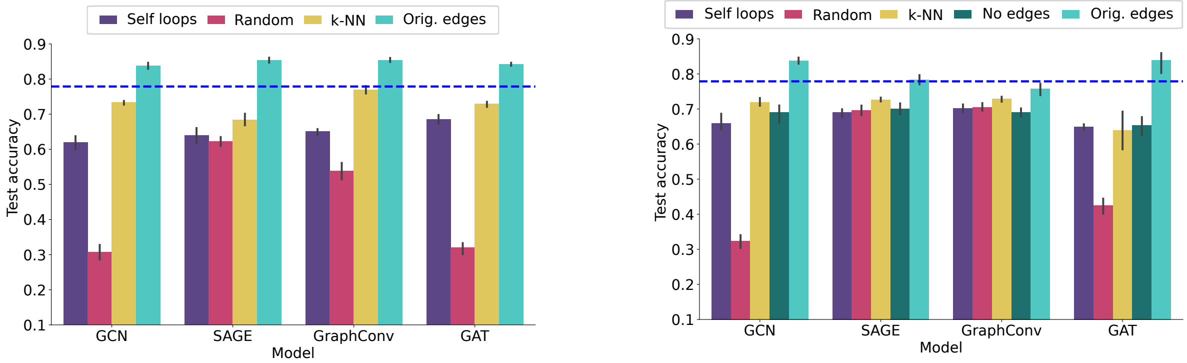
Dataset	Nr. Nodes	Score	GNN	Baseline	Performance Difference
UKBB brain age 25%	1841	MAE	3.6804 ± 0.08	**3.5577 ± 0.00**	-0.1227
UKBB brain age 50%	3362	MAE	3.7658 ± 0.18	**3.5363 ± 0.00**	-0.2295
UKBB brain age 75%	4884	MAE	3.7022 ± 0.11	**3.4651 ± 0.00**	-0.2371
UKBB brain age 100%	6406	MAE	3.5034 ± 0.06	**3.4185 ± 0.00**	-0.0849
Synthetic	5 000	ACC	0.7980 ± 0.03	**0.8164 ± 0.00**	0.0184
Synthetic	10 000	ACC	0.8100 ± 0.04	**0.8532 ± 0.00**	0.0432
Synthetic	20 000	ACC	0.8895 ± 0.01	**0.9072 ± 0.00**	0.0177
Synthetic	30 000	ACC	0.9179 ± 0.01	**0.9413 ± 0.00**	0.0234

Table 8: Performance differences between the baseline and GNNs for different subsets of the original dataset UKBB brain age (reported MAE) and a synthetically generated dataset (reported accuracy ACC). The column *Performance Difference* indicates the performance of the baseline minus the performance of the GNN.

Ground Truth Graph Structures Based on these results, we argue that the graph construction methods currently utilised for population graphs are insufficient. Only a meaningful graph structure that adds additional information to the node features leads to better performance of GNNs compared to baseline methods. To support this, we investigate commonly used graph construction methods for population graph studies on the frequently used benchmark citation datasets CORA, CITESEER, and PUBMED (Yang et al., 2016). They provide a "ground truth" graph structure, which we can evaluate in comparison to the graphs resulting from graph construction methods used for population graph studies. This allows us to investigate how the different graph construction methods perform compared to a given "ground-truth" adjacency matrix. The results of the best-performing GNNs and baselines on all three datasets are summarised in Table 9. The experiments on all benchmark citation network datasets have shown that GNNs can improve performance compared to simple baseline methods. However, even for the CITESEER dataset, a ridge classifier outperforms all GNN methods and neural sheaf diffusion networks. The results for the CORA dataset are also visualised in Figure 5. Only the usage of the original edges outperforms the baseline methods, while all static and dynamic graph construction methods yield poor results. This supports the hypothesis that the graph construction methods for population graphs do not add relevant information to the node features. More detailed results can be found in the appendix.

Table 9: **Summary of results on benchmark datasets** of different baseline methods and the best GNNs and Neural Sheaf Models.

Method	CORA	CITESEER	PUBMED
Random forest	0.7788 ± 0.00	0.7480 ± 0.01	0.7286 ± 0.01
Ridge	0.7860 ± 0.00	**0.7720 ± 0.00**	0.7350 ± 0.00
Linear/Logistic	0.5750 ± 0.00	0.5600 ± 0.00	0.7310 ± 0.00
GNN *k*-NN	0.7692 ± 0.01	0.6908 ± 0.01	0.6908 ± 0.01
GNN orig. edges	0.8540 ± 0.01	0.7548 ± 0.01	0.8760 ± 0.01 [1]
Neural Sheaf	**0.8730 ± 0.01** [3]	0.7714 ± 0.02 [3]	**0.8949 ± 0.00** [3]



(a) **Static** graph construction

(b) **Dynamic** graph construction

Figure 5: **Results on the CORA dataset** with static (left) and dynamic (right) graph construction.

6 Discussion

In this work, we evaluate the performance of medical population graphs on five population graph datasets and compare state-of-the-art graph learning techniques to well-tuned baseline models. We consistently observe the following findings:

1. **GCN and GAT are poorly suited for population graph studies.** GNNs using GraphSAGE and GraphConv convolutions consistently outperform GCN and GAT models, which leads to the conclusion that the latter methods are unsuitable for GNNs in population graph studies. We attribute this to the fact that GCN and GAT networks are highly affected by the graph structure, whereas GraphSAGE and GraphConv networks are more robust in this regard. This also manifests in the fact that GCN and GAT networks benefit more from dynamic graph construction than the other two convolutions and that GraphSAGE and GraphConv models can perform equally well on random graph structures.

2. **The utilisation of population graphs with the goal of multi-modal data integration might not be as promising as believed.** The most frequently used method for the construction of population graphs includes a separation of features into node features and ones utilised for edge construction. We show that using all available features for edge construction and as node features might lead to better results and argue that a concatenation of all features is easily doable –except

when using images as node features. We see potential in using population graphs in different settings where the connectivity information cannot easily be integrated with the node features.

3. **None of the state-of-the-art GNN methods significantly outperform well-tuned baseline methods** (see Table 2). This raises the question of whether population graphs –in the way they are currently used– have any benefit over graph-agnostic models. In Section 5.4, we investigate the interplay of the graph structure and the performance of the GNNs on a population graph dataset and conclude that only a nearly perfect graph structure leads to a better performance of GNNs compared to baseline models, which has not been possible with the current graph construction methods.

4. **Better graph construction methods are required.** The experiments on the benchmark datasets and the synthetically generated graph structures with different homophily values (see Figures 4 and 5) show that GNNs can improve downstream task performance if the graph structure is "meaningful". However, current graph construction methods do not lead to valuable graph structures, which makes graph construction the performance bottleneck in these settings. The same is represented by the fact that random graph structures often achieved comparable results to approaches like k-NN graphs.

We furthermore note that all baseline models are easy to implement using standard libraries such as `scikit-learn` (Pedregosa et al., 2011), are significantly faster to fit than the training of GNNs, and do not require extensive hyperparameter tuning.

7 Conclusion and Future Work

Medical population graphs were first introduced by Parisot et al. (2017) to allow for a population-wide representation of a cohort of patients. The idea behind the utilisation of population graphs is that subjects that share similar phenotypes (and are therefore neighbours in the population graph), also show similar pathologies. Thus, the neighbouring nodes are hoped to improve model performance when using graph deep learning methods. They have since then been combined with GNNs and used on multiple medical datasets. Most works utilise population graphs as a method for multi-modal data integration (Parisot et al., 2017; Kazi et al., 2019; Cosmo et al., 2020; Bintsi et al., 2023a). Here, a subset of the features are used as node features (usually imaging features), while other features (usually non-imaging) are used to generate the graph structure (the edges).

In this work, we perform an extensive study on how GNNs are used in the context of population graphs and compare different graph-learning methods to graph-agnostic baseline models. We use five medical population graph datasets, including all publicly available datasets used for population graph studies in related works. We utilise state-of-the-art (a) static graph construction methods, (b) dynamic graph construction methods, and (c) neural sheaf diffusion models. The latter have been designed to address two of the most dominant problems of GNNs: over-smoothing and performance on low-homophily graphs. We investigate the usage of neural sheaf diffusion models since the graph construction methods for population graphs seem to result in unideal graph structures, which might benefit from the use of neural sheaf diffusion models.

Even though we reach comparable results to related works on population graphs with GNNs for all methods, none of the GNNs significantly out-perform the strongest baseline method. This raises the question of how powerful population graphs indeed are and whether they are a suitable data representation combined with GNNs. We conclude that currently available graph construction methods are the performance bottleneck of GNNs on population graphs compared to graph-agnostic methods. We see a need for either more advanced methods to learn a graph structure that contains additional meaningful information to the node features or novel ideas on how to build population graphs from additional information that cannot be represented as node features. When using synthetically generated graph structures, we observe that only graphs with higher homophily than possible to extract from the node features result in better performance of GNNs compared to properly tuned graph-agnostic methods such as a random forest or linear regression (Figure 4). Even a dynamic graph construction method, which optimises the graph structure during training, does not reach a "good enough" graph structure. Also, models designed for "low-quality" graph structures (e.g. neural sheaf diffusion models) do not improve performance on population graphs. The fact that our baseline models outperform the results reported in related works emphasises the importance of appropriate tuning of baseline methods in general. It shows that the currently available graph construction methods for population graphs are insufficient.

There are a few more graph construction methods that we did not evaluate in this work, such as *Similarity Scores*. The first one was introduced by Parisot et al. (2017) and followed by several extensions and modifications (Ghorbani et al., 2022; Vivar et al., 2021; Pellegrini et al., 2022; Peng et al., 2022; Lu et al., 2022). In this work, we focus on using k-NN graphs since this method has been shown to achieve the best results in related works (Bintsi et al., 2023b) and preliminary experiments. Furthermore, investigating other graph convolutions or different GNN architectures in combination with specific population graph setups might give more insights. One example would be higher-order GNNs for node-level predictions (Li et al., 2021). We would see this as a fitting method for longitudinal studies. Finally, it could be interesting to evaluate additional graph assessment metrics (Luan et al., 2021; Xie et al., 2020; Luan et al., 2022) and their correlation with graph construction methods and model performance.

We generally see three future directions for population graph studies. Either (a) new and better graph construction methods need to be developed for population graphs to bring benefits to medical downstream tasks, (b) innovative applications of population graphs that truly benefit from the usage of connectivity information need to be explored, or (c) the usage of population graphs in combination with GNNs does not seem valuable for the performance of medical downstream tasks. It would be interesting to follow up with a theoretical analysis. Our experiments showed that the graph construction is a major performance bottleneck for population graphs. We believe this to be a good starting point for follow-up analyses. For better graph construction methods, we see the requirement of increasing the information content of the graph structure compared to the node features alone. This could potentially be achieved by encoding information in the graph structure that cannot be trivially added to the node features, such as genetic similarity between subjects or the risk groups in survival analysis. Other potentially interesting applications of population graphs, where the edges add additional information to the node features, are geospatial graphs. This could include analysing location-based health data, disease spreading, or tracing local differences in medication or care units. Also, time-series data has not been explored in great detail in the context of population graphs, which might add more valuable information to the graph structure. The concept of population graphs has, with a few exceptions (Keicher et al., 2021), mostly focused on vector data instead of images. This is mostly because image data is much larger and, therefore, more difficult to fit into memory in the context of population graphs. This could be another interesting future direction to improve the predictive power of population graphs.

Acknowledgements

TM and SS were supported by the ERC (Deep4MI - 884622). This work has been conducted under the UK Biobank application 87802. SS has furthermore been supported by BMBF and the NextGenerationEU of the European Union.

References

David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors*, 21(14): 4758, 2021.

Wenjia Bai, Hideaki Suzuki, Jian Huang, Catherine Francis, Shuo Wang, Giacomo Tarroni, Florian Guitton, Nay Aung, Kenneth Fung, Steffen E Petersen, et al. A population-based phenome-wide association study of cardiac and aortic structure and function. *Nature medicine*, 26(10):1654–1662, 2020.

Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Graph neural networks in network neuroscience. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Kyriaki-Margarita Bintsi, Vasileios Baltatzis, Rolandos Alexandros Potamias, Alexander Hammers, and Daniel Rueckert. Multimodal brain age estimation using interpretable adaptive population-graph learning. *arXiv preprint arXiv:2307.04639*, 2023a.

Kyriaki-Margarita Bintsi, Tamara T. Mueller, Sophie Starck, Vasileios Baltatzis, Alexander Hammers, and Daniel Rueckert. A comparative study of population-graph construction methods and graph neural networks for brain age regression, 2023b.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Liò, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022.

Stephen Bonner, Ian P Barrett, Cheng Ye, Rowan Swiers, Ola Engkvist, Andreas Bender, Charles Tapley Hoyt, and William L Hamilton. A review of biomedical datasets relating to drug discovery: a knowledge graph perspective. *Briefings in Bioinformatics*, 23(6), 2022.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Li Chen, Thomas Hatsukami, Jenq-Neng Hwang, and Chun Yuan. Automated intracranial artery labeling using a graph neural network and hierarchical refinement. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23*, pp. 76–85. Springer, 2020.

James H Cole. Multimodality neuroimaging brain-age in uk biobank: relationship to biomedical, lifestyle, and cognitive factors. *Neurobiology of aging*, 92:34–42, 2020.

Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 643–653. Springer, 2020.

Padraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers-a tutorial. *ACM computing surveys (CSUR)*, 54(6):1–25, 2021.

Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Mahsa Ghorbani, Anees Kazi, Mahdieh Soleymani Baghshah, Hamid R Rabiee, and Nassir Navab. Ra-gcn: Graph convolutional network for disease prediction problems with imbalanced data. *Medical Image Analysis*, 75:102272, 2022.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pp. 729–734, 2005.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3:315–358, 2019.

Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. Inceptiongcn: receptive field aware graph convolutional network for disease prediction. In *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, pp. 73–85. Springer, 2019.

Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael M Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1606–1617, 2022.

Matthias Keicher, Hendrik Burwinkel, David Bani-Harouni, Magdalini Paschali, Tobias Czempiel, Egon Burian, Marcus R Makowski, Rickmer Braren, Nassir Navab, and Thomas Wendler. U-gat: Multimodal graph attention network for covid-19 outcome prediction. *arXiv preprint arXiv:2108.00860*, 2021.

Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. *Advances in Neural Information Processing Systems*, 34: 4314–4327, 2021.

Dongkwan Kim and Alice Oh. How to find your friendly neighborhood: Graph attention design with self-supervision. *arXiv preprint arXiv:2204.04879*, 2022.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Jianxin Li, Hao Peng, Yuwei Cao, Yingtong Dou, Hekai Zhang, S Yu Philip, and Lifang He. Higher-order attribute-enhancing heterogeneous graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):560–574, 2021.

Derek Lim, Xiuyu Li, Felix Hohne, and Ser-Nam Lim. New benchmarks for learning on non-homophilous graphs. *arXiv preprint arXiv:2104.01404*, 2021.

Siyuan Lu, Ziquan Zhu, Juan Manuel Gorriz, Shui-Hua Wang, and Yu-Dong Zhang. Nagnn: classification of covid-19 based on neighboring aware representation from deep graph neural network. *International Journal of Intelligent Systems*, 37(2):1572–1598, 2022.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, and Doina Precup. When do we need gnn for node classification? *arXiv preprint arXiv:2210.16979*, 2022.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? *arXiv preprint arXiv:2106.06134*, 2021.

José Teófilo Moreira-Filho, Meryck Felipe Brito da Silva, Joyce Villa Verde Bastos Borba, Arlindo Rodrigues Galvão Filho, Eugene Muratov, Carolina Horta Andrade, Rodolpho de Campos Braga, and Bruno Junior Neves. Artificial intelligence systems for the design of magic shotgun drugs. *Artificial Intelligence in the Life Sciences*, pp. 100055, 2022.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Tamara T. Mueller, Sophie Starck, Leonhard F. Feiner, Kyriaki-Margarita Bintsi, Daniel Rueckert, and Georgios Kaissis. Extended graph assessment metrics for graph neural networks, 2023a.

Tamara T Mueller, Siyu Zhou, Sophie Starck, Friederike Jungmann, Alexander Ziller, Orhun Aksoy, Danylo Movchan, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert. Body fat estimation from surface meshes using graph neural networks. In *International Workshop on Shape in Medical Imaging*, pp. 105–117. Springer, 2023b.

Tamara T Mueller, Sophie Starck, Alina Dima, Stephan Wunderlich, Kyriaki-Margarita Bintsi, Kamilia Zaripova, Rickmer Braren, Daniel Rueckert, Anees Kazi, and Georgios Kaissis. A survey on graph construction for geometric deep learning in medicine: Methods and recommendations. *Accepted at Transactions on Machine Learning Research*, 2024. URL `https://openreview.net/forum?id=sWlHhfijcS`.

Johannes C Paetzold, Julian McGinnis, Suprosanna Shit, Ivan Ezhov, Paul Büschl, Chinmay Prabhakar, Mihail I Todorov, Anjany Sekuboyina, Georgios Kaissis, Ali Ertürk, et al. Whole brain vessel graphs: a dataset and benchmark for graph learning and neuroscience (vesselgraph). *arXiv preprint arXiv:2108.13233*, 2021.

Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrerro Moreno, Ben Glocker, and Daniel Rueckert. Spectral graph convolutions for population-based disease prediction. In *International conference on medical image computing and computer-assisted intervention*, pp. 177–185. Springer, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Chantal Pellegrini, Nassir Navab, and Anees Kazi. Unsupervised pre-training of graph transformers on patient population graphs. *arXiv preprint arXiv:2207.10603*, 2022.

Liang Peng, Nan Wang, Nicha Dvornek, Xiaofeng Zhu, and Xiaoxiao Li. Fedni: Federated graph learning with network inpainting for population-based disease prediction. *IEEE Transactions on Medical Imaging*, 2022.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12:e1001779, 2015.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Gerome Vivar, Anees Kazi, Hendrik Burwinkel, Andreas Zwergal, Nassir Navab, Seyed-Ahmad Ahmadi, et al. Simultaneous imputation and classification using multigraph geometric matrix completion (mgmc): Application to neurodegenerative disease classification. *Artificial Intelligence in Medicine*, 117:102097, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

Yiqing Xie, Sha Li, Carl Yang, Raymond Chi-Wing Wong, and Jiawei Han. When do gnns work: Understanding and improving neighborhood aggregation. In *IJCAI'20: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,{IJCAI} 2020*, 2020.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Shuangzhi Yu, Shuqiang Wang, Xiaohua Xiao, Jiuwen Cao, Guanghui Yue, Dongdong Liu, Tianfu Wang, Yanwu Xu, and Baiying Lei. Multi-scale enhanced graph convolutional network for early mild cognitive impairment detection. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, pp. 228–237. Springer, 2020.

Shuai Zheng, Zhenfeng Zhu, Zhizhe Liu, Zhenyu Guo, Yang Liu, Yuchen Yang, and Yao Zhao. Multi-modal graph learning for disease prediction. *IEEE Transactions on Medical Imaging*, 41(9):2207–2216, 2022.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

8. Differentially Private Guarantees for Analytics and Machine Learning on Graphs: A Survey of Results

Tamara T. Mueller, Dmitrii Usynin, Johannes C. Paetzold, Daniel Rueckert, and Georgios Kaissis

Journal: Journal of Privacy and Confidentiality, Volume 14, No. 1, February 2024.

Synopsis: We study the applications of differential privacy (DP) in the context of graph- structured data and discuss the formulations of DP applicable to the publication of graphs and their associated statistics as well as machine learning on graph-based data, including graph neural networks (GNNs). Interpreting DP guarantees in the context of graph- structured data can be challenging, as individual data points are interconnected (often non-linearly or sparsely). This connectivity complicates the computation of individual privacy loss in differentially private learning. The problem is exacerbated by an absence of a single, well-established formulation of DP in graph settings. This issue extends to the domain of GNNs, rendering private machine learning on graph-structured data a challenging task. A lack of prior systematisation work motivated us to study graph-based learning from a privacy perspective. In this work, we systematise different formulations of DP on graphs, discuss challenges and promising applications, including the GNN domain. We compare and separate works into graph analytics tasks and graph learning tasks with GNNs. We conclude our work with a discussion of open questions and potential directions for further research in this area.

Contributions of thesis author: literature research, design of categorisation scheme, manuscript writing.

DIFFERENTIAL PRIVACY GUARANTEES FOR ANALYTICS AND MACHINE LEARNING ON GRAPHS: A SURVEY OF RESULTS

TAMARA T. MUELLER, DMITRII USYNIN, JOHANNES C. PAETZOLD, RICKMER BRAREN, DANIEL RUECKERT, AND GEORGIOS KAISSIS

Institute for Artificial Intelligence in Medicine and Healthcare and Institute of Radiology, Technical University of Munich
e-mail address: tamara.mueller@tum.de

Institute for Artificial Intelligence in Medicine and Healthcare and Institute of Radiology, Technical University of Munich

Department of Informatics, Technical University of Munich; Institute for Tissue Engineering and Regenerative Medicine, Helmholtz Zentrum München

Institute of Radiology, Technical University of Munich

Institute for Artificial Intelligence in Medicine and Healthcare and Institute of Radiology, Technical University of Munich; Department of Computing, Imperial College London

Institute for Artificial Intelligence in Medicine and Healthcare and Institute of Radiology, Technical University of Munich; Department of Computing, Imperial College London; OpenMined

Abstract. We study the applications of differential privacy (DP) in the context of graph-structured data and discuss the formulations of DP applicable to the publication of graphs and their associated statistics as well as machine learning on graph-based data, including graph neural networks (GNNs). Interpreting DP guarantees in the context of graph-structured data can be challenging, as individual data points are interconnected (often non-linearly or sparsely). This connectivity complicates the computation of individual privacy loss in differentially private learning. The problem is exacerbated by an absence of a single, well-established formulation of DP in graph settings. This issue extends to the domain of GNNs, rendering private machine learning on graph-structured data a challenging task. A lack of prior systematisation work motivated us to study graph-based learning from a privacy perspective. In this work, we systematise different formulations of DP on graphs, discuss challenges and promising applications, including the GNN domain. We compare and separate works into graph analytics tasks and graph learning tasks with GNNs. We conclude our work with a discussion of open questions and potential directions for further research in this area.

Key words and phrases: differential privacy, graph-structured data, graph analytics, graph neural networks.

JOURNAL OF PRIVACY
AND CONFIDENTIALITY

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.v0.i0.000

1. Introduction

Many real-world datasets like social networks, molecules, population data or electronic health records don't naturally befit a row-and-column (tabular) representation as they hold complex internal connections and relationships. Such data can often be efficiently represented using graphs as data structures. The additional intrinsic structural information maintained by this representation holds great potential for data analytics and learning tasks on such graph-structured data. A graph's interconnected nature can be leveraged by appropriate algorithms and graph-based learning models and can be deployed in contexts such as market value prediction [1], fake news detection [2] and drug development [3]. Within the last two decades, "traditional" algorithms such as triangle counting, node degree estimation etc. have been complemented or superseded by advanced machine learning applications on graph-structured data, made possible by the introduction of Graph Neural Networks (GNNs) [4]. Such models have since then been successfully applied to various learning scenarios [5, 6, 7]. These works demonstrate that a graph's connectivity confers valuable additional information, and allows analysts to leverage the interaction between individual data points, which can significantly improve the accuracy of learning tasks compared to reducing graph-structured data to a tabular form [8]. However, the information contained in graph-structured data is often highly sensitive in nature in the sense that either the data in the graph's *nodes*, the *connections between nodes* or *both* represent sensitive information mandating protection.

Moreover, the rich inter-node relationships render graph-structured data more vulnerable to attacks that attempt do disclose the private data of individuals contained within the graph without their consent [9, 10]. Such attacks can take form of membership inference (MIA) [11], where the adversary attempts to verify if a record that they possess was part of the sensitive dataset (e.g. a patient's electronic health record). MIA, in fact, has a higher fidelity in graph-based settings, due to additional information that intrinsically lies in the structure of a graph [12]. Another commonly used attack is termed an attribute (or feature) inference attack [13]. It aims to reconstruct sensitive features of individuals in the training dataset and typically involves an adversary having access to a non-overlapping dataset of publicly available attributes which, alongside the predictions of the trained model, are used to determine the value of a sensitive feature that belongs to a target participant. Furthermore, models trained on graph-structured data, such as GNNs, were shown to be susceptible to model inversion attacks (MInv) [14], which allow the adversary to extract sensitive training data by leveraging the internal representations of the model (e.g. reverse-engineering a model update into disclosing which data point corresponds to this specific update). Authors in [9] show that MInv attacks can be adapted to graph-based learning. Notably, seeing as graph-structured data captures information not just about individuals themselves, but about their relationships with other participants, all of these attacks can potentially compromise privacy of *multiple participants at once.*

The increasing popularity of graph-based analytics and machine learning coupled with the regulatory and ethical mandates to protect sensitive data imply that privacy enhancing technologies (PETs) [15] need to be applied in order to provide formal guarantees of privacy. Differential privacy (DP) [16] was proposed to objectively quantify the privacy loss of individuals whose data is subjected to algorithmic processing and is now regarded as the *gold standard* of formal privacy guarantees. Differentially private algorithms upper-bound the amount of information that can be inferred by an adversary from observing a computation's output, thus mitigating the attacks discussed above. The utilisation of DP mechanisms thus allows to train machine learning models on sensitive datasets while preserving privacy of contributors' data. However, the adaptation of DP to graph-structured data is non-trivial for two main reasons: (1) there exist several notions of DP on graph-structured data, which protect different components of the graph, and thus need to be selected carefully and appropriately to the application; (2) due to the formal definition of DP, its realisation on graphs encompasses several additional implementation challenges compared to tabular data.

To promote the development of responsible and privacy-preserving machine learning systems, we identify the requirement for a comprehensive systematisation of knowledge in the areas of differentially private graph analytics and graph machine learning tasks. In this work, we investigate existing implementations, their limitations and application areas, as well as a number of challenges associated with differentially private learning on graph-based structures and promising directions for future work. We distinguish two lines of works: (1) non-machine learning graph analytics methods and (2) machine learning approaches on graph-structured data with Graph Neural Networks (GNNs). This distinction allows us to emphasise open challenges and highlight opportunities to transferring DP techniques from graph analytics methods to GNNs. The outline of the remaining work and our main contributions can be summarised as follows:

- In Section 2, we provide an introduction to graph-structured data and graph neural networks, as well as a formal definition of DP;
- We formalise the three main notions of DP on graph-structured data: *edge-level*, *node-level*, and *graph-level* DP in Sections 2 and expand them by introducing several additional notions of DP in Section 4;
- We demonstrate how different DP formulations can be applied in various settings in Section 5 and how graph analytics and graph learning under DP can be compared in all scenarios;
- We identify limitations and open challenges of these approaches and pinpoint promising areas of future work in the domain of DP on graph-structured data in Section 6.

2. Background

In this section, we formalise the concept of DP, introduce the three main notions on DP on graph-structured data, as well as the concept of sensitivity, the Gaussian and the Laplace mechanisms, and provide a brief introduction to graph-structured data and graph neural networks (GNNs).

2.1. **Graph-Structured Data.**

In the following, we will refer to a graph $G = (V, E)$ as a collection containing a set of nodes $V = \{v_1, v_2, ..., v_n\}$ and a set of edges $E = \{e_1, e_2, ..., e_m\}$, n and $m \in \mathbb{N}$. Here, n determines the number of nodes in the graph and m the number of edges. The data contained in the graph can be split into the attributes contained in the nodes V of the graph, which can be referred to as node features, and the data held by the connections E between the nodes. The edges can optionally also contain edge attributes, holding additional information about the tightness or nature of the connection.

2.2. **Differential Privacy.** Differential privacy (DP) is a stability condition on randomised algorithms that makes their outputs approximately invariant to an inclusion or exclusion of a single individual [16]. In the words of the authors of [16], DP promises "to protect individuals from any additional harm that they might face due to their data being in the private database that they would not have faced had their data not been part of [the database]". This allows one to interpret DP as guaranteeing an upper bound on the *effect size* introduced by the inclusion or exclusion of the individual's data [17]. The DP framework and its associated techniques allow data analysts to draw conclusions about datasets while preserving the privacy of individuals. We note that the DP guarantee makes *no assumption* about potential correlations between datapoints, however the "standard" interpretation of DP can behave in unpredictable ways when applied naïvely to data with such correlations such as graphs, prompting the more specific definitions introduced below.

In a setting of DP on graph-structured data, we assume that an analyst $\mathcal{A}$ is entrusted with a database D containing sensitive graph-structured data. From D a neighbouring (in this work we additionally use the term *adjacent*) dataset D' is constructed by either (a) removing or adding one node and its adjacent edges (*node-level* DP), (b) removing or adding one edge (*edge-level* DP), or (c) removing or adding one graph (*graph-level* DP). Formally, DP can be defined as follows:

Definition 2.1 ((ε-δ)-DP)**.** *A randomised algorithm $\mathcal{M}$ is (ε-δ)-differentially private if for all $S \subseteq$* Range($\mathcal{M}$) and all neighbouring datasets D and D' in X the following (symmetric) statement holds:

$$\mathbb{P}[\mathcal{M}(D) \in S] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(D') \in S] + \delta. \tag{2.1}$$

The definition of neighbouring datasets on graph-structured data depends on the desired formulation of privacy in the setting (i.e. which attributes need to be kept private, such as outgoing edges for instance). Therefore, the desired notion (as well as the associated mechanisms) of privacy preservation depend on what the data owner requires to protect, the structure of the graph and the desired application to ensure a context-appropriate interpretation of the DP guarantee. In order to employ differentially private algorithms to process graph-structured data, the property of neighbouring datasets thus needs to be formally defined. The three main notions of DP on graphs can be formalised as follows:

Definition 2.2 (*Edge-level* DP)**.** Under *edge-level* differential privacy, two graphs G and G' are neighbouring if they differ in a single edge (either through addition or through removal of the edge) [18]. (ε-δ) edge differential privacy is therefore preserved if equation (2.1) holds for all events S and all pairs of neighbours G, G' that differ in a single edge. In this setting, two graphs $G = \{V, E\}$ and $G' = \{V', E'\}$ are neighbours if

$$V' = V \wedge E' = E \setminus e_i, \tag{2.2}$$

where $e_i \in E$.

Definition 2.3 (*Node-level* DP)**.** Under *node-level* DP, two graphs $G = \{V, E\}$ and $G' = \{V', E'\}$ are defined as neighbouring if they differ in a single node and its corresponding edges (achieved through a node removal/addition) [19]. (ε-δ)-node differential privacy is therefore preserved if equation (2.1) holds for all events S and all pairs of neighbours G, G', that differ in a single node and its corresponding edges:

$$V' = V \setminus v_i \wedge E' = E \setminus c, \tag{2.3}$$

where v_i is a node in V and c is the set of all edges connected to v_i.

Figure 1 visualises these two main definitions of DP on graphs. Two neighbouring datasets (graphs) under *node-level* DP and *edge-level* DP are displayed in sub-figures **A** and **B**, respectively.

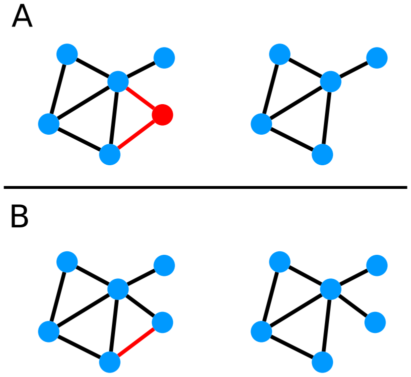


Figure 1: Two neighbouring graphs in the context of (**A**) *node-level* DP and (**B**) *edge-level* DP. By removing (**A**) one node and its adjacent edges or (**B**) one edge (displayed in red), two neighbouring graphs can be transformed into each other.

For multi-graph datasets, we can define a different notion of privacy:

Definition 2.4 (*Graph-level* DP)**.** Under *graph-level* DP, we define two multi-graph datasets $D = \{G_{11}, G_{12}, \ldots, G_{1n}\}$ and $D' = \{G_{21}, G_{22}, \ldots, G_{2m}\}$ to be neighbours if they differ in one single graph (achieved through the addition or removal of one entire graph). (ε-δ)-graph differential privacy is therefore preserved if equation (2.1) holds for all events S and all pairs of neighbouring datasets D and D', where

$$D' = D \setminus G_{1i}, \tag{2.4}$$

and $G_{1i} \in D$.

We now assume that the analyst $\mathcal{A}$ executes a function (or *query*) f over the graph dataset. When considering DP in GNNs, the function f is a repeated composition of the forward pass, loss calculation, and gradient computation of the graph neural network (resulting in a "database" of gradients). In order to determine the magnitude of noise that needs to be added, we are required to calculate the sensitivity of the function that noise is applied to. We will consider either the L_1- or the L_2-sensitivity of f.

Definition 2.5 (L_2-sensitivity Δ_2 of f)**.** Let f be defined as above and X be the set of all neighbouring databases. We can define the L_2-sensitivity of f as:

$$\Delta_2(f) := \max_{D,D' \in X, D \simeq D'} \|f(D) - f(D'))\|_2. \tag{2.5}$$

We note that the maximum is taken over all neighbouring pairs of datasets in X.

Using the definition of L_2-sensitivity, we can formalise the Gaussian Mechanism on f:

Definition 2.6 (Gaussian Mechanism)**.** Let Δ_2 and f be defined as above. The Gaussian Mechanism $\mathcal{M}$ is applied to the function $\mathbf{y} = f(x)$, $y \in \mathbb{R}^n$, as follows:

$$\mathcal{M}(\mathbf{y}) = \mathbf{y} + \xi, \tag{2.6}$$

where $\xi \sim \mathcal{N}(0, \sigma\mathbb{I}^n)$. $\mathbb{I}^n$ is the identity matrix with n diagonal elements and σ is calibrated to Δ_2.

Similarly to L_2-sensitivity, we can define the L_1-sensitivity as:

Definition 2.7 (L_1-sensitivity Δ_1 of f)**.**

$$\Delta_1(f) := \max_{D,D' \in X, D \simeq D'} ||f(D) - f(D')||_1. \tag{2.7}$$

When it is clear from context, we will omit the argument and write just $\Delta_{1/2}$.

Definition 2.8 (Laplace Mechanism)**.** Let Δ_1 and f be defined as above. The Laplace Mechanism $\mathcal{M}$ is applied to the output $\mathbf{y} = f(x)$, $\mathbf{y} \in \mathbb{R}^n$, as follows:

$$\mathcal{M}(\mathbf{y}) = \mathbf{y} + (\xi_1, \xi_2, \dots, \xi_n), \tag{2.8}$$

where ξ_i are I.I.D. draws from Lap $\left(0, \frac{\Delta_1}{\varepsilon}\right)$.

2.2.1. *Local and Central DP.* In general, one can furthermore distinguish between *local* and *central* DP. Under local differential privacy (LDP) [20] the data owner performs the noise perturbation step before the data reaches the analyst. Such interpretation can be preferable in low-trust collaborative learning settings, as no party other than its owner has access to the data before the learning task commences. Data owners only share a perturbed version of their training data, which reduces the amount of information an analyst can infer about the shared data itself, while still allowing to draw insights from the privatised aggregated data [7]. Local DP thus bounds the information at the data source itself, minimising the potential privacy exposure [15]. An adversary is, therefore, unable to infer the input value with high confidence, but is possible to approximate the target query if provided with a large number of noisy samples [7]. More details about local DP on graph-structured data can be found in Section 4.6.

When DP is, on the other hand, applied to the output of the computation instead of the input data, one speaks of central differential privacy. In this case, the noise is not added directly to the input data but instead to the computation outputs. Due to the properties of DP, only a bounded quantity of additional information can be derived about the data belonging to an individual, while the overall statistics of the whole dataset can still be approximately evaluated.

2.3. **Graph Neural Networks.** To allow machine learning to be performed directly on graph-structured data, GNNs were proposed [4]. They leverage the full underlying structure of the dataset and maximise learning capacity by directly learning *on the graph.* GNNs can be applied to either single graph or multi-graph datasets, depending on task and application. The three major application areas of GNNs are *node classification* (where one label is predicted for each node in the graph), *edge prediction* (where edges are predicted or labeled), and *graph classification* (where one label is predicted for each graph).

A key concept for the successful application of GNNs is message passing [21], where information is shared along edges and therefore propagated among neighbourhoods of nodes. This property enables the utilisation of the full dimensionality of graph datasets. However, this typically complicates the disentanglement of contributions by individual nodes, making the calculation of individual privacy loss per each participant a challenging task.

3. Systematisation Methodology

We conducted a survey of papers that intersect the domains of graph analytics or deep learning on graphs with differential privacy. We employed the *Google Scholar* and the *Web of Science* search engines and examined papers that contained the keywords "node-", "edge-", "graph-" "differential privacy" between January, 2007 and February 2022. Our searches often had to be coupled (e.g. "node differential privacy graphs"), as notions such as *graphs* or *nodes* are often used in unrelated concepts such as computation graphs or network nodes. We selected 51 studies, which we partitioned based on the application of DP employed in each work: *node-level* DP, *edge-level* DP, *graph-level* DP, and whether *local DP* was applied in the respective works. Furthermore, we separated the works into *graph analytics* and *GNN training* applications. We additionally recorded the contexts in which DP was applied. A summary of the works that we discuss in this study can be found in Table 1.

We observed that a large number of studies concentrate on the usage of graph datasets but explicitly not on the application of GNNs. The large amount of research in the context of DP on graphs in general shows the importance of applying differentially private algorithms to graph-structured data. However, applications of DP to GNNs are currently underrepresented, presumably due to the fact that GNNs are a relatively recent deep learning method, and the application of DP to GNNs entails several challenges. For example, there is no singular explicit notion of "DP" in different graph machine learning settings, as discussed below. Furthermore, the here-presented systematisation of different possibilities to apply differential privacy to graph neural networks will act as a comprehensive guide to practitioners and aid in the the development of new methods in this area. With the advent of privacy-preserving machine learning and the strong interest in geometric deep learning applications, we strongly believe the differentially private training of GNNs to be a promising future research area with several applications to sensitive data. We therefore explicitly decided to include both graph analytics and machine learning on graphs in our survey. Some exemplary application areas are discussed in Section 5.

4. DP formulations on Graph-Structured Data

In this section, we outline and discuss methods from the research field of differentially private graph analytics and graph machine learning. We identify and consider two separate lines of work: **(a)** DP in traditional graph analytics methods and **(b)** DP in graph neural networks. We therefore separate the works in Table 1 depending on their association with one of those categories. We also indicate the notion of DP that was applied in the respective research in the columns *Edge-DP*, *Node-DP*, *Graph-DP*, and *LDP* and summarise ranges of the privacy budget ε if they were reported in the respective works. The line of work of DP in traditional graph analytics **(a)** includes methods for privately computing graph statistics like degree-distributions [23], frequent sub-graph-mining [34], and sub-graph counting [31], as well as private graph release [24, 43, 6]. The works of DP for GNN training **(b)** include, for instance, text classification [5], whole-graph classification [68], and attacks on GNNs [9].

The first application of differentially private computation on graph data was introduced by Nissim et al. [22]. Authors showed an estimation of the cost associated with the computation of a minimum spanning tree and triangle counts in a differentially private manner. In their work, the authors opted for the utilisation of *edge-level* DP.

As indicated in Table 1, we generally observe a focus on *edge-level* DP in earlier papers, compared to a more frequent utilisation of *node-level* DP in more recent works. We attribute

	Edge-DP	**Node-DP**	**Graph-DP**	**LDP**	**Year**	**Reference**	**Context**	ε
Graph Analytics	✓	✓			2007	Nissim et al. [22]	Estimation for spanning trees	-
	✓				2009	Hay et al. [23]	Graph degree estimation	$[0.01; 1]$
	✓				2009	Mir et al. [24]	Graph estimation	-
					2011	Gehrke et al.** [25]	Zero-knowledge statistics estimation	-
	✓				2011	Machanavajjhala et al. [26]	Privacy in social graphs	$[0.5; 3]$
	✓				2011	Sala et al. [27]	Release of private graphs	$[0.1; 100]$
	✓				2011	Karwa et al. [18]	Private subgraph counting	0.5
	✓				2012	Gupta et al. [28]	Synthetic data for graph cuts	-
	✓				2012	Karwa et al. [29]	Release of graph degree sequences	-
	✓				2012	Mir et al. [30]	Private release of graph distribution	0.2
	✓	✓			2013	Blocki et al. [31]	Restricted sensitivity for DP	-
		✓			2013	Chen et al. [32]	Private graph database aggregation	$[0.1; 0.5]$
		✓			2013	Kasiviswanathan et al. [33]	Private graph analysis	-
			✓		2013	Shen et al. [34]	Private graph pattern mining	$[0.1; 1]$
	✓				2013	Wang et al. [35]	Private spectral graph analysis	460
	✓				2013	Wang et al. [36]	Private spectral graph analysis	-
	✓				2014	Chen et al. [37]	Correlated network data release	$[0.6; 1]$
	✓				2014	Lu et al. [38]	Estimation of graph model parameters	$[0.1, 1]$
		✓			2014	Proserpio et al. [39]	Synthetic graph generation	$[0.01; 10]$
		✓			2014	Raskhodnikova et al. [40]	DP analysis of graphs	-
	✓	✓	✓		2014	Task et al. [41]	Private social network analysis	-
		✓			2016	Day et al. [42]	Private graph distribution release	$[0.1; 2]$
	✓				2016	Jorgensen et al. [43]	Private attributed graph models	$[1; 20]$
		✓			2016	Raskhodnikova et al. [44]	Private release of graph statistics	-
	✓			✓	2016	Wang et al. [45]	Private aggregation of data	$[0; 2]$
	✓	✓		✓	2017	Qin et al. [46]	Private release of social graphs	$[0; 7]$
	✓	✓			2017	Zhu et al. [47]	Applications of differential privacy	-
	✓	✓		✓	2018	Cormode et al. [48]	Private data release	-
		✓			2018	Macwan et al. [49]	Private release of graph data	0.5
	✓				2019	Arora et al. [50]	Graph sparsification	-
		✓			2019	Sealfon et al. [51]	Estimation of graph statistics	-
					2019	Sun et al. [52]	Subgraph statistics, decentralised DP	$[1; 10]$
		✓			2019	Yuxuan et al. [53]	Private histogram release	-
	✓				2020	Chen et al. [54]	Private synthetic data release	$[2; 5]$
		✓			2020	Liu et al. [55]	Node strength distribution	$[0.1; 2]$
		✓			2020	Zhang et al. [56]	Private social graph release	$[0.1; 20]$
		✓		✓	2020	Zhang et al. [57]	Control-flow graph coverage analysis	$[2^{-5}; 2^5]$
		✓			2021	Iftikhar et al. [58]	Private release of degree distribution	$[0.01; 10]$
		✓			2021	Fichtenberger et al. [59]	Private dynamic graph algorithms	-
	✓			✓	2021	Imola et al. [60]	Private sub-graph counting	$[0; 2]$
		✓			2021	Lan et al. [61]	Private node strength histogram release	$[0.1; 2]$
		✓			2021	Liu et al. [62]	Private degree histogram release	$[0.1; 2]$
		✓			2021	Sealfon et al. [63]	Private graph density estimation	-
	✓	✓		✓	2021	Xia et al. [64]	Benchmark platform for DP on graphs	-
	✓			✓	2021	Zheng et al. [65]	Private graph publication framework	-
	✓				2021	Zheng et al. [66]	Network Generation	$[0.1; 440]$
	Edge-DP	**Node-DP**	**Graph-DP**	**LDP**	**Year**	**Reference**	**Context**	ε
GNNs				✓	2020	Sajadmanesh et al.* [7]	Locally private GNNs	$[0.01; 3]$
		✓			2021	Daigavane et al. [67]	Node-level DP in GNNs	$[5; 30]$
					2021	Igamberdiev et al.* [5]	Private text classification	$[1; 100]$
					2021	Olatunji et al.* [6]	Private GNN and graph data release	$[1; 40]$
					2021	Zhang et al.* [9]	Attacks on GNNs	$[1; 10]$
			✓		2022	Mueller et al. [68]	Graph-level DP for graph classification	$[0.5; 20]$

Table 1: Summary of existing works on DP on graphs, ordered ascending by publication year and alphabetically within the same year. The works are split into *Graph Analytics* and *GNNs*. Ticks in columns **Edge-DP**, **Node-DP**, and **Graph-DP** specify which notion of privacy was used. A tick in column *LDP* indicates that the authors used local DP. The asterisks (*) indicates that the DP notion is not clearly stated. Two asterisks (**) indicate the utilisation of zero-knowledge privacy (see Section 4.5.3). The column ε reports the privacy budget that was evaluated in the respective works.

this to the fact that *node-level* DP is more challenging to achieve, but offers stronger privacy guarantees (as it considers the privacy of a node and all its adjacent edges). Works on *graph-level* DP are quite rare. However, we believe this notion of DP to be promising and given that different works name the same concept differently, we still included graph-level DP in Table 1.

We furthermore observe that, in the works discussing DP on GNNs, authors frequently omit to specifically assign the guarantees provided to one of the aforementioned DP notions, which highlights the need for more systematic approaches to defining DP in graph learning tasks. We attribute this lack of specification to missing systematisation of terminology in this area as well as the challenging task to differentiate the individual notions of privacy in graph learning tasks and their dependence on the dataset and the application area.

4.1. **Sensitivity Calculation on Graphs.** As described above, ensuring data privacy on graphs presents additional challenges compared to structured databases such as image or tabular datasets, since the data points are inter-connected and the graph structure itself can contain sensitive information. Furthermore, depending on the application it can be desired to protect different parts of the graph. One fundamental challenge is therefore the issue of sensitivity calculation.

In cases of graphs, this value can be challenging to obtain as it depends not only on the structure of the graph but also on the attributes of the query function. Two main methods have been proposed to obtain node differentially private algorithms which are either based on (a) the utilisation of projections, for which sensitivity can be bounded, or (b) on computing Lipschitz extensions [44, 31, 32]. Raskhodnikova et al. [44] study the efficient computation of Lipschitz extensions for multi-dimensional functions on graphs, which can be obtained in polynomial time, and determine that they do not always exist - in comparison to Lipschitz extensions for 1-dimensional functions.

In the next sections, we give more details about the different definitions of DP on graphs in *node-level*, *edge-level*, *graph-level* DP as well as some alterations and combinations of these, with respective interpretations of what is implied by neighbouring datasets in each setting.

4.2. **Edge-Level Differential Privacy.** There exist several approaches that allow to release graph statistics with *edge-level* DP guarantees, including sub-graph counts [29], spanning tree estimation [22], degree distributions [41, 23] and graph cuts [28]. Those settings set a focus on privatising the relationships between nodes. This can be applied to social network graphs [23, 24] or location graphs [69], where the edges contain sensitive information, but the data represented in the nodes of the graph are assumed to be publicly known or non-sensitive.

4.3. **Node-Level Differential Privacy.** *Node-level* differential privacy is a strictly stronger guarantee than edge-level differential privacy [33]. This is of particular importance in scenarios where graphs are very sparse, and thus, the removal of a single node can alter the graph structure severely. For instance, the number of triangles in a graph with n nodes can increase by $\binom{n}{2}$ when inserting a single additional node. Consequently, these functions tend to have high sensitivity [44], resulting in an unnecessarily large noise magnitude. Bounded-degree graphs (graphs where each node has an upper limit of edges and the degree of each node is therefore bounded) can assist in lowering the sensitivity. Here, the removal of a single node results in an upper-bounded change in edges which typically leads to a reduced impact on the output of the algorithm. When calculating the number of triangles in a graph, for instance, maximum change of a D-bounded-degree graph is $\binom{D}{2}$ which is strictly smaller than $\binom{n}{2}$ if $D < n$.

Settings that can benefit most from this formulations of DP are those that put an emphasis on the data within the node itself yet additionally privatise the connections between the nodes include studies on social networks [46, 31], degree histogram distribution [62, 58, 49], and recommendation systems [26].

4.4. **Graph-Level Differential Privacy.** So far, *graph-level* DP has not been explored in great detail, neither in the context of graph analytics nor in GNNs. Task et al. [41] name this notion of privacy *partition privacy* and show its application to graph analytics of social networks. Shen et al. [34] investigate the mining of frequent graph patterns in multi-graph datasets and apply the mechanism of *graph-level* DP to their algorithm. They use Markov Chain Monte Carlo (MCMC) random walks to discover frequently appearing sub-graphs in the graph dataset and infer graph statistics under graph-level DP.

In the context of GNN training, *graph-level* DP can be applied in learning settings that investigate graph classification tasks, e.g. drug discovery or molecule classification [70], discovering disease-specific biomarkers of brain connectivity [71, 72], or shape analysis [73]. This way, privacy guarantees can be given to the individuals, whose sensitive information is contained in those multi-graph datasets. For instance, in the setting of drug discovery, a group of pharmaceutical companies can collaborate on a graph classification task, while bounding the information that can be inferred about their individual molecules, which represent the private data in this context. Mueller et al. [68] apply graph-level DP for classification tasks on several sensitive datasets, implementing the concept of graph-level DP on GNNs and showing potential applications.

4.5. **Further Definitions of DP on Graphs.** We consider node-, edge-, and graph-level DP to be the three main categories of DP guarantees on graph-structured data. However, there exist additional notions of DP on that have not yet found a widespread application and are mostly derived from the notions formalised above. Here, we provide further details about those additional definitions and variations of applied notions of DP.

4.5.1. *k-Edge Differential Privacy.* One such formulation is *k-edge differential privacy* introduced by Hay et al. [23]. It defines a stricter notion of *edge-level* DP, where two graphs $G = \{V, E\}$ and $G' = \{V', E'\}$ are neighbours if $|V \oplus V'| + |E \oplus E'| \leq k$. Hereby, $\oplus$ denotes the symmetric difference. If $k = 1$, the definition recovers *edge-level* DP. However, if $k = |V|$ *k-edge-level* DP is a stricter definition than *node-level* DP, as the set of neighbouring graphs in the definition of *node-level* DP is a subset of the neighbouring graphs under k-edge-level DP. For nodes with a degree smaller then k, k-edge-level DP provides an equivalent protection as *node-level* DP. Nodes with a degree $\geq k$ face more exposure, since they have more edges. However, one can argue that those high degree nodes have a higher impact on the general graph structure and it might therefore be necessary to expose them to larger privacy risks to allow analysts to accurately measure graph statistics. The authors experimentally evaluate their notion of k-edge-differential privacy on social network data from Flickr, LiveJournal, Orkut, and YouTube.

4.5.2. *Out-Link Differential Privacy.* Another definition of DP on graphs was introduced by Task et al. [41] and is termed *out-link differential privacy.* In this context directed graphs are considered, where it is possible to distinguish between incoming and outgoing edges of nodes. Under this notion, two datasets are considered to be neighbouring if all *out-links* (outgoing edges) of an arbitrary node are added or removed. Formally, two graphs $G = \{V, E\}$ and $G' = \{V', E'\}$ are neighbours, if $V = V'$ and $E' = E - \{(v_1, v_2)|v_1 = x\}$ for an $x \in V$. (v_1, v_2) hereby defines an edge going from node v_1 to node v_2.

Out-link DP is strictly weaker then *node-level* DP, but in many applications comparable to *edge-level* DP. Under this notion of DP, an attacker would not be able to determine whether a person x contributed their data to the construction of the graph and participants in the graph can hide their out-links. In the setting of a social network, for instance, a person x can deny friendships. Others can still claim to be friends with person x, but the latter can deny that those connections are mutual (i.e. that person x has out-going links to adjacent nodes). The authors argue that *out-link* privacy simplifies sensitivity computation and reduces noise addition requirements, enabling queries that would be infeasible under previous DP definitions.

Similar to k-edge-level DP, *out-link* DP can also be extended to k-out-link privacy. In this case, neighbouring datasets are considered, that differ in k out-links compared to the original dataset. When considering 2-out-link privacy, for example, two nodes can simultaneously deny all their out-links. This would also enable to protect a complete mutual edge, resulting in *edge-level* DP in addition to *out-link* DP.

4.5.3. *Zero-Knowledge Privacy.* Gehrke et al. [25] introduce a stricter formulation of *node-level* DP, namely *zero-knowledge privacy* on graphs, which authors argue is particularly desirable in social network analysis. It relies on a notion similar to the one of cryptographic zero-knowledge proofs [74], which entails that a protocol participant obtains a computation result with "zero additional knowledge" about the data used to perform this computation. A privacy mechanism $\mathcal{M}$ is $(\mathbf{Agg}, \varepsilon)$-zero-knowledge private if there exists a simulator $\mathcal{S}$ and an agg from the family of algorithms $\mathbf{Agg}$ such that for all neighbouring datasets D_1 and D_2 the following holds: $\mathcal{M}(D_1) \approx_\varepsilon \mathcal{S}(agg(D_2))$ [25]. Authors in [25] apply this definition to ensure that a mechanism does not release additional information apart from "aggregate information" which is considered acceptable to release to ensure usability.

4.5.4. *Relationship Differential Privacy.* Imola et al. [60] introduce a notion called *relationship DP*, a definition falling under local DP. Here, one edge in a graph is masked during the entire learning process. In a setting of social network analysis, relationship DP assumes that each user only knows their own connections (i.e. friends), requiring users to have a higher degree of "trust" when interacting with their immediate neighbours. Given two users v_i and v_j that share a link in the social network, under relationship-DP a user v_i has to trust its adjacent user v_j not to leak information about their shared connection. Intuitively, *edge-level LDP* considers the edge from user v_i to user v_j and the edge from user v_j to user v_i to be two separate "secrets", whereas relationship DP assumes that the two edges represent the same "secret". (More details about edge-level LDP can be found in Section 4.6.) Therefore, the trust model of relationship DP is a stronger one than the one of *edge-level LDP*, which does not hold any assumptions about what other users do, but weaker than the one of centralised edge-level DP, where all edges are held by a centralised party. If a randomised algorithm $\mathcal{M}$ provides ε-edge-level LDP, then $\mathcal{M}$ provides 2ε-relationship DP, given that an edge (v_i, v_j) affects two elements in the adjacency matrix of the graph and the property of group privacy [16].

The authors apply this formulation of privacy to algorithms for sub-graph, k-star, and triangle counting, which can be used to analyse connection patterns in graphs.

4.5.5. *Edge-Weight Privacy.* For shortest path or distance queries on graphs, edge-level and node-level DP are not well suited, since both queries usually return a set of edges, which violates both edge-level and node-level DP. Therefore, Sealfon [75] introduced a different notion of privacy on graphs: *edge-weight privacy.* This notion of privacy is applicable if the edge weights of a graph contain private data, whereas the graph structure itself is publicly available and does not need to be protected. An example would be traffic data in a known street system.

4.5.6. *Node Attribute Privacy.* Chen et al. [54] define another notion of privacy for attributed graphs. An attributed graph $G = (V, E, X)$ is the set of vertices V, edges E and node attributes X. In this definition of privacy, two graphs are defined to be neighbouring if they differ in one edge or in the attribute vector of one node. So in this scenario, the presence of nodes is assumed to be non-private, whereas the connections (edges) between the nodes as well as the attributes that define the nodes contain private information. This definition can for example be useful in social networks, where the existence of a profile can be publicly known but friendships and personal attributes (stored in the profiles/nodes) are private.

4.6. **Local DP on Graphs.** There exist several works that target the preservation of local differential privacy (LDP) on graph-structured data. The advantage of local DP [76] in comparison to central DP is that no trusted third party is required. LDP can and has been applied to both, classical graph analytics and graph neural networks. Qin et al. [46] define *edge-level* and *node-level* LDP in the context of neighbour lists. A neighbour list of a vertex v_i in a directed graph with n vertices is defined to be an n-dimensional bit vector $(b_1, \ldots, b_n)$, where $b_i = 1$, $i \in [1; n]$, if and only if there exists an edge (v_i, v_j), going from v_i to v_j, in the graph, otherwise $b_i = 0$. *Edge-level LDP* is then defined for two neighbour lists that differ in exactly one bit, whereas *node-level LDP* is defined for any two neighbour lists.

4.6.1. *Locally private graph analytics.* Examples for LDP in graph analytics tasks include Zhang et al. [57], who perform control-flow graph coverage analysis under *node-level* LDP and Imola et al. [60], who apply LDP to sub-graph counting, k-star and triangle counts while preserving *edge-level* LDP.

4.6.2. *Locally private GNNs.* LDP can also be applied to GNNs, where settings such as decentralised social networks can benefit from this property, as shown by Sajadmanesh and Gatica-Perez [7]. They introduce a privacy-preserving architecture-agnostic GNN algorithm, which preserves private node features under LDP. Their architecture includes an LDP encoder and an unbiased rectifier, which functions as the communicator between the server and the graph. This algorithm can be applied in a setting where either the node features or the labels (or in certain cases both) are to be kept private regardless of the GNN architecture. Authors use a so-called *multi-bit mechanism* which allows the nodes to perturb their features before passing them to the server. The server then processes this noisy data through the first convolutional layer. GNNs aggregate the node features before passing them through the activation function, which can be used as a denoising mechanism to average out the noise that was injected into the node features in the first place. The authors employ a generalised randomised response mechanism [77] to preserve privacy of node labels. However, they explicitly do not preserve *node-level* or *edge-level* DP but protect the privacy of node features and labels. This leaves the graph structure itself unprotected, which remains an open challenge in this context.

4.7. **DP for Graph Neural Networks.** While the notion of DP on traditional graph analytics and statistics applications (particularly for private data release) is well established, there exist significantly fewer studies on differentially private GNN training. This can be attributed to multiple factors, one of them being the number of different GNN machine learning settings (e.g. single- and multi-graph settings). This renders the identification of a standardised method for differentially private GNN training significantly more challenging. Furthermore, GNN learning is not yet a fully established area of research, leaving a number of learning contexts unexplored. In this section we introduce two methods that have been used to achieve differentially private training on GNNs.

4.7.1. *DP-SGD Training of GNNs.* One of the most common methods to perform differentially private training in (non-graph) machine learning is differentially private stochastic gradient descent (DP-SGD) [78]. Here, a gradient descent step is privatised through bounding the gradient L_2-norm (clipping) and through the addition of calibrated noise, such that the output of the gradient calculation over two neighbouring datasets can –with high probability– not be well distinguished. This concept is not limited to SGD and can be applied to other first-order optimisation techniques, e.g. Adam. In standard machine learning, the clipping in DP-SGD is applied to the gradient *of each individual data point* to minimise the amount of noise that has to be added to the gradients. This method, naturally befitting structured databases with well-defined notions of what an "individual" gradient entails, does not seamlessly extend to graph machine learning in all cases. For graph classification tasks, for instance, each graph can be seen as an individual entity in a multi-graph dataset and, therefore, *graph-level* DP can be seen as a natural formulation in these learning settings. Here the standard procedure of DP-SGD can be transferred from database queries to graph learning tasks, matching database entries (rows) with individual graphs. This has been shown in [68]. Even though graph-level DP has not been deeply explored in research so far, we believe this to be an interesting and promising research are with multiple application areas, for example in medical settings with population graphs or brain networks (see Sections 5.2 and 5.3).

However, this approach is not directly transferable to GNNs in a single-graph setting, because the individual data points in a graph (where its nodes or edges) cannot be separated without breaking up the graph structure, which is essential to the message passing mechanism of GNNs. This not only precludes a notion of "per-sample" gradients, but also *privacy amplification by sub-sampling*, which states that a DP mechanism run on a random subsample of a population results in tighter privacy guarantees than when applied to the whole population [79]. To counter this effect, Igamberdiev et al. [5] implement a graph splitting method, which partitions the graph into smaller batches to approximate sub-sampling amplification and apply DP mechanisms to graph neural networks. Daigavane et al. [67] recently introduced an extension to DP-SGD to enable the training of node-differentially private multi-layer GNNs, whereas previous works were constrained to single-layer GNNs. They implement (sub-sampled) DP-SGD by sampling the local neighbourhood of a node and by generalising the approach of privacy amplification by sub-sampling through the analysis of affected per-sample gradient terms in a batch. In general however, no universally valid method to assign one of the DP formulations discussed above to the application of DP-SGD in GNNs on single-graph datasets has been proposed so far. Sajadmanesh and Gatica-Perez [7] address this problem by applying local DP on the node features, without protecting the graph structure. Given the so far limited amount of work in this area, we believe that the utilisation of DP-SGD in GNNs to represent an important open research question which needs to be explored in more detail.

4.7.2. *Private Aggregation of Teacher Ensembles.* Differentially private stochastic gradient descent is one of the most common methods to offer DP guarantees in machine learning. However, there are also alternative methods of preserving DP in machine learning, one being private aggregation of teacher ensembles (PATE), introduced by Papernot et al. [80]. PATE and its variants (e.g. [81, 82, 83, 84]) leverage an ensemble (a collection) of so-called *teacher models* that are trained on disjoint datasets containing sensitive data. These models are not published but instead used as teacher models for a separate student model. The student model cannot access any single teacher model nor the underlying data. It instead relies on a noisy voting algorithm performed across all teacher models to make a prediction [80]. One notable limitation of PATE is the reliance on a publicly available unlabelled dataset that is utilised by the teacher model. In general, this is a rather strong assumption, particularly in contexts relying on scarce, private datasets, such as medical data, limiting how generally it can be adopted as the means of differentially private training. In general, PATE should be considered a private student-teacher data labeling mechanism rather than necessarily representing a method for private collaborative training.

This shortcomings of PATE techniques are compounded by a low utility of PATE in graph settings as well as the limited generality beyond graph classification settings, as the physical separation of datasets in graph learning destroys structural information, significantly reducing the utility of the trained model [6]. Therefore, Olatunji et al. [6] recently introduced a framework named *PrivGNN*, which also leverages a student-teacher training paradigm for GNNs. The authors generate pseudolabels for public query nodes using specialised GNN models while adding noise to the predictions. The method requires two datasets: labeled private data for the teacher model and unlabeled public data for the student model. In the end, the public student model is released. It is trained using the noisy pseudo-labels and is differentially private based on the post-processing property of DP. The authors therefore implement a method for private release of trained graph neural networks and show their results on three node classification datasets.

5. Application Areas for DP on Graphs

In this section we discuss how our findings from above can be and have previously been applied to graph learning tasks in order to establish which formulations of DP are most suitable for each context, and give insights into a selection of potential application areas for DP on graph-structured data. Lastly, we provide an outlook on promising future research in those settings. We chose three exemplary learning contexts to allow us to cover all commonly used formulations of DP on graphs (i.e. node-level, edge-level and graph-level DP). Overall, more contexts relying on sensitive (or proprietary) data can benefit from a formalisation of DP, such as drug discovery [85] or location-based learning [86]. We leave an in-depth investigation of privacy in these settings as future work.

5.1. **Social Networks.** One of the more well-researched areas of private learning on graphs concerns social graphs [25, 27, 31, 41, 46], where the personally-identifying information is contained in the nodes of the graph and/or in the edges, defining the interactions between individuals, that could potentially allow to uniquely identify them (e.g. when spatio-temporal data is published [87]). As a result, there exist two concievable routes to perform private learning on such data: *edge-level* DP to protect the connections to other individuals in the graph and prevent unique identification of users like in [26, 18, 27] and *node-level* DP to protect the data of each individual itself (as well as the outgoing edges) like in [25, 30, 31, 46, 55, 56].

Numerous works have previously been employed to allow private release of social graphs or their associated statistics [30, 41, 49]. Sajadmanesh et al. [7] utilise locally differentially private GNNs in the context of social networks. The focus on social network data for the utilisation of DP in GNNs shows the high importance of protecting privacy in these settings, as well as the associated risks inherent to working with such datasets.

5.2. **Population Graphs.** The large amount of medical data collected by multiple medical institutions as well as personally through wearable devices, for instance, lead to mounting challenges of structuring these multi-modal datasets. One approach of handling this data heterogeneity is the construction of population graphs, which have found widespread adoption in medical research [88, 89, 90]. These data structures allow to encapsulate the information about patients across multiple departments and time periods (e.g. spatio-temporal patient data [91]), leveraging much more relevant information and leading to better predictions. One such scenario could involve representing each patient as a node and the whole patient population/cohort by a graph comprising the individuals, as described e.g. in [92, 93]. Connections between patients can, for instance, be based on their similarity (like in [94]). An advantage of creating such patient population graphs for the application of DP mechanisms is that the graph can be explicitly degree-bounded, limiting the impact of individual nodes on the graph structure.

Alternatively, each node can be patient-specific data about a single individual collected at different times by various specialists. Either of these contexts would benefit from the utilisation of *node-level* DP in order to quantify and limit the amount of information revealed when node-level data is processed or released, as they are relying on extremely sensitive data contained in each node.

5.3. **Brain Networks.** Here, we give an example for a graph classification problem on multi-graph datasets. In such setting it is not the information contained in a single node or inter-node connections that need to be kept private, but rather the information contained in a graph as a whole. One prime example of such dataset that contains sensitive information on a whole-graph level, rather than on the level of its individual constituents is a brain network graph [95]. Such data is used extensively in neuroimaging problems [96, 97, 98]. However, similarly to most medical datasets, due to the difficulty of obtaining such data (both because of the complexity of the task as well as of the privacy concerns) it is essential that the learning task is augmented with a suitable privacy-preservation mechanisms. In the case of brain network graphs, information about the value of individual voxels, or single connections to other voxels in the brain network are not necessarily personally identifying. Nonetheless, a collection of such interconnected points is considered to be a particularly sensitive medical dataset and it thus needs to be protected. For this setting, *graph-level* DP is a particularly suitable technique for data release. To date, there only exists a small number of such implementations of differentially private multi-graph learning and we envision that such formulation can gain significance as part of the future work in the area. We recall that DP deep learning on brain graphs (with learning tasks similar to [99] for instance) can be implemented through a straightforward utilisation of DP-SGD, similarly to Euclidean contexts.

6. Challenges and Outlook

In this section, we discuss a number of challenges associated with differentially private graph analytics, some of which can be attributed to the inter-connected nature of graphs, while others are inherent to DP itself. Note that we also discuss a number of potential complications arising in DP GNN training as well as in DP graph analytics and data release.

6.1. **Privacy Accounting.** Typically, in differentially private machine learning settings privacy loss can be bounded per individual data point (i.e. per image or table record), thus considering data points independently from each other, simplifying privacy loss accounting. However, due to the intrinsic inter-dependency of nodes in a graph, independence cannot be guaranteed and therefore quantifying the contribution of each individual becomes non-trivial. Thus, there arises a need for concrete definitions which would allow the data owner(s) to determine the exact formulation of differentially private training that is applicable in the specific application areas. As noted by [19], the guarantees given by *edge-level* DP and *node-level* DP have different implications, which are based on the exact features data owners wish to protect.

DP is inherently compositional, that is, DP algorithms composed with each-other yet again yield a DP algorithm [16]. However, the heterogeneous composition of different formulations of DP in a graph setting (e.g. simultaneously accounting for learning the adjacency matrix of a graph and for node classification) has not been studied previously, and we consider it a promising avenue for future research.

Furthermore, the reliance on random sampling of sub-populations from the dataset (*privacy amplification by sub-sampling*) can result in not only a better performance, but additionally much tighter privacy bounds, providing stronger privacy guarantees for the participants [79]. This approach has been applied by [67], e.g., where random sub-graphs are sampled during the private training of GNNs.

Another focus of DP is followed by techniques aiming to account for *individual privacy loss* [100]. Here, a "bespoke" privacy guarantee is given to each individual participating in a computation, typically combined with a method to automatically terminate their participation when their individual privacy budget is exhausted. As the process of deciding to continue or halt a computation by considering the currently spent privacy budget is an instance of *fully adaptive* composition, additional mechanisms are introduced: a privacy odometer (which tracks the privacy expenditure in the process of computation, without having to specify a privacy budget in advance) and the privacy filter (which stops the computation once the privacy budget is exceeded). The combination of these tools allows for a finer-grained control of the information that can be learned from each individual data point and potentially higher utility. The ability to compute individual privacy loss can allow a selective removal of individual nodes (and their corresponding edges), resulting in a much finer control of individual privacy expenditure. This method can permit tighter privacy bounding in settings where amplification by sub-sampling is not possible. However, it is also limited in applicability whenever the notion of a single individual within the graph is ill-defined.

6.2. **Privacy-Utility Trade-Offs.** As briefly discussed in section 4, DP in general adversely affects the utility of the model or of the results derived from a differentially private graph analytics. Utility if often measured by the accuracy of a query or with similar evaluation metrics. Therefore, similar to differentially private machine learning on Euclidean data or release of statistics derived from the sensitive data, there persists an issue of *privacy-utility trade-off*. This implies that the more "private" the result of the computation is (e.g. the lower the value of epsilon is), the less useful information can be inferred from that result not just by the adversary, but also by the end user of the trained model, potentially hindering the scientific progress based on the insights that could have otherwise been obtained from the study. This is further exacerbated by the inter-connected nature of the graphs, as it is not possible to guarantee independence of individual nodes, as we discussed above. Therefore, operations that limit the amount of information that can be derived from these nodes (e.g. through DP statistics release) affect not just the individuals, but also additional nodes connected to them. Thus, the utility loss can become more problematic when compared to datasets with independent data points and inflict additional penalties on the results of the computation. We note that this discussion is relevant to both graph datasets and GNNs, as the nature of GNN learning can only make full use of the data if these properties of graphs are preserved. Relying on GNN models pre-trained on publicly available data (similar to [101, 102, 103]) could severely reduce the negative impact that DP has on utility, when used in transfer learning contexts. Here a model is trained on public data and subsequently fine-tuned on private data where higher privacy can be achieved, while having better utility. This approach was demonstrated in [78] and more recently in [104] for non-graph machine learning tasks, demonstrating that –whereas training to the same utility *from scratch*– requires about one order of magnitude more data, results comparable to non-private training can easily be achieved by transfer learning.

6.3. **Computational Performance.** Beyond the aforementioned trade-offs in model generalisation performance, the utilisation of differential privacy is also associated with a computational performance overhead when employed in deep learning settings. This can be attributed to a requirement for per-sample gradient calculation, imposing a significant burden on model performance at train time. Moreover, due to noise addition and gradient clipping, models typically converge more slowly, thus prolonging the required training time [105].

6.4. **Interpretability of DP in Graphs.** DP can often be difficult to reason over from the perspectives of fairness [106] and explainability [107]. Moreover, its correct application is complicated by the introduction of unintuitive parameters like ε or δ [108, 109], or by the requirement to understand additional DP definitions like node, edge or graph-level DP. Thus, besides systems which automate sensitivity calculations and the application of DP to generic machine learning workflows [110], works similar to [111] are required, which investigate user expectations and interpretations of DP, paving the way for an improved *user experience* for practitioners.

Interpretability of GNNs in general is a highly discussed task in literature. The authors in [68] use an explainability method called *GNNExplainer* [112] to visualise and quantify the similarity between graph neural networks trained with and without DP-SGD to evaluate whether the privately trained network considers the same edges in the graph as important as the network trained with standard ML. We see potential in methods like these to get a better insight into differenitially private GNNs and increase their interpretability.

6.5. **Synthetic Graph Generation.** One final graph learning context that still remains an open challenge is private synthetic graph generation. The ability to generate synthetic samples allows one to augment existing datasets with additional data points in a privacy-neutral way, resulting in more diverse data representations. This, in turn, improves utility of the model trained on this data as well as empirically reduces the effectiveness of inference attacks [113]. There exist prior works in the area [66, 35, 24, 46, 29] that allow to generate graph-structured data in a private manner, however, authors outline a number of limitations. Firstly, the effect of privacy-utility trade-off is much more profound in graph generation tasks, forcing the model owner to either lower the privacy guarantees or to generate graphs of much lower utility. Secondly, the number of DP formulations that are applicable to synthetic graph generation is rather limited: To-date, stronger privacy formulations like node-level DP are not yet widespread in the setting of private synthetic graph generation. Chen et al. [54], e.g., explore synthetic graph generation of social graphs under *edge-level* DP. Gupta et al. [28] introduce a method for synthetic graph generation specifically tailored to graph cuts. Additionally, graph generation has so far been limited to simple benchmark datasets and has not been widely investigated under the lens of the privacy-utility trade-off in more challenging contexts. Qin et al. [46], e.g., therefore resort to LDP to generate synthetic decentralized social graphs. Since this particular application of private graph-based learning is relatively new, we identify this to be a promising area of future work in the graph domain.

7. Conclusion

In this work, we explore and systematise applications of differential privacy in graph analytics and on graph neural networks. We discovered 51 works that perform differentially private data processing of graph structures, which we classify by the DP formulations employed in each work and summarise our findings in Table 1. We identify three main DP formulations with regards to the attributes of graphs considered to be sensitive: (1) edge-level, (2) node-level, and (3) graph-level differential privacy. We additionally discuss machine learning tasks (in particular those relying on GNNs) that require utilisation of sensitive graph-structured data and could hence benefit from a formalisation of differentially private learning. Subsequently, we discuss the limitations of DP when applied to such learning contexts, some of which are inherent to the choice of DP learning setting and some attributable to the inter-connected nature of graph structures specifically. We conclude our discussion with an analysis of graph learning tasks on sensitive data, summarise which DP formulations are suitable for different learning problems and identify promising areas of future research. We hope that our work offers practitioners a helpful overview of the current state of DP employed in graph-based learning, and will stimulate both foundational and application-focused future research.

8. Acknowledgements

D.U. received funding from the TUM-Imperial College London Joint Academy of Doctoral Studies. R.B. received funding from the German Research Foundation (DFG) Special Priority Programme 2177 "Radiomics: next generation of biomedical imaging", project reference 428215948. D.R. and G.K. received funding from the UK Research and Innovation London Medical Imaging & Artificial Intelligence Centre for Value Based Healthcare. The funders played no role in the design of the study, the preparation of the manuscript or the decision to publish.

References

[1] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. Exploring graph neural networks for stock market predictions with rolling window analysis. *arXiv preprint arXiv:1909.10660*, 2019.

[2] Adrien Benamira, Benjamin Devillers, Etienne Lesot, Ayush K Ray, Manal Saadi, and Fragkiskos D Malliaros. Semi-supervised learning and graph neural networks for fake news detection. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 568–569. IEEE, 2019.

[3] Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles Roberts, Jian Tang, et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021.

[4] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[5] Timour Igamberdiev and Ivan Habernal. Privacy-preserving graph convolutional networks for text classification. *arXiv preprint arXiv:2102.09604*, 2021.

[6] Iyiola E Olatunji, Thorben Funke, and Megha Khosla. Releasing graph neural networks with differential privacy guarantees. *arXiv preprint arXiv:2109.08907*, 2021.

[7] Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. *CoRR abs/2006.05535*, 12, 2020.

[8] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[9] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. Graphmi: Extracting private graph data from graph neural networks. *arXiv preprint arXiv:2106.02820*, 2021.

[10] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Dependence makes you vulnerable: differential privacy under dependent tuples. In *NDSS*, volume 16, pages 21–24, 2016.

[11] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[12] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. *arXiv preprint arXiv:2101.06570*, 2021.

[13] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429*, 2021.

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[15] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[16] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[17] Michael Carl Tschantz, Shayak Sen, and Anupam Datta. Sok: differential privacy as a causal property. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 354–371. IEEE, 2020.

[18] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 4(11):1146–1157, 2011.

[19] Pennsylvania State University Adam Smith. Differentially private analysis on graphs, 2016.

[20] Xingxing Xiong, Shubo Liu, Dan Li, Zhaohui Cai, and Xiaoguang Niu. A comprehensive survey on local differential privacy. *Security and Communication Networks*, 2020, 2020.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[22] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[23] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178. IEEE, 2009.

[24] Darakhshan J Mir and Rebecca N Wright. A differentially private graph estimator. In *2009 IEEE International Conference on Data Mining Workshops*, pages 122–129. IEEE, 2009.

[25] Johannes Gehrke, Edward Lui, and Rafael Pass. Towards privacy for social networks: A zero-knowledge based definition of privacy. In *Theory of cryptography conference*, pages 432–449. Springer, 2011.

[26] Ashwin Machanavajjhala, Aleksandra Korolova, and Atish Das Sarma. Personalized social recommendations-accurate or private? *arXiv preprint arXiv:1105.4254*, 2011.

[27] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98, 2011.

[28] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of cryptography conference*, pages 339–356. Springer, 2012.

[29] Vishesh Karwa and Aleksandra B Slavković. Differentially private graphical degree sequences and synthetic graphs. In *International Conference on Privacy in Statistical Databases*, pages 273–285. Springer, 2012.

[30] Darakhshan Mir and Rebecca N Wright. A differentially private estimator for the stochastic kronecker graph model. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 167–176, 2012.

[31] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 87–96, 2013.

[32] Shixi Chen and Shuigeng Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 653–664, 2013.

[33] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, pages 457–476. Springer, 2013.

[34] Entong Shen and Ting Yu. Mining frequent graph patterns with differential privacy. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 545–553, 2013.

[35] Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 329–340. Springer, 2013.

[36] Yue Wang and Xintao Wu. Preserving differential privacy in degree-correlation based graph generation. *Transactions on data privacy*, 6(2):127, 2013.

[37] Rui Chen, Benjamin CM Fung, S Yu Philip, and Bipin C Desai. Correlated network data publication via differential privacy. *The VLDB Journal*, 23(4):653–676, 2014.

[38] Wentian Lu and Gerome Miklau. Exponential random graph estimation under differential privacy. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 921–930, 2014.

[39] Davide Proserpio, Sharon Goldberg, and Frank McSherry. Calibrating data to sensitivity in private data analysis: A platform for differentially-private analysis of weighted datasets. *Proceedings of the VLDB Endowment*, 7(8):637–648, 2014.

[40] Sofya Raskhodnikova and Adam Smith. *Private Analysis of Graph Data*, pages 1–6. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[41] Christine Task and Chris Clifton. What should we protect? defining differential privacy for social network analysis. In *State of the Art Applications of Social Network Analysis*, pages 139–161. Springer, 2014.

[42] Wei-Yen Day, Ninghui Li, and Min Lyu. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data*, pages 123–138, 2016.

[43] Zach Jorgensen, Ting Yu, and Graham Cormode. Publishing attributed social graphs with formal privacy guarantees. In *Proceedings of the 2016 international conference on management of data*, pages 107–122, 2016.

[44] Sofya Raskhodnikova and Adam Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 495–504. IEEE, 2016.

[45] Yue Wang, Xintao Wu, and Donghui Hu. Using randomized response for differential privacy preserving data collection. In *EDBT/ICDT Workshops*, volume 1558, pages 0090–6778, 2016.

[46] Zhan Qin, Ting Yu, Yin Yang, Issa Khalil, Xiaokui Xiao, and Kui Ren. Generating synthetic decentralized social graphs with local differential privacy. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 425–438, 2017.

[47] Tianqing Zhu, Gang Li, Wanlei Zhou, and S Yu Philip. *Differential privacy and applications*. Springer, 2017.

[48] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[49] Kamalkumar R Macwan and Sankita J Patel. Node differential privacy in social graph degree publishing. *Procedia computer science*, 143:786–793, 2018.

[50] Raman Arora and Jalaj Upadhyay. On differentially private graph sparsification and applications. *Advances in neural information processing systems*, 32:13399–13410, 2019.

[51] Jonathan Ullman and Adam Sealfon. Efficiently estimating erdos-renyi graphs with node differential privacy. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Haipei Sun, Xiaokui Xiao, Issa Khalil, Yin Yang, Zhan Qin, Hui Wang, and Ting Yu. Analyzing subgraph statistics from extended local views with decentralized differential privacy. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 703–717, 2019.

[53] Zhang Yuxuan, Wei Jianghong, Li Ji, Liu Wenfen, and Hu Xuexian. Graph degree histogram publication method with node-differential privacy. *Journal of Computer Research and Development*, 56(3):508, 2019.

[54] Xihui Chen, Sjouke Mauw, and Yunior Ramírez-Cruz. Publishing community-preserving attributed social graphs with a differential privacy guarantee. *Proceedings on Privacy Enhancing Technologies*, 2020(4):131–152, 2020.

[55] Ganghong Liu, Xuebin Ma, and Wuyungerile Li. Publishing node strength distribution with node differential privacy. *IEEE Access*, 8:217642–217650, 2020.

[56] Sen Zhang, Weiwei Ni, and Nan Fu. Community preserved social graph publishing with node differential privacy. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1400–1405. IEEE, 2020.

[57] Hailong Zhang, Sufian Latif, Raef Bassily, and Atanas Rountev. Differentially-private control-flow node coverage for software usage analysis. In *USENIX Security Symposium*, pages 1021–1038, 2020.

[58] Masooma Iftikhar and Qing Wang. dk-projection: Publishing graph joint degree distribution with node differential privacy. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 358–370. Springer, 2021.

[59] Hendrik Fichtenberger, Monika Henzinger, and Wolfgang Ost. Differentially private algorithms for graphs under continual observation. *arXiv preprint arXiv:2106.14756*, 2021.

[60] Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. Locally differentially private analysis of graph statistics. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[61] Sun Lan, Huang Xin, Wu Yingjie, and Guo Yongyi. Sensitivity reduction of degree histogram publication under node differential privacy via mean filtering. *Concurrency and Computation: Practice and Experience*, 33(8):e5621, 2021.

[62] Wenfen Liu, Bixia Liu, Qiang Xu, and Hui Lei. Graph node strength histogram publication method with node differential privacy. In *Journal of Physics: Conference Series*, volume 1757, page 012186. IOP Publishing, 2021.

[63] Adam Sealfon and Jonathan Ullman. Efficiently estimating erdos-renyi graphs with node differential privacy. *Journal of Privacy and Confidentiality*, 11(1), 2021.

[64] Siyuan Xia, Beizhen Chang, Karl Knopf, Yihan He, Yuchao Tao, and Xi He. Dpgraph: A benchmark platform for differentially private graph analysis. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2808–2812, 2021.

[65] Xu Zheng, Lizong Zhang, Kaiyang Li, and Xi Zeng. Efficient publication of distributed and overlapping graph data under differential privacy. *Tsinghua Science and Technology*, 27(2):235–243, 2021.

[66] Xu Zheng, Nicholas McCarthy, and Jer Hayes. Network generation with differential privacy. *arXiv preprint arXiv:2111.09085*, 2021.

[67] Ameya Daigavane, Gagan Madan, Aditya Sinha, Abhradeep Guha Thakurta, Gaurav Aggarwal, and Prateek Jain. Node-level differentially private graph neural networks. *arXiv preprint arXiv:2111.15521*, 2021.

[68] Tamara T. Mueller, Johannes C. Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, and Georgios Kaissis. Differentially private graph classification with gnns, 2022.

[69] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 15–24, 2016.

[70] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

[71] Xiaoxiao Li, Nicha C Dvornek, Yuan Zhou, Juntang Zhuang, Pamela Ventola, and James S Duncan. Graph neural network for interpreting task-fmri biomarkers. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 485–493. Springer, 2019.

[72] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

[73] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1850–1859, 2020.

[74] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[75] Adam Sealfon. Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 29–41, 2016.

[76] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[77] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR, 2016.

[78] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[79] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *arXiv preprint arXiv:1807.01647*, 2018.

[80] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[81] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[82] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

[83] Sergio Yovine, Franz Mayr, Sebastián Sosa, and Ramiro Visca. An assessment of the application of private aggregation of ensemble models to sensible data. *Machine Learning and Knowledge Extraction*, 3(4):788–801, 2021.

[84] Qiuchen Zhang, Jing Ma, Jian Lou, Li Xiong, and Xiaoqian Jiang. Towards training robust private aggregation of teacher ensembles under noisy labels. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1103–1110. IEEE, 2020.

[85] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13(1):1–23, 2021.

[86] Moritz Kessel, Peter Ruppel, and Florian Gschwandtner. Bigml: A location model with individual waypoint graphs for indoor location-based services. *PIK*, 13, 2010.

[87] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3(1):1–5, 2013.

[88] Pietro Barbiero, Ramon Viñas Torné, and Pietro Lió. Graph representation forecasting of patient's medical conditions: Toward a digital twin. *Frontiers in genetics*, 12, 2021.

[89] M Duplaga et al. Universal electronic health record mudr. *Transformation of Healthcare with Information Technologies*, 105:190, 2004.

[90] R Müller, O Thews, C Rohrbach, M Sergl, and K Pommerening. A graph-grammar approach to represent causal, temporal and other contexts in an oncological patient record. *Methods of information in medicine*, 35(02):127–141, 1996.

[91] Chuanren Liu, Fei Wang, Jianying Hu, and Hui Xiong. Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 705–714, 2015.

[92] Jianliang Gao, Tengfei Lyu, Fan Xiong, Jianxin Wang, Weimao Ke, and Zhao Li. Mgnn: a multimodal graph neural network for predicting the survival of cancer patients. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1697–1700, 2020.

[93] Chengsheng Mao, Liang Yao, and Yuan Luo. Medgcn: Graph convolutional networks for multiple medical tasks. *arXiv preprint arXiv:1904.00326*, 2019.

[94] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. *Medical image analysis*, 48:117–130, 2018.

[95] Edward T Bullmore and Danielle S Bassett. Brain graphs: graphical models of the human brain connectome. *Annual review of clinical psychology*, 7:113–140, 2011.

[96] Qingbao Yu, Erik B Erhardt, Jing Sui, Yuhui Du, Hao He, Devon Hjelm, Mustafa S Cetin, Srinivas Rachakonda, Robyn L Miller, Godfrey Pearlson, et al. Assessing dynamic brain graphs of time-varying connectivity in fmri data: application to healthy controls and patients with schizophrenia. *Neuroimage*, 107:345–355, 2015.

[97] Olaf Sporns. From simple graphs to the connectome: networks in neuroimaging. *Neuroimage*, 62(2):881–886, 2012.

[98] Mathilde Ménoret, Nicolas Farrugia, Bastien Pasdeloup, and Vincent Gripon. Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 618–622. IEEE, 2017.

[99] Jonas Richiardi, Sophie Achard, Horst Bunke, and Dimitri Van De Ville. Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. *IEEE Signal processing magazine*, 30(3):58–70, 2013.

[100] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[101] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867, 2020.

[102] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the*

14th ACM International Conference on Web Search and Data Mining, pages 265–273, 2021.

[103] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160, 2020.

[104] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data), 2021.

[105] Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy, 2022.

[106] Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, pages 15–19, 2020.

[107] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.

[108] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. " i need a better description": An investigation into user expectations for differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3037–3052, 2021.

[109] Georgios Kaissis, Moritz Knolle, Friederike Jungmann, Alexander Ziller, Dmitrii Usynin, and Daniel Rueckert. A unified interpretation of the gaussian mechanism for differential privacy through the sensitivity index. *arXiv preprint arXiv:2109.10528*, 2021.

[110] Dmitrii Usynin, Alexander Ziller, Moritz Knolle, Daniel Rueckert, and Georgios Kaissis. An automatic differentiation system for the age of differential privacy. *arXiv preprint arXiv:2109.10573*, 2021.

[111] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 309–315, 2019.

[112] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. *arXiv preprint arXiv:1903.03894*, 2019.

[113] William Paul, Yinzhi Cao, Miaomiao Zhang, and Phil Burlina. Defending medical image diagnostics against privacy attacks using generative methods: Application to retinal diagnostics. In *Clinical Image-Based Procedures, Distributed and Collaborative Learning, Artificial Intelligence for Combating COVID-19 and Secure and Privacy-Preserving Machine Learning*, pages 174–187. Springer, 2021.

9. Differentially Private Graph Neural Networks for Whole-Graph Classification

Tamara T. Mueller, Johannes C Paetzold, Chinmay Prabhakar, Dmitrii Usynin, Daniel Rueckert, and Georgios Kaissis

Journal: IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 45, Issue: 6, 01 June 2023.

Synopsis: Graph Neural Networks (GNNs) have established themselves as state-of-the-art for many machine learning applications such as the analysis of social and medical networks. Several among these datasets contain privacy-sensitive data. Machine learning with differential privacy is a promising technique to allow deriving insight from sensitive data while offering formal guarantees of privacy protection. However, the differentially private training of GNNs has so far remained under-explored due to the challenges presented by the intrinsic structural connectivity of graphs. In this work, we introduce a framework for differential private graph-level classification. Our method is applicable to graph deep learning on multi-graph datasets and relies on differentially private stochastic gradient descent (DP-SGD). We show results on a variety of datasets and evaluate the impact of different GNN architectures and training hyperparameters on model performance for differentially private graph classification, as well as the scalability of the method on a large medical dataset. Our experiments show that DP-SGD can be applied to graph classification tasks with reasonable utility losses. Furthermore, we apply explainability techniques to assess whether similar representations are learned in the private and non-private settings. Our results can also function as robust baselines for future work in this area.

Contributions of thesis author: data processing, design and implementation of source code, execution of experiments, manuscript writing.

Differentially Private Graph Neural Networks for Whole-Graph Classification

Tamara T. Mueller[1,2], **Johannes C. Paetzold**[3,6], **Chinmay Prabhakar**[4], **Dmitrii Usynin**[1,2], **Daniel Rueckert**[1,5], **and Georgios Kaissis**[1,2,5]

[1]Chair for AI in Medicine and Healthcare, Department of Informatics, Technical University of Munich
[2]Department of Diagnostic and Interventional Radiology, Faculty of Medicine, Technical University of Munich
[3]Department of Informatics, Technical University of Munich
[4]Department of Quantitative Bio Medicine, University of Zurich
[5]Department of Computing, Imperial College London
[6]Institute for Tissue Engineering and Regenerative Medicine, Helmholtz Zentrum München

Abstract—Graph Neural Networks (GNNs) have established themselves as state-of-the-art for many machine learning applications such as the analysis of social and medical networks. Several among these datasets contain privacy-sensitive data. Machine learning with differential privacy is a promising technique to allow deriving insight from sensitive data while offering formal guarantees of privacy protection. However, the differentially private training of GNNs has so far remained under-explored due to the challenges presented by the intrinsic structural connectivity of graphs. In this work, we introduce a framework for differential private graph-level classification. Our method is applicable to graph deep learning on multi-graph datasets and relies on differentially private stochastic gradient descent (DP-SGD). We show results on a variety of datasets and evaluate the impact of different GNN architectures and training hyperparameters on model performance for differentially private graph classification, as well as the scalability of the method on a large medical dataset. Our experiments show that DP-SGD can be applied to graph classification tasks with reasonable utility losses. Furthermore, we apply explainability techniques to assess whether similar representations are learned in the private and non-private settings. Our results can also function as robust baselines for future work in this area.

1 Introduction

The introduction of geometric deep learning, and more specifically Graph Neural Networks (GNNs) [1], [2], has enabled training ML models on data in non-Euclidean spaces with state-of-the-art performance in many applications. GNNs are able to directly leverage the graph structure of the data and propagate the information stored in nodes of the graph along the edges connecting nodes with each other. Thus, the information flow through the network respects the underlying topology of the graph.

In general, GNNs have been employed in three types of problem areas: node classification, edge prediction, and graph classification. In this work, we focus on graph classification tasks. In the setting of graph classification (also termed graph property prediction), the dataset consists of multiple independent graphs and a GNN is trained to predict one label for each individual graph, predicting a specific property of the whole graph. Application areas of geometric deep learning range from social networks [3] to medical applications [4], [5], drug discovery or molecule classification [6], spatial biological networks [7] and shape analysis [8]. Drawing meaningful insights in many of these application areas fundamentally relies upon the utilisation of privacy-sensitive, often scarce, training data belonging to individuals. For example when using functional magnetic resonance imaging (fMRI) for identifying disease-specific biomarkers of brain connectivity like in [4] and [9], the graph data encodes sensitive, patient-specific medical data.

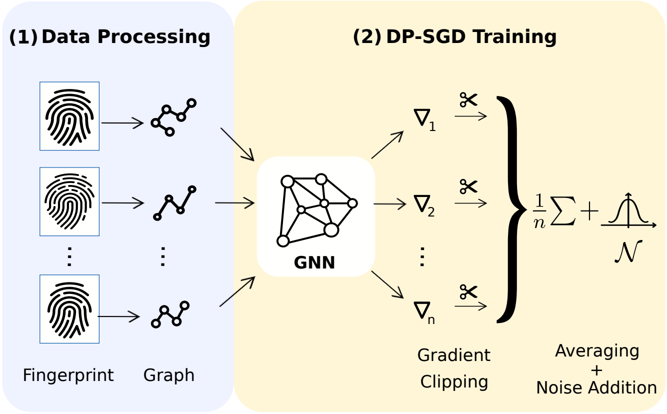


Fig. 1: Overview of our differentially private training method for graph classification on a fingerprint dataset. In step (1) the fingerprint images are converted into graphs, which are then in step (2) passed to a GNN model, which is trained with differentially private stochastic gradient descent (DP-SGD). The individual gradients are clipped, then averaged and Gaussian noise is added.

The reliance on sensitive data in machine learning holds potential for misuse and can therefore be associated with the risks to individual participants' privacy. Various machine learning contexts have been shown vulnerable to be exploited by malicious actors, resulting in a leakage of private attributes [10], of membership information [11]

The source code is available at https://github.com/tamaramueller/DP-GNNs.

or even in full dataset reconstruction [12], [13]. In graph machine learning, the data and the models trained on that data are *by design* more vulnerable to adversarial attacks targeting privacy of the data owners. This is attributed to the fact that graphs incorporate additional information that is absent from typical Euclidean training contexts, such as the relational information about the nodes in the graph. This auxiliary, highly descriptive information can be leveraged by an adversary to assist them in sensitive information extraction, which has been demonstrated in a number of prior works [14], [15], [16]. Such attacks can also be facilitated by the choice of learning context in cases the model is trained collaboratively. For instance, transductive collaborative learning renders attacks aimed at disclosing the membership of individual training points trivial [15]. Of note, such additional information embedded in graphs is often essential for effective GNN training and is, thus, non-trivial to privatise or remove, as it would be highly detrimental to the performance of the model.

It is thus apparent that the implementation of privacy-enhancing techniques is required to facilitate the training of models of sensitive graph-structured data, but such techniques must also respect the particularities of graph machine learning. Our work utilises a formal method of privacy preservation termed differential privacy (DP) [17] which, when applied to machine learning training, is able to objectively quantify the privacy loss for individual input data points. DP methods have been successfully applied to numerous problems such as medical image analysis [18], [19], natural language processing (NLP) [20], reinforcement learning [21] or generative models [22] and have shown promising results. DP guarantees that the information gain from observing the output of an algorithm trained on datasets differing in one individual is (sometimes with high probability), bounded by a (typically small) constant.

In this work, motivated by the above-mentioned requirements for objective privacy guarantees in machine learning tasks involving graph-structured data, we study the problem of efficient differentially private graph neural network training for graph classification tasks. To the best of our knowledge, ours is the first work that demonstrates the application of differential privacy to whole graph classification tasks. We investigate and evaluate privacy-utility trade-offs on several datasets and compare the learned representations between DP and non-DP trained models using explianability methods for GNNs. This comparison can offer insights into differences regarding model parameters, which are considered as important for the decision making, under different training conditions. In our work, we extend the utilisation of differentially private stochastic gradient descent (DP-SGD) [23], a technique designed for the training of regular neural networks. Due to its compatibility with existent deep learning workflows, it can be seamlessly adapted to GNN use cases and therefore offers high generalisability to new model architectures and problem spaces. We demonstrate that DP-SGD can be applied to graph learning and evaluate our results with respect to privacy budgets and network performance on five different datasets. Combined with our investigation of the explainability technique *GNNExplainer* to determine differences between DP and non-DP models, this work can serve as a baseline for future work in this area. Our contributions can be summarised as follows:

1) We formally extend the application of DP-SGD to graph classification tasks with GNNs;
2) To demonstrate its utility, we apply our method to commonly utilised graph neural networks on a number of benchmark and real-world datasets and investigate the effects of DP training on model utility and privacy guarantees;
3) To assess how similar the representations between privately and non-privately trained models are, we apply GNNExplainer, a state-of-the-art explainability technique tailored to graph neural networks.

2 Related Work

Specific facets of differentially private graph analysis have been addressed in prior work: Since the introduction of differentially private computation on graph data in 2007 by Nissim et al. [24], *node-level* and *edge-level* DP have been established as the two DP formalisms on graphs [25]. As discussed in the *Theory* section below, the definition of DP relies on the notion of adjacent datasets, that is, datasets differing in the data of one individual. In the setting of tabular data for example, two datasets are adjacent if they differ in one row. In node-level DP, two graph datasets are interpreted as adjacent if one node and its incident edges is inserted or removed. For edge-level DP, on the other hand, two datasets are regarded as adjacent if they differ in exactly one edge. As real-world graphs are prevalently sparse, the removal of a single node can severely alter the graph's structure [26], whereas removal of an edge typically has a less severe impact on the resulting graph structure.

Implementations of the aforementioned techniques have been presented in the context of graph neural network training. For instance, Igamberdiev et al. [27] explore the application of DP on Graph Convolutional Networks (GCNs) [28] for node classification. They evaluate privacy guarantees for text classification on benchmark datasets and achieve rigorous privacy guarantees while maintaining high model performance. Daigavan et al. [29] formalise the notion of node-level DP on one-layer GNNs with an extension of privacy amplification by sampling to GNNs and evaluate their method on several benchmark datasets in node classification tasks. Different approaches to the here introduced application of differential privacy have been explored in the context of federated learning on graphs and locally private graph neural network training. Zhou et al. [30], for example, introduce a vertically federated GNN for node classification tasks and Sajadmanesh et al. [31] introduce a framework to train locally private GNNs. These works stand in contrast to the notion of graph-level DP, which ensures data privacy of a graph as a whole.

DP is one of the most frequently used methods in deep learning that offer privacy guarantees. Furthermore, it is the only approach that gives formal guarantees for privacy as well as a quantification of the guaranteed privacy. However, there exist other empirical methods next to differential privacy that allow to privatise sensitive data of individuals, which have also been applied to GNN training in node classification and edge prediction tasks. Liao et al. [32] introduce a method to filter specific node feature attributes using adversarial training of GNNs and therefore achieve a strong defence against inference attacks. Their method is in parallel to our work, since they do not ensure differential privacy guarantees for each graph as a whole, but instead address an information obfuscation problem where the goal of an adversary is to infer specific node attributes in a graph. Other works like the privacy-preserving network embedding introduced by Han et al. [33] and the privacy-preserving GCN model by Hu et al. [34] also do not give differential privacy guarantees. They show other methods for protecting private links in graph-structured data [33] and user-specific sensitive node features [34], respectively.

However, to our knowledge, the application of DP algorithms specifically to graph property prediction has neither been formalised nor evaluated.

3 Theoretical Preliminaries

In this section, we introduce and formalise the theory to train graph neural networks for graph property prediction using the concept of differentially private stochastic gradient descent (DP-SGD).

3.1 GNNs for Graph Property Prediction

The objective of graph classification (also known as graph property prediction) is to predict a specific property of interest for an entire graph $\mathcal{G}$. In our examples, $\mathcal{G}$ represents an unweighted and undirected graph with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. The nodes $\mathcal{V}$ are represented by a vector or a matrix of node features. Graph classification aims to predict a property for each graph $\mathcal{G}_i$, $i \in [1, \cdots, N]$ in a multi-graph dataset $D = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$ with N graphs. A GNN used for graph property prediction needs to map the embedded node features into a unified representation of the whole graph using a readout layer (e.g. global max pooling). This single unified embedded graph representation allows to learn a prediction for the whole graph.

3.2 Differential Privacy

Differential Privacy (DP) [17] is a theoretical framework and collection of techniques aimed at enabling analysts to draw conclusions from datasets while safeguarding individual privacy. Intuitively, an algorithm preserves DP if its outputs are approximately invariant to the inclusion or exclusion of a single individual in the dataset over which the algorithm is executed. The DP guarantee is given in terms of probability mass/density of the algorithm's outputs.

In the current study, we assume that an analyst $\mathcal{A}$ is entrusted with a multi-graph database D of cardinality N containing privacy-sensitive graphs $\mathcal{G}_i \in D, i \in [1, \cdots, N]$ by a group of individuals. We assume that each individual's graph is only present in the database once. From D, an *adjacent* database D' of cardinality $N \pm 1$ can be constructed by adding or removing a single individual's graph. We denote adjacency by $D \simeq D'$. The set (*universe*) of all adjacent databases forms a metric space X with associated metric d_X, in our case, the Hamming metric.

We additionally assume that $\mathcal{A}$ executes a *query function* f over an element of X. In our study, the application of f represents a sequential composition of the forward pass, loss calculation and gradient computation of a graph neural network for each individual input (training example) to f. We then define the global L_2-sensitivity of f as follows:

Definition 3.1 (Global L_2-sensitivity of f). Let f, X and d_X be defined as above. Additionally, let Y be the metric space of f's outputs with associated metric d_Y. When Y is the Euclidean space and d_Y the L_2 metric, we define the (global) L_2-sensitivity Δ of f as:

$$\Delta := \max_{D, D' \in X, D \simeq D'} \frac{d_Y(f(D), f(D'))}{d_X(D, D')}. \tag{1}$$

We remark that the maximum is taken over all adjacent database pairs in X. Moreover, Δ describes a Lipschitz condition on f, implying that $\Delta \equiv K_f$, where K_f is the Lipschitz constant of f. This in turn implies that $\Delta = \sup \|\nabla f\|_2$. In our case, the L_2-sensitivity of the loss function therefore corresponds to the upper bound on its gradient.

We can now define the Gaussian Mechanism on f:

Definition 3.2 (Gaussian Mechanism). Let f, Δ be defined as above. The Gaussian mechanism $\mathcal{M}$ operates on the outputs of f, $\mathbf{y} = f(x)$, where $\mathbf{y} \in \mathbb{R}^n$ as follows:

$$\mathcal{M}(\mathbf{y}) = \mathbf{y} + \xi, \tag{2}$$

where $\xi \sim \mathcal{N}(0, \sigma\mathbb{I}^n)$, σ is calibrated to Δ, and $\mathbb{I}^n$ is the identity matrix with n diagonal elements.

When σ is appropriately calibrated to Δ, $\mathcal{M}$ preserves (ε, δ)-DP:

Definition 3.3 ((ε, δ)-DP). $\mathcal{M}$ preserves (ε, δ)-DP if, $\forall\, S \subseteq \mathrm{Range}(\mathcal{M})$ and all adjacent databases D, D' in X:

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq e^{\varepsilon}\mathbb{P}(\mathcal{M}(D') \in S) + \delta. \tag{3}$$

We remark that the definition is symmetric.

3.3 DP-SGD

Abadi et al. [23] introduced an extension to stochastic gradient descent (SGD), termed DP-SGD to enable the differentially private training of neural networks. Here, at each training step, the Gaussian Mechanism is used to privatise the individual gradients of each training example before the model parameters are updated. However, since the sensitivity of the loss function in deep neural networks is – in general – unbounded, the gradient L_2-norm of each individual training example is *clipped*, that is, projected to an L_2-ball of a predefined radius to artificially induce a bounded sensitivity condition before noise is applied. Tracking the privacy expenditure over the course of training (*privacy accounting*) is enabled through the *composition* property of DP, stating that repeated application of DP algorithms over the same data predictably degrades the privacy guarantees. In our study, a relaxation of DP termed Rényi DP (RDP) [35] is used for privacy accounting, due to its favourable compositional properties. RDP guarantees can be converted to (ε, δ)-DP.

DP-SGD is widely regarded as the gold-standard for privacy preserving deep learning, as it is generically applicable to all types of gradient-based optimisation and protects both features and labels. It can be easily adapted to e.g. regression or generative modelling workflows. Other DP methods are substantially less flexible [23]. Private aggregation of teacher ensembles (PATE) [36] for example, is only usable for classification tasks and requires large public datasets, which, especially in the medical field, cannot be procured in many cases.

3.4 DP Notions on Graph-Structured Datasets

There exist three major tasks in the context of GNN training: node classification/regression, edge prediction, and graph classification/regression. Similar to the existence of multiple tasks in graph deep learning, there also exist different notions of DP on graph-structured datasets, that specifically relate to different notions of adjacent datasets. For *node-level DP*, two datasets are interpreted as adjacent, if they vary in one node and all its adjacent edges [37]. If the notion of adjacent datasets is based on the inclusion or exclusion of one edge, this notion of DP is called *edge-level DP* [38]. Node-level DP is a strictly stronger privacy guarantee in comparison to edge-level DP [26]. As real-world graphs are prevalently sparse, the removal of a single node can severely alter the graph's structure [26], whereas removal of an edge typically has a less severe impact on the resulting graph structure. However, in case of multi-graph datasets, a third notion of DP can come into play. Here, two datasets can be defined to be adjacent if they differ in one graph. The resulting DP-guarantee is then *graph-level DP* [39], which we utilise in this work. For more details we refer to [39].

4 Experiments

4.1 Datasets

We evaluate the application of DP-SGD in the context of graph property prediction tasks on five datasets. We rely on three publicly available benchmark datasets, a dataset from the UK Biobank [40], and a synthetic dataset, generated to provide a reproducible and easy to control proof-of-concept. The three benchmark datasets tackle the problems of molecule classification (Molbace), fingerprint classification, and Left Bundle Branch Block (LBBB) detection on electrocardiogram (ECG) data. Table 1 provides an overview of the datasets and their characteristics and more detailed information about the datasets can be found in the Appendix.

Dataset	Mean num. nodes	Num. graphs	Num. node features	Num. classes
Synthetic	20	1,000	9	2
Fingerprints	7.6	1,900	2	4
Molbace	34	1,513	9	2
ECG	12	1,125	512	2
Organ Meshes	7546.7	151,910	3	5

TABLE 1: Overview of the utilised datasets and their characteristics. We report the mean number of nodes, in case the dataset contains graphs of varying sizes.

Synthetic Dataset

In order to derive a proof-of-concept of the novel application of DP-SGD on graph classification tasks, we construct a synthetic dataset, in which parameters can be manually controlled to create an easily controllable dataset where high accuracy can be achieved in a non-private setting and we can evaluate how DP-SGD training at different strengths of privacy guarantee impacts utility. We generate $1,000$ individual Erdős-Rényi graphs, equally distributed to two classes. Each graph consists of twenty nodes which contain nine features each. The node features are sampled from a normal distribution with different mean values and the same standard deviation, corresponding to the label class of the graph. The edge connection probabilities vary slightly between the two classes.

Fingerprints Dataset

Fingerprint classification aims to separate images of fingerprints into the different classes - arch, left, right, and whorl - from the Galton-Henry classification system [41], [42]. A large within-class variability and a small separation between classes makes fingerprint classification a challenging task [43]. We rely on the dataset introduced by Riesen et al. [44] and provided by TU Datasets [45] to perform differentially private graph classification on fingerprints. The graphs are extracted from the images based on directional variance and the task follows the Galton-Henry classification scheme of five classes. We merge the five classes into four classes following the approach described in [44]. Differentially private ML naturally befits this task, as it allows one to privatise the utilisation of the uniquely identifying fingerprint data for e.g. training machine learning models in tasks such as automated authentication.

Molbace Dataset

To perform molecule classification in a binary graph classification setting, we use the benchmark dataset *Molbace* from the OGB database [46], where the *Molbace* dataset is adapted from MoleculeNet [47]. It consists of $1,513$ graphs, where each graph represents a molecule. Edges represent molecular bonds and nodes correspond to the atoms in the molecule. Each node contains 9 node features and the average number of nodes per graph is 34. We split the dataset into $1,210$ training graphs, 152 test graphs and 151 validation graphs. Node features contain atom features; for example the atomic number, chirality, formal charge, or whether the atom is in a ring or not. The prediction task of this dataset is to correctly classify whether the molecule inhibits HIV virus replication [46]. Such a task is representative of federated learning workflows with per-site (local) DP application, in which e.g. several pharmaceutical companies wish to jointly train a model for molecule property prediction, while wishing to limit the disclosure of their (possibly proprietary) molecule structures from third parties.

ECG Dataset

For the task of electrocardiogram (ECG) classification, we use the publicly available ECG dataset from the China Physiological Signal Challenge (CPSC) 2018 challenge dataset [48]. We formulate a classification task between ECGs showing signs of a Left Bundle Branch Block and normal ECGs showing a sinus rhythm. The ECG data consists of twelve ECG signal channels (*leads*), recorded at different locations on the human torso and extremities. Leads affixed to the extremities constitute signal channels I, II, II, aVR, aVF and aVL. Leads affixed to the chest are used to derive signal channels V1

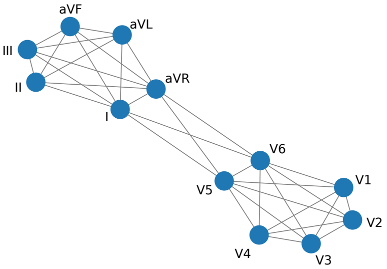


Fig. 2: Graph visualisation of ECG data. We connected the different signal channels based on the medical location of the leads as well as prior knowledge. Leads I, II, III, aVF, aVL, and aVR are located on the extremities and the remaining leads on the chest.

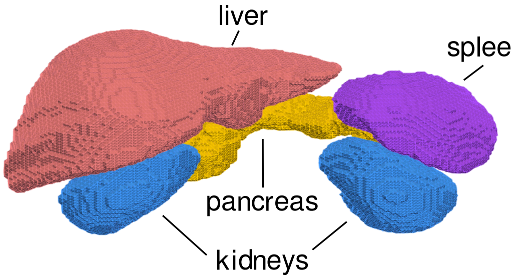


Fig. 3: Organ meshes extracted from segmenations of UK Biobank data [40]. The organs shown in this figure are the liver (coral), the spleen (purple), the left and right kidneys (blue) and the pancreas (yellow).

through V6. To construct a graph dataset from the ECG data, we utilise this medical motivation and divide the ECG extremity signal channels from the chest signal channels by fully connecting the extremity and chest subgraphs. In addition, we utilise prior knowledge about the leads which are typically used by physicians to delineate LBBB from sinus rhythm and thus connected channels I, aVR, V5, and V6. The structure of those graphs is visualised in Figure 2. The dataset we use contains ECG data of $1,125$ subjects. As ECG signals are periodic, we sub-sample the signals by only retaining the first 512 signal points of each channel, leading to 512 node features in the graphs. The binary classification dataset is highly imbalanced with 207 subjects showing signs of LBBB and 918 having normal ECG curves. Evidently, ECG data, like all medical data is highly sensitive, and thus requires formal methods of privacy protection.

Organ Meshes Dataset

To investigate the scalability of our method to large sensitive medical datasets, we perform an organ mesh classification task on $151,910$ organ surface meshes extracted from $30,382$ subjects from the UK Biobank database [40]. As a first step, the five organs liver, spleen, left and right kidney, and pancreas were segmented using the segmentation pipeline of [49]. Secondly, the organ meshes were extracted from those segmentations using the *marching cubes* algorithm [50] implementation by [51]. Figure 3 shows an example visualisation of the surface meshes of one subject. Each organ is represented as an individual graph in the dataset and the task is to classify which of the five organs is represented by the surface mesh. Node features contain the three dimensional coordinates of the organs with respect to the original magnetic resonance imaging (MRI) scan of the subject.

4.2 GNN Models for Graph Classification and DP-SGD Training

Since the adoption of deep learning techniques to graph learning, most state-of-the-art methods for graph classification rely on a variant of *message passing* to aggregate information across the nodes [52], [53], [54], [55], [56].

For our experiments, we implement a variety of GNN models to compare performance and evaluate the impact of DP on different graph learning techniques. We use GraphSAGE [57], Graph Attention Networks (GATs) [58], Graph Convolutional Networks (GCNs) [28], and chebyshev spectral graph convolutions (Cheb) [59]. For each dataset, we perform hyperparameter searches, leading to different models for each application. The depth of the GNNs varies from two to three layers with/without Instance Normalisation layers and with/without dropout, depending on the problem space. We do not use Batch Normalisation because of its incompatibility with differentially private training; Batch Normalisation, by taking averages across the batch during the forward pass, leaks information over samples in a batch and precludes the computation of *per-sample* gradients necessary for DP-SGD. More details about the model architectures can be found in the supplementary material.

When training graph classification models with DP-SGD, we follow the standard procedure of DP-SGD training. Firstly, a privacy budget is set in terms of ε, then the model is trained with a specific noise multiplier that defines the amount of Gaussian noise added to the gradients of the model and a L_2-sensitivity bound. The model can then be trained a certain number of iterations, until the privacy budget ε is reached. We then report the scores of the best-performing model out of the ones trained before the privacy budget is exhausted. For all differentially private training runs, we set $\delta = \frac{1}{N}$, where N is the cardinality of the dataset and monitor the performance of the algorithm with different privacy budgets ε. Across all experiments, we utilise the same model architectures for DP-SGD and SGD training with the removal of potential dropout layers for DP-SGD training. In Table 2 we report the mean performance as well as the standard deviation of five independent runs for each experiment. We evaluate different scores for each model: ROC AUC, Accuracy, Sensitivity, Specificity and F1 Score. Hereby sensitivity reports the rate between the true positives and the sum of the true positives and false negatives. Specificity is the rate between the true negatives and the sum of the true negatives and false positives. The ROC AUC score is the Compute Area Under the Receiver Operating Characteristic Curve with a *micro* average for multi-class datasets. Accuracy is the rate between the true positives and all samples and the F1 Score reports the harmonic mean of the precision and recall, also using a *micro* averaging strategy for multi-class datasets.

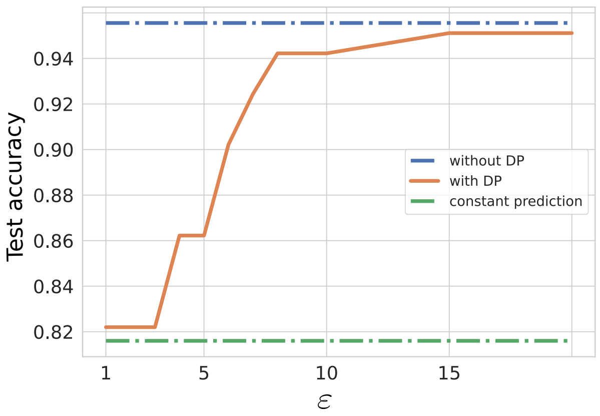


Fig. 4: Impact of ε on test accuracy on ECG dataset. The performance increases with larger ε values and looser privacy guarantees. The top dashed line (blue) indicates the performance without DP, the lower dashed line (green) a constant prediction and the solid line in the middle (orange) the model performance with different ε values: $\varepsilon \in \{1, 2, \ldots, 10, 15, 20\}$.

5 Experimental Results

In this section, we evaluate our results, compare DP-SGD training with standard SGD training and show the impact of different privacy budgets on model performances. The results achieved on the four datasets are summarised in Table 2.

Summary of Results

For all datasets, we observe similar behaviour, namely a correlation between stronger privacy budgets and diminished model performance. Although this phenomenon is – in general – an unavoidable, information-theoretic consequence of the trade-off between privacy and utility, the individual models exhibit different behaviour with regards to their individual tolerances towards the amount of Gaussian noise added for DP-SGD, as well as the tolerances towards gradient clipping. For instance, for the *synthetic* dataset, an ε value of 5 does not lead to accuracy loss, whereas for the *Molbace* dataset, a privacy budget of $\varepsilon = 10$ already results in diminished model accuracy. Interestingly, the performance of DP-SGD training is overall not substantially influenced by the choice of GNN architecture (GCN, GAT, GraphSAGE, or ChebNet). We observe high performance and similar convergence rates for all architectures, indicating the robust performance of DP-SGD training. For a comparison of the training behaviours please see our Figure in the supplementary material.

For all models, we observe an increased inter-run variability with stronger privacy guarantees. This behaviour is reflected in the higher standard deviations reported in Table 2, and we attribute this phenomenon to the increased randomness injected by the DP mechanism.

Exemplarily, we visualise the impact of a stronger privacy guarantee on the performance on the *ECG* dataset in Figure 4. Given that the dataset is highly imbalanced, a constant prediction (marked by the lower dashed green line in Figure 4) would result in an approximate test accuracy of 81.6%. We examine the dependency of the results on the choice of ε and report the different performances. With a very strong privacy guarantee (corresponding to a low ε value), the performance of the network is barely better than a constant prediction. The looser the privacy guarantee (larger ε value) the better the performance; for a very loose ε the results reach non-DP performance. Interestingly, for some models we observe identical performance between DP-SGD and normal training, e.g. *Fingerprint-GCN*, where the DP-SGD model (privacy budget of $\varepsilon = 5$) reaches slightly higher performance then the normal training, see Table 2; this beneficial effect can be attributed to the regularising effects of gradient norm bounding and noise injection, indicating that – within certain constraints – DP training can go hand-in-hand with excellent overall model performance and generalisability.

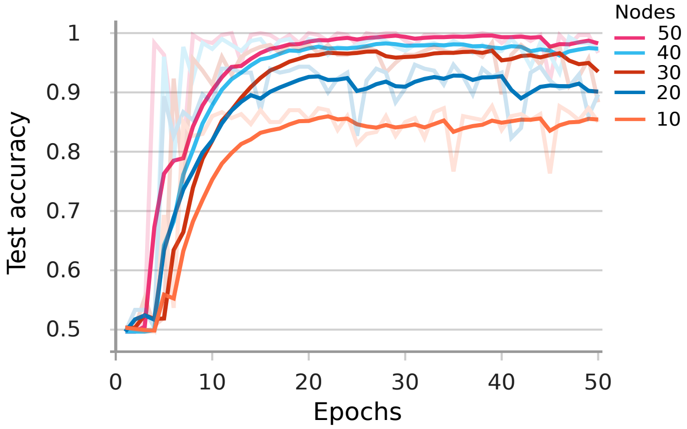


Fig. 5: Impact of graph size to performance under DP: Increasing graph sizes result in better performance and faster convergence. The privacy guarantees are set to $\varepsilon = 2.3$.

Scalability

In order to investigate the scalability of our approach, we vary the size of the created Erdős-Rényi graphs in the *synthetic* dataset between 10 and 500 nodes per graph. Figure 5 shows the impact of the graph size on the performance under DP using a three-layer GCN and $\varepsilon = 2.3$. We visualised the performances of graph sizes between 10 and 50 nodes and find that performance improves with increasing graph size in these ranges. Beyond 50 nodes, the performance remains consistently high, which is why these plots were not included in Figure 5. This behaviour indicates a strong performance of our model across varying graph sizes, i.e. robust scalability. Furthermore, with the utilisation of the large organ mesh dataset, we could show that our method also performs excellently for graphs with a large number of nodes and edges as well as large datasets with more than 100, 000 graphs. In this dataset, we observe low utility loss in the range of 10^{-3} even in a very high privacy regime of $\varepsilon = 0.5$. In comparison, many deep learning networks require a more loose privacy guarantee to achieve high performance [60].

Data	Network	Training	ROC-AUC	Accuracy	Sensitivity	Specificity	F1-Score	Noise	L_2-Clip	ε
Synthetic	GCN	SGD	0.934 ± 0.01	0.934 ± 0.01	0.955 ± 0.03	0.913 ± 0.03	0.934 ± 0.01	-	-	-
		DP-SGD	0.918 ± 0.02	0.918 ± 0.02	0.897 ± 0.03	0.940 ± 0.02	0.917 ± 0.02	1.0	3.0	5.0
		DP-SGD	0.907 ± 0.02	0.910 ± 0.20	0.869 ± 0.04	0.946 ± 0.20	0.907 ± 0.02	2.0	3.0	1.0
		DP-SGD	0.757 ± 0.11	0.756 ± 0.10	0.936 ± 0.06	0.575 ± 0.28	0.756 ± 0.10	2.2	3.0	0.5
	GAT	SGD	0.912 ± 0.01	0.912 ± 0.01	0.940 ± 0.03	0.883 ± 0.06	0.912 ± 0.01	-	-	-
		DP-SGD	0.893 ± 0.01	0.893 ± 0.01	0.895 ± 0.03	0.891 ± 0.03	0.893 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.872 ± 0.02	0.872 ± 0.02	0.827 ± 0.04	0.907 ± 0.07	0.872 ± 0.02	2.0	3.0	1.0
		DP-SGD	0.575 ± 0.07	0.575 ± 0.07	0.730 ± 0.35	0.419 ± 0.42	0.576 ± 0.08	2.2	3.0	0.5
	SAGE	SGD	0.903 ± 0.02	0.902 ± 0.02	0.913 ± 0.04	0.893 ± 0.08	0.903 ± 0.02	-	-	-
		DP-SGD	0.918 ± 0.01	0.918 ± 0.01	0.907 ± 0.03	0.933 ± 0.02	0.918 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.893 ± 0.01	0.892 ± 0.01	0.872 ± 0.04	0.914 ± 0.03	0.893 ± 0.01	2.0	3.0	1.0
		DP-SGD	0.598 ± 0.10	0.598 ± 0.11	0.609 ± 0.47	0.587 ± 0.39	0.598 ± 0.10	2.2	3.0	0.5
Fingerprints	GCN	SGD	0.856 ± 0.01	0.785 ± 0.01	0.785 ± 0.01	0.928 ± 0.00	0.785 ± 0.01	-	-	-
		DP-SGD	0.863 ± 0.01	0.794 ± 0.01	0.771 ± 0.01	0.932 ± 0.00	0.794 ± 0.01	1.0	3.0	5.0
		DP-SGD	0.844 ± 0.02	0.766 ± 0.04	0.733 ± 0.05	0.921 ± 0.01	0.766 ± 0.04	1.8	3.0	1.0
		DP-SGD	0.796 ± 0.06	0.693 ± 0.10	0.658 ± 0.09	0.898 ± 0.03	0.693 ± 0.09	2.3	3.0	0.5
	GAT	SGD	0.857 ± 0.01	0.786 ± 0.01	0.764 ± 0.02	0.929 ± 0.01	0.786 ± 0.01	-	-	-
		DP-SGD	0.849 ± 0.02	0.774 ± 0.03	0.733 ± 0.04	0.924 ± 0.01	0.770 ± 0.03	1.0	3.0	5.0
		DP-SGD	0.812 ± 0.02	0.728 ± 0.03	0.661 ± 0.01	0.906 ± 0.01	0.730 ± 0.03	1.8	3.0	1.0
		DP-SGD	0.737 ± 0.05	0.605 ± 0.08	0.585 ± 0.08	0.871 ± 0.03	0.610 ± 0.08	2.3	3.0	0.5
	SAGE	SGD	0.876 ± 0.02	0.814 ± 0.02	0.802 ± 0.03	0.940 ± 0.01	0.814 ± 0.02	-	-	-
		DP-SGD	0.869 ± 0.01	0.804 ± 0.01	0.788 ± 0.02	0.935 ± 0.01	0.804 ± 0.01	1.0	3.0	5
		DP-SGD	0.861 ± 0.01	0.792 ± 0.01	0.776 ± 0.01	0.932 ± 0.00	0.791 ± 0.01	1.8	3.0	1
		DP-SGD	0.712 ± 0.06	0.568 ± 0.08	0.529 ± 0.09	0.853 ± 0.03	0.568 ± 0.08	2.3	3.0	0.5
ECG	GCN	SGD	0.979 ± 0.01	0.932 ± 0.01	0.744 ± 0.03	0.979 ± 0.01	0.845 ± 0.02	-	-	-
		DP-SGD	0.983 ± 0.01	0.904 ± 0.02	0.581 ± 0.07	0.983 ± 0.01	0.727 ± 0.06	0.6	5.0	10
		DP-SGD	0.983 ± 0.01	0.923 ± 0.01	0.644 ± 0.12	0.983 ± 0.01	0.772 ± 0.09	0.8	5.0	5.0
		DP-SGD	0.986 ± 0.02	0.824 ± 0.03	0.169 ± 0.23	0.986 ± 0.02	0.231 ± 0.28	1.5	5.0	1.0
	GAT	SGD	0.983 ± 0.01	0.922 ± 0.04	0.675 ± 0.19	0.983 ± 0.01	0.781 ± 0.17	-	-	-
		DP-SGD	0.968 ± 0.03	0.899 ± 0.01	0.637 ± 0.11	0.968 ± 0.03	0.762 ± 0.11	0.6	5.0	10
		DP-SGD	0.960 ± 0.01	0.909 ± 0.02	0.712 ± 0.12	0.960 ± 0.01	0.811 ± 0.08	0.8	5.0	5.0
		DP-SGD	0.991 ± 0.01	0.846 ± 0.01	0.200 ± 0.11	0.991 ± 0.01	0.319 ± 0.11	1.5	5.0	1.0
	SAGE	SGD	0.985 ± 0.01	0.946 ± 0.01	0.757 ± 0.04	0.985 ± 0.01	0.856 ± 0.02	-	-	-
		DP-SGD	0.972 ± 0.01	0.932 ± 0.02	0.767 ± 0.09	0.972 ± 0.01	0.854 ± 0.06	0.6	5.0	10
		DP-SGD	0.973 ± 0.02	0.928 ± 0.02	0.738 ± 0.09	0.973 ± 0.02	0.835 ± 0.06	0.8	5.0	5.0
		DP-SGD	0.951 ± 0.07	0.841 ± 0.02	0.402 ± 0.30	0.951 ± 0.07	0.493 ± 0.24	1.5	5.0	1.0
Molbace	GCN	SGD	0.743 ± 0.00	0.655 ± 0.02	0.511 ± 0.03	0.820 ± 0.01	0.629 ± 0.02	-	-	-
		DP-SGD	0.699 ± 0.01	0.670 ± 0.01	0.723 ± 0.02	0.608 ± 0.01	0.660 ± 0.01	0.5	5.0	20
		DP-SGD	0.688 ± 0.01	0.609 ± 0.01	0.412 ± 0.01	0.834 ± 0.01	0.552 ± 0.01	0.6	5.0	10
	GAT	SGD	0.781 ± 0.01	0.726 ± 0.02	0.691 ± 0.07	0.766 ± 0.06	0.721 ± 0.02	-	-	-
		DP-SGD	0.747 ± 0.02	0.580 ± 0.02	0.333 ± 0.07	0.862 ± 0.03	0.475 ± 0.07	0.5	5.0	20
		DP-SGD	0.692 ± 0.03	0.518 ± 0.04	0.153 ± 0.10	0.935 ± 0.04	0.248 ± 0.14	0.6	5.0	10
	SAGE	SGD	0.785 ± 0.00	0.654 ± 0.01	0.484 ± 0.02	0.848 ± 0.01	0.616 ± 0.01	-	-	-
		DP-SGD	0.717 ± 0.00	0.620 ± 0.01	0.901 ± 0.00	0.299 ± 0.01	0.448 ± 0.02	0.5	5.0	20
		DP-SGD	0.701 ± 0.00	0.550 ± 0.01	0.262 ± 0.00	0.879 ± 0.01	0.403 ± 0.01	0.6	5.0	10
Organ Meshes	GCN	SGD	0.997 ± 0.00	0.988 ± 0.00	0.988 ± 0.00	0.997 ± 0.00	0.988 ± 0.00	-	-	-
		DP-SGD	0.946 ± 0.00	0.940 ± 0.00	0.940 ± 0.00	0.985 ± 0.00	0.940 ± 0.00	0.791	2.0	1.0
		DP-SGD	0.946 ± 0.00	0.934 ± 0.00	0.934 ± 0.00	0.984 ± 0.00	0.934 ± 0.00	1.07	1.5	0.5
	Cheb	SGD	0.992 ± 0.00	0.983 ± 0.00	0.983 ± 0.00	0.996 ± 0.00	0.983 ± 0.00	-	-	-
		DP-SGD	0.978 ± 0.00	0.933 ± 0.00	0.933 ± 0.00	0.983 ± 0.00	0.933 ± 0.00	0.796	2.5	1.0
		DP-SGD	0.982 ± 0.00	0.925 ± 0.00	0.924 ± 0.00	0.981 ± 0.00	0.925 ± 0.00	1.094	2.5	0.5
	SAGE	SGD	0.996 ± 0.00	0.992 ± 0.00	0.991 ± 0.00	0.998 ± 0.00	0.991 ± 0.00	-	-	-
		DP-SGD	0.981 ± 0.00	0.938 ± 0.00	0.937 ± 0.00	0.984 ± 0.00	0.938 ± 0.00	0.796	1.0	1.0
		DP-SGD	0.988 ± 0.00	0.937 ± 0.00	0.937 ± 0.00	0.984 ± 0.00	0.937 ± 0.00	1.06	2.5	0.5

TABLE 2: Summary of our experimental evaluation on four datasets: *Synthetic*, *Fingerprints*, *ECG*, *Molbace*, and *Organ Meshes* with different network types. We report results with SGD and DP-SGD training as well as varying privacy budgets ε. The scores are evaluated on the test sets with a standard deviation based on five independent runs. We find that our models achieve high performance when using our proposed DP-SGD training method. The performance decreases gradually when increasing privacy guarantees.

Explainability

The interpretability of GNNs is a challenging and frequently discussed task in research. Recently, approaches like the GNNExplainer [61] formalised methods which can be used to interpret the results of trained GNNs. We make use of this method to interpret the differences in learned representations between models trained with DP-SGD and non-private SGD and visualise the results in Figure 6. The GNNExplainer is an approach for post-hoc interpretation of predictions generated by a trained GNN. It is used to identify which edges in the graph represent essential connections for a prediction of the network, thus indicating nodes important for the final prediction. GNNExplainer prunes the original graph to only contain the nodes and edges with the highest impact on the model prediction. We apply the GNNExplainer to our results on the *Fingerprints* dataset, comparing a GCN model trained with standard SGD and three GCN models trained with DP-SGD with $\varepsilon = 5$, $\varepsilon = 1$ and $\varepsilon = 0.5$. We set the GNNExplainer threshold for edge importance to 0.2. Qualitatively, we observe that the GNNExplainer results of the DP models and the standard models appear very similar, if not identical for some examples, see Figures 6 and supplementary material. In these Figures, (**A**) visualises an example of an original graph from the *Fingerprints* dataset, containing all edges. Figures (**B**) and (**C**) show the pruned graphs for SGD and DP-SGD training, respectively. In the lower example (**2**) in Figure 6, both GNNExplainer graphs are identical (almost identical in the upper row), showing that in both models the same edges and nodes have a high impact on the models' predictions. This indicates that the feature importance is the same (or almost the same) between both models and that the feature importance is not compromised by the privacy guarantees achieved through DP training.

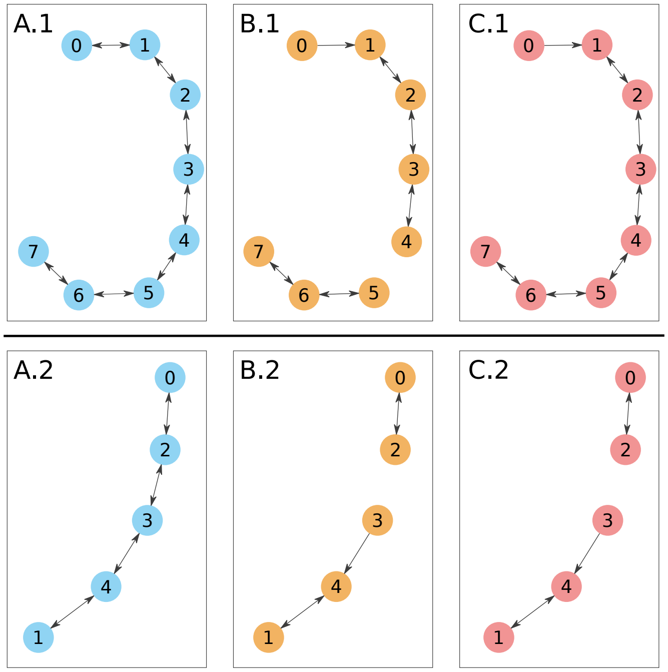


Fig. 6: Visualisation of two GNNExplainer examples. The original graph (**A**) is shown in blue, the resulting graph from the GNNExplainer and the model trained with SGD in orange (**B**) and with DP-SGD in red (**C**). In the example in the upper row (**1**) the two graphs (**B**) and (**C**) differ slightly, whereas in the lower example (**2**) both GNNExplainer graphs (**B**) and (**C**) are equal, meaning that the two models consider the same edges to be relevant. The privacy budget for the models trained with DP-SGD was set to ε=5.

ε	ROC-AUC	IoU (orig. $\|$ DP)	IoU (DP $\|$ non-DP)
5.0	0.863	0.652	**0.765**
1.0	0.844	0.592	**0.736**
0.5	0.796	0.617	**0.609**

TABLE 3: Mean IoU scores of ten test samples from the *Fingerprint* dataset for comparing edges between ($\|$) the original graph, the GNNExplainer graph of the model trained with SGD, and the GNNExplainer graph of the model trained with DP-SGD. The IoU between the original graph and the non-DP graph is 0.739. The IoU between the DP and the non-DP graphs decreases with a smaller ε value which corresponds to smaller ROC AUC results.

To provide a quantitative estimation of GNNExplainer similarity of our results, we propose and use an Intersection over Union (IoU) score, measuring the pair-wise overlap of edges in the three resulting graphs. The IoU score of two graphs A and B is defined as follows:

$$\frac{|\mathcal{E}_A \cap \mathcal{E}_B|}{|\mathcal{E}_A \cup \mathcal{E}_B|}, \tag{4}$$

where $\mathcal{E}_X$ represents the set of all edges in Graph X and $|\cdot|$ denotes the cardinality of a set. Table 3 summarises the results of the mean IoU values between the original graph and the GNNExplainer graph based on training with DP, and the two resulting GNNExplainer graphs from DP-SGD and SGD training. The IoU score of the original graph and the GNNExplainer graph of the model trained with standard SGD is 0.739 for all graphs. We compare the overlap between the graphs with the model performance, reported by the ROC AUC score. We find a high IoU score for DP vs. non-DP models, which is in line with the GNNExplainer plots we observe in Figure 6. Moreover, we observe that our GNNExplainer IoU score of the DP and the non-DP models slightly decreases with a smaller ε and smaller ROC AUC scores, see Table 3. The increase in the IoU score between the original model and the DP model with $\varepsilon = 0.5$ most likely only indicates that the DP trained model with $\varepsilon = 0.5$ considers more edges as relevant than the model trained with $\varepsilon = 1.0$. These qualitative and quantitative GNNExplainer results indicate that our proposed DP graph classification models exhibit strong and similar inductive biases compared to "normal" GNNs while preserving privacy guarantees.

6 Discussion, Conclusion, and Future Work

Our work introduces and evaluates differentially private graph classification, a formal method to offer quantifiable privacy guarantees in applications where sensitive data of individuals is represented as a whole graph. Such contexts include medical data (as shown in our ECG classification example), where DP can enable training of machine learning models while maintaining both regulatory compliance and adherence to ethical standards mandating the protection of health-related data.

GNN training is possible with strong privacy guarantees and excellent utility

Our experiments on benchmark and real-world datasets demonstrate that the training of GNNs for graph classification is viable with high utility and tight privacy guarantees. Especially the large scale mesh classification dataset achieved almost perfect accuracy even with very tight privacy bounds of $\varepsilon = 0.5$. Expectedly, we observe a privacy-performance trade-off for all datasets, whereby a decrease in the value of ε results in a decline in the accuracy of the model, as demonstrated in Figure 4. The amount of performance loss is task and dataset dependent.

GNNs learn similar features in the private and non-private scenarios

Additionally, we investigate the utilisation of explainability techniques to compare the representations learned by models trained with SGD and DP-SGD. The application of the GNNExplainer indicates that models trained with DP-SGD learn similar relevant representations to the non-privately trained models. To quantitatively demonstrate the results of the GNNExplainer, we calculated an IoU score on the edges considered important by the technique between the resulting graphs. We observe an overall high IoU with a slight decline in overlap with tighter privacy guarantees, indicating that – as expected – the high levels of noise required to achieve such guarantees eventually become detrimental to learning.

Private GNN training can help alleviate social impacts of machine learning

We strongly believe that the implementation of formal techniques for privacy preservation like DP in the setting of GNN training will mitigate the risks of using sensitive data in ML tasks. In the case of medical data (as in the *ECG* dataset example), we believe the utilisation of privacy preserving methods to also hold positive effects in terms of encouraging data owners (such as patients) to make their data accessible for research purposes. Evidently, such implementations must go hand in hand with educating potential stakeholders in the correct application of DP mechanisms, including the appropriate choice of parameters like ε. In this work, we rely exclusively on public datasets collected with informed consent or with approval of institutional review boards wherever applicable.

Limitations

Inherent to the concept of differential privacy in machine learning is a performance-to-privacy trade-off. While our experiments visually illustrate the implications of the trade-off and provide insight into its practical importance in the context of machine learning on graphs, the actual relationship between privacy and accuracy is highly task- and user-specific [62], [63]. Therefore, we note that one can interpret the value of ε as an additional design-parameter that needs to be optimised for in order to minimise the adverse effects that DP can have on performance in the context of graph classification (or most other learning tasks in general).

Future work

In our experiments we utilise a limited set of standard model architectures (GCN, GraphSAGE, GAT, ChebNet). Evidently, more sophisticated architectures have been designed and deployed to real world problems. As our proposed approach is general, we assume that an extension to such advanced graph learning models is natural and should exhibit similar behaviour, and we intend to expand our purview to such models in future investigations.

While the GNNExplainer concept can provide initial clues to interpret and explain GNN training and the intrinsic differences between models trained with SGD and DP-SGD, it is only an initial step towards full explainability and interpretability. We consider this to be a highly relevant and an interesting direction for future research. In particular, we aim to investigate the effects of differentially private GNN learning on adversarial robustness of the model. We hypothesise that – similarly to Euclidean settings – [64], [65] DP should have a mitigating effect against attacks that diminish the utility of the trained model in the context of machine learning on graphs. Furthermore, we believe that a comparison of different explainability techniques like [66], [67], [68], [69] will provide even more insight into the differences between DP and non-DP training, which we also intend to investigate in future work.

References

[1] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[2] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019.

[4] Xiaoxiao Li, Nicha C Dvornek, Yuan Zhou, Juntang Zhuang, Pamela Ventola, and James S Duncan. Graph neural network for interpreting task-fmri biomarkers. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 485–493. Springer, 2019.

[5] Chengsheng Mao, Liang Yao, and Yuan Luo. Medgcn: Graph convolutional networks for multiple medical tasks. *arXiv preprint arXiv:1904.00326*, 2019.

[6] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.

[7] Johannes C Paetzold, Julian McGinnis, Suprosanna Shit, Ivan Ezhov, Paul Büschl, Chinmay Prabhakar, Anjany Sekuboyina, Mihail Todorov, Georgios Kaissis, Ali Ertürk, et al. Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[8] Xin Wei, Ruixuan Yu, and Jian Sun. View-gcn: View-based graph convolutional network for 3d shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1850–1859, 2020.

[9] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

[10] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 619–633, Toronto Canada, January 2018. ACM.

[11] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[12] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. *arXiv preprint arXiv:1911.07135*, 2019.

[13] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting Gradients–How easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020.

[14] Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chengqiang Lu, Chuanren Liu, and Enhong Chen. Graphmi: Extracting private graph data from graph neural networks. *arXiv preprint arXiv:2106.02820*, 2021.

[15] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. Node-level membership inference attacks against graph neural networks. *arXiv preprint arXiv:2102.05429*, 2021.

[16] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla. Membership inference attack on graph neural networks. *arXiv preprint arXiv:2101.06570*, 2021.

[17] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[18] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1):1–8, 2021.

[19] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021.

[20] Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, Zumrut Muftuoglu, Sahib Singh, and Fatemehsadat Mireshghallah. Benchmarking differential privacy and federated learning for bert models. *arXiv preprint arXiv:2106.13973*, 2021.

[21] Hankz Hankui Zhuo, Wenfeng Feng, Yufeng Lin, Qian Xu, and Qiang Yang. Federated deep reinforcement learning. *arXiv preprint arXiv:1901.08277*, 2019.

[22] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 151–164. Springer, 2019.

[23] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[24] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[25] Pennsylvania State University Adam Smith. Differentially private analysis on graphs, 2016.

[26] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography Conference*, pages 457–476. Springer, 2013.

[27] Timour Igamberdiev and Ivan Habernal. Privacy-preserving graph convolutional networks for text classification. *arXiv preprint arXiv:2102.09604*, 2021.

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[29] Ameya Daigavane, Gagan Madan, Aditya Sinha, Abhradeep Guha Thakurta, Gaurav Aggarwal, and Prateek Jain. Node-level differentially private graph neural networks. *arXiv preprint arXiv:2111.15521*, 2021.

[30] Jun Zhou, Chaochao Chen, Longfei Zheng, Huiwen Wu, Jia Wu, Xiaolin Zheng, Bingzhe Wu, Ziqi Liu, and Li Wang. Vertically federated graph neural network for privacy-preserving node classification. *arXiv preprint arXiv:2005.11903*, 2020.

[31] Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2130–2145, 2021.

[32] Peiyuan Liao, Han Zhao, Keyulu Xu, Tommi Jaakkola, Geoffrey J Gordon, Stefanie Jegelka, and Ruslan Salakhutdinov. Information obfuscation of graph neural networks. In *International Conference on Machine Learning*, pages 6600–6610. PMLR, 2021.

[33] Xiao Han, Leye Wang, Junjie Wu, and Yuncong Yang. Large-scale privacy-preserving network embedding against private link inference attacks. *arXiv preprint arXiv:2205.14440*, 2022.

[34] Hui Hu, Lu Cheng, Jayden Parker Vap, and Mike Borowczak. Learning privacy-preserving graph convolutional network with partially observed sensitive attributes. In *Proceedings of the ACM Web Conference 2022*, pages 3552–3561, 2022.

[35] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[36] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[37] Sofya Raskhodnikova and Adam Smith. Differentially private analysis of graphs. *Encyclopedia of Algorithms*, 2016.

[38] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 4(11):1146–1157, 2011.

[39] Tamara T Mueller, Dmitrii Usynin, Johannes C Paetzold, Daniel Rueckert, and Georgios Kaissis. Sok: Differential privacy on graph-structured data. *arXiv preprint arXiv:2203.09205*, 2022.

[40] Steffen E Petersen, Paul M Matthews, Fabian Bamberg, David A Bluemke, Jane M Francis, Matthias G Friedrich, Paul Leeson, Eike Nagel, Sven Plein, Frank E Rademakers, et al. Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of uk biobank-rationale, challenges and approaches. *Journal of Cardiovascular Magnetic Resonance*, 15(1):1–10, 2013.

[41] F Galton. Finger prints macmillan. *London. 246p*, 1892.

[42] ER Henry. Classification and uses of fingerprints london. *George Rutledge and Sons, Limited*, 54, 1900.

[43] Michel Neuhaus and Horst Bunke. A graph matching based approach to fingerprint classification using directional variance. In *International Conference on Audio-and Video-Based Biometric Person Authentication*, pages 191–200. Springer, 2005.

[44] Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 287–297. Springer, 2008.

[45] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[46] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs, 2021.

[47] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[48] Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, et al. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7):1368–1373, 2018.

[49] Turkay Kart, Marc Fischer, Thomas Küstner, Tobias Hepp, Fabian Bamberg, Stefan Winzeck, Ben Glocker, Daniel Rueckert, and Sergios Gatidis. Deep learning-based automated abdominal organ segmentation in the uk biobank and german national cohort magnetic resonance imaging studies. *Investigative Radiology*, 56(6):401–408, 2021.

[50] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.

[51] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[52] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[53] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 257–266, 2019.

[54] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Flag: Adversarial data augmentation for graph neural networks, 2020.

[55] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin R Benson. Combining label propagation and simple models outperforms graph neural networks. *arXiv preprint arXiv:2010.13993*, 2020.

[56] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

[57] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[58] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[59] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[60] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

[61] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. *arXiv preprint arXiv:1903.03894*, 2019.

[62] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.

[63] Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. "I need a better description": An Investigation Into User Expectations For Differential Privacy. *arXiv preprint arXiv:2110.06452*, 2021.

[64] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. *arXiv preprint arXiv:2009.03561*, 2020.

[65] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[66] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

[67] Lukas Faber, Amin K Moghaddam, and Roger Wattenhofer. Contrastive graph neural network explanation. *arXiv preprint arXiv:2010.13663*, 2020.

[68] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020.

[69] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.

[70] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[71] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[72] Richard Zou Horace He. functorch: Jax-like composable function transforms for pytorch. https://github.com/pytorch/functorch, 2021.

[73] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[74] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32:9240, 2019.

Tamara T. Mueller Tamara Mueller is a Ph.D. student at the chair for AI in Medicine and Healthcare at the Technical University of Munich. Her main research interests include the application of geometric deep learning to medical imaging tasks and differential privacy of deep learning models - as well as the intersection of both. Tamara obtained her M.Phil. in advanced computer science at the University of Cambridge in 2018.

Johannes C. Paetzold Johannes C. Paetzold is a Postdoctoral researcher in computer science at Imperial College London. He is also the Artificial Intelligence Team Leader at the Institute for Tissue Engineering and Regenerative Medicine at the Helmholtz Zentrum München. His main interest is the development of deep learning and graph learning methods for large biological networks such as vessels and neurons. Further research interests include topology-aware machine learning and generative models.

Chinmay Prabhakar Chinmay Prabhakar is a Ph.D. student at the University of Zurich. His main research interest is the development of deep learning and graph learning methods on biological networks. Further research interest includes application of multi-modal learning specifically text and image data in medical domain.

Dmitrii Usynin Dmitrii is a PhD student at a Joint Academy of Doctoral Studies (JADS) launched between Imperial College London and Technical University of Munich. His research interests lie in the domain of adversarial influence in collaborative machine learning, privacy-preserving machine learning and trustworthy artificial intelligence. Dmitrii is also a privacy researcher at OpenMined, working on federated learning and differential privacy in healthcare. Dmitrii graduated from Imperial College London with an MEng in Computing and a distinguished project titled "Privacy-Preserving Machine Learning in a Medical Domain".

Daniel Rueckert Daniel Rueckert (Fellow, 2015) is Alexander von Humboldt Professor for AI in Medicine and Healthcare at the Technical University of Munich. He is also Professor of Visual Information Processing in the Department of Computing at Imperial College London. He received a PhD from Imperial College in 1997. He has published more than 500 journal and conference articles in the area of medical image computing. He served as associate editor of IEEE Transactions on Medical Imaging and is a member of the editorial board of Medical Image Analysis. In 2014, he has been elected as a Fellow of the MICCAI society, and in 2015 he was elected as a Fellow of the Royal Academy of Engineering and of the IEEE. More recently he has been elected as Fellow of the Academy of Medical Sciences (2019) and as fellow of the American Institute for Medical and Biological Engineering (2021).

Georgios Kaissis Dr Georgios Kaissis is an adjunct assistant professor and attending radiologist at the Technical University of Munich, where he leads the Trustworthy and Privacy-Preserving Artificial Intelligence group at the Institute of Artificial Intelligence in Medicine. He also leads the Reliable Artificial Intelligence group at the Institute for Machine Learning in Biomedical Imaging at Helmholtz Zentrum Munich. He obtained his medical and doctoral degrees at LMU Munich, his Master's in Business Administration for Healthcare at FAU Nuremberg and conducted a PostDoc at the Department of Computing at Imperial College London, where he remains as an honorary research associate. His research focuses on next-generation privacy-preserving machine learning techniques and especially differential privacy and its applications to medical deep learning.

10. Differentially Private Graph Neural Networks for Medical Population Graphs and the Impact of the Graph Structure

Tamara T. Mueller*, Maulik Chevli*, Ameya Daigavane, Daniel Rueckert, and Georgios Kaissis

*shared first authorship

Conference: 21st IEEE International Symposium on Biomedical Imaging, 2024, accepted

Synopsis: We initiate an empirical investigation of differentially private graph neural networks for medical population graphs. In this context, we examine privacy-utility trade-offs at different privacy levels on both real-world and synthetic datasets and perform auditing through membership inference attacks. Our findings highlight the potential and the challenges of this specific DP application area, which comes with an additional difficulty of graph structure construction that potentially complicates graph deep learning. We find evidence that the underlying graph structure constitutes a potential factor for larger performance gaps by showing a correlation between the degree of graph homophily and the accuracy of the trained model.

Contributions of thesis author: data processing, design and execution of experiments, manuscript writing.

DIFFERENTIALLY PRIVATE GRAPH NEURAL NETWORKS FOR MEDICAL POPULATION GRAPHS AND THE IMPACT OF THE GRAPH STRUCTURE

Tamara T. Mueller$^{\star 1}$ *Maulik Chevli*$^{\star 1}$ *Ameya Daigavane*^{2}
Daniel Rueckert^{1,3} *Georgios Kaissis*^{1,4}

* equal contribution, ^{1}Technical University of Munich, ^{2}Massachusetts Institute of Technology,
^{3}Imperial College London, ^{4}Helmholtz Zentrum Munich

ABSTRACT

We initiate an empirical investigation of differentially private graph neural networks for medical population graphs. In this context, we examine privacy-utility trade-offs at different privacy levels on both real-world and synthetic datasets and perform auditing through membership inference attacks. Our findings highlight the potential and the challenges of this specific DP application area, which comes with an additional difficulty of graph structure construction that potentially complicates graph deep learning. We find evidence that the underlying graph structure constitutes a potential factor for larger performance gaps by showing a correlation between the degree of graph homophily and the accuracy of the trained model.

Index Terms— Differential Privacy, Graph Neural Networks, Medical Population Graphs

1. INTRODUCTION

Graph neural networks (GNNs) are powerful methods to apply deep learning (DL) to non-Euclidean data like graphs or manifolds [1]. A graph $G := (V, E)$ is defined as a set of nodes V and a set of edges E, connecting nodes. A neighbourhood of a node $v \in V$ contains all nodes $u \in V$ for which an edge e_{uv} from u to v exists. GNNs follow a *message passing* scheme, where node features are aggregated across neighbourhoods of n hops [2, 3].The utilisation of GNNs has shown improved performance over graph agnostic methods, even on datasets not exhibiting an intrinsic graph structure. Their application has therefore been expanded to datasets which require constructing a graph structure before learning. One example are medical population graphs [4]. Here, a cohort is represented by one graph, where nodes represent subjects and their medical data, and edges connect similar subjects. The construction of the graph's edges is an important step in this pipeline since a poor graph structure can hinder graph learning [5, 6]. This has been attributed to several different graph properties, one of them being *homophily*, which is a measure of the ratio between same-labelled and differently-labelled neighbours. High homophily indicates that the majority of nodes in all neighbourhoods share the same label as the nodes of interest.

1.1. Privacy Preserving DL with Differential Privacy

DL on medical data comes with high privacy risks. The utilised data –e.g. medical images or clinical data– is highly sensitive and it has been shown that trained DL models are prone to data leakage, where even full medical images can be reconstructed [7, 8]. Therefore, privacy-preserving methods need to be applied in order to protect the patients' data. The gold standard for training DL methods while providing formal privacy guarantees, is *differential privacy* (DP) [9]. DP operates based on a privacy budget ε, where a small ε ensures high privacy and a large ε low guarantees. Intuitively, DP guarantees that the output of an algorithm is approximately invariant to the addition/removal/replacement of one subject in the database [9]. That is, for two so-called *neighbouring* databases, that differ in one record, the output of the algorithm remains similar. Formally, a randomised algorithm $\mathcal{A}$ satisfies (ε,δ)-DP if, for all neighbouring databases $\mathcal{D}$ and $\mathcal{D}'$ and all subsets $\mathcal{S} \subseteq$ Range($\mathcal{A}$), the following symmetric statement holds:

$$\mathbb{P}\left[\mathcal{A}(\mathcal{D}) \in \mathcal{S}\right] \leq e^{\varepsilon}\mathbb{P}\left[\mathcal{A}(\mathcal{D}') \in \mathcal{S}\right] + \delta.$$

One way to ensure DP for DL models is by applying a variant of stochastic gradient descent (SGD): DP-SGD [10] during training. Here, the model's per-sample gradients are clipped (to ensure boundedness in L_2-norm) and then calibrated noise is added.

While being originally defined databases with rows and columns, the application of DP to graph neural networks for node classification tasks presents two main challenges: (1) The connection and information exchange between data points requires specific definitions of DP for graph-structured data; (2) In addition, the unboundedness of neighbourhoods in a graph makes privacy amplification by sub-sampling –meaning that a DP algorithm, which is executed on random subsamples of a population provides higher privacy guarantees than when executed on the whole population– non-trivial. In tabular datasets, individual data points can be treated separately. This is not the case in graph learning settings, where nodes are connected and share information [11]. This contradicts the principle of *per-sample gradients* in DP methods and thus requires specialised notions of DP on graphs [11].

In this work, we focus on *node-level DP*, which protects the sensitive information stored in the node features of population graphs as well as the connections between neighbouring nodes. For that, we define two datasets D and D' to be neighbouring if they differ in one node and all its adjacent edges.

DP with sufficiently strong guarantees naturally protects against membership inference attacks (MIAs), which aim to infer whether a certain individual was part of the training set or not. There are a few works investigating MIAs on GNNs [12, 13]. These and works on other privacy attacks on GNNs such as *link stealing* attacks [14] and *inference attacks* [15] highlight the vulnerability of GNNs compared to non-graph machine learning methods. In this work, we extend the state-of-the-art MIA technique by [16] to GNNs for the purpose of empirically validating the privacy guarantees.

1.2. Node-level DP for GNNs

So far, only few works have investigated DP training of GNNs. Daigavane et al. [17] introduced a privacy amplification by sub-sampling technique for multi-layer GNN training with DP-SGD. To enable a sensitivity analysis for DP-SGD in multi-layer GNNs, the authors apply a graph neighbourhood sampling scheme. Here, the number of k-hop neighbours is bounded to a maximum node degree. This ensures that the learned feature embeddings throughout training are influenced by at most a bounded number of nodes. Furthermore, the standard privacy amplification by sub-sampling technique for DP-SGD is extended, such that a gradient can depend on multiple subjects in the dataset: First, a local k-hop neighbourhood of each node with a bounded number of neighbours is sampled. Next, a subset $\mathcal{B}_t$ of n sub-graphs is chosen uniformly at random from the set of sub-graphs that constitute the training set. On these sub-samples, standard DP-SGD is applied by clipping the gradients, adding noise, and using the noisy gradients for the update steps. The noise is hereby calibrated to the sensitivity with respect to any individual node, which has been bounded via sub-sampling of the input graph. The authors of [17] show generally good performance at various privacy levels, motivating us to adopt their technique for this work. Chien et al. [18] introduce a method for DP GNNs under node-level DP for highly heterophilic graphs (e.g. homophily of 0.02), which does not apply to population graph settings.

1.3. Contributions

In this work, we investigate privacy-utility trade-offs of DP GNN training on medical population graphs. Our contributions are as follows: (1) To the best of our knowledge, our work demonstrates the first successful application of DP to GNNs in medical population graphs, (2) we empirically investigate the success of membership inference attacks (MIAs) at different levels of privacy protection and (3) analyse the interplay between graph structure and model performance, highlighting homophily as a key factor influencing model utility.

2. EXPERIMENTS AND RESULTS

All experiments are performed based on the node-level DP GNN implementation of [17], using graph convolutional networks (GCNs) [2] and the transductive learning approach. Transductive learning means that all node features and edges are included in the forward pass, but only the training labels are used for backpropagation. We note that the utilised DP GNN implementation [17] does not guarantee privacy at inference. As a baseline for comparison, we also train a multi-layer perceptron (MLP) for specific experiments below. We use three medical datasets which are frequently used in the context of population graphs [4]: The TADPOLE dataset studies Alzheimer's disease and functions as a benchmark dataset for population graphs [4, 19]. We also use an in-house COVID dataset as a realistic, noisy, and small medical dataset, with the task of predicting whether a COVID patient will require intensive care unit (ICU) treatment and the ABIDE dataset from the autism brain imaging data exchange [20], where we perform a binary classification task. The ABIDE dataset is highly challenging and therefore lends itself to investigating the impact of the graph structure on our experiments. We therefore report results on the ABIDE dataset for two different graph structures, constructed using either 5 or 30 neighbours. Furthermore, we evaluate our experiments on a synthetically generated binary classification dataset to investigate the impact of different graph structures on the performance of DP population graphs under controlled conditions. All graph structures are generated using a k-nearest neighbours approach. Here, k is a hyperparameter, which specifies how many neighbours each node has, and the k most similar nodes are connected. We note that this graph construction is not private, which aligns with DP graph learning on other datasets, where an original graph structure is provided. Given that we utilise node-level DP guarantees, this graph structure is then protected during training. Details about the datasets as well as all δ values used for DP-SGD training are summarised in Table 1.

Table 1: Homophily, node count, and δ values for all datasets.

Dataset	Nr. of nodes	Homophily	δ
TADPOLE	1 277	0.7392	$1.31 \cdot 10^{-4}$
COVID	65	0.7569	$2.78 \cdot 10^{-3}$
ABIDE (k=5)	871	0.6009	$1.92 \cdot 10^{-4}$
Synthetic	1 000	varying	$1.79 \cdot 10^{-4}$

2.1. DP Training of GNNs on Population Graphs

We summarise the results of non-DP and DP training at different privacy budgets in Table 2. As expected, a higher privacy guarantee results in lower model performance. For the TADPOLE dataset, a DP guarantee of $\varepsilon = 20$ achieves performance comparable to non-DP training and even at $\varepsilon = 10$, performance is only about two percent lower than non-DP results.

Table 2: Test set accuracy (%) of non-DP and DP models at different ε-values across five random seeds.

Dataset	**Non-DP**	**Sub-graphing**	**DP** ($\varepsilon = 20$)	**DP** ($\varepsilon = 15$)	**DP** ($\varepsilon = 10$)	**DP** ($\varepsilon = 5$)
TADPOLE	72.73 ± 1.39	**76.09 ± 1.73**	72.42 ± 0.94	71.02 ± 1.22	70.39 ± 0.43	69.45 ± 1.82
COVID	72.31 ± 11.51	**73.85 ± 3.77**	69.23 ± 8.43	69.23 ± 12.87	66.15 ± 10.43	56.92 ± 12.87
ABIDE (k=5)	58.86 ± 0.81	**65.14 ± 2.37**	57.83 ± 2.02	55.54 ± 2.62	53.71 ± 2.73	54.17 ± 2.97
ABIDE (k=30)	**68.51 ± 2.75**	65.83 ± 3.57	53.49 ± 4.27	53.37 ± 1.47	51.89 ± 4.40	51.43 ± 3.47

We attribute this to the informative underlying graph structure of the TADPOLE dataset, which stabilises graph learning. For the ABIDE dataset, non-DP performance is better for the graph structure that uses 30 neighbours (k=30) compared to only 5 neighbours. However, larger neighbourhoods lead to more noise being added during DP-SGD training, which impacts the privacy-utility trade-off on this graph. For all datasets apart from the ABIDE dataset with 30 neighbours (k=30), the model trained without DP, but employing sub-graph sampling ("sub-graphing") and gradient clipping out-performs the non-DP model trained without these techniques. We attribute this to the regularising effect of both aforementioned methods. As seen, the ABIDE dataset has overall much lower accuracy and simultaneously, the homophily of the ABIDE dataset is the lowest among all datasets. We therefore further investigate the impact of the homophily of the graph structure on the performance of DP population graphs in Section 2.3.

2.2. Membership Inference Attacks

The dependencies between graph elements render GNNs more vulnerable to MIA. Moreover, in the transductive setting of graph learning, test node features are included in the forward pass, which facilitates MIA [12]. To empirically audit the privacy leakage of sensitive patient data from our GNN models, we employ the MIA implementation of Carlini et al. [16]. We perform these experiments on the GNNs trained on the TADPOLE dataset, as it is known that higher model accuracy improves MIA success [16]. The adversary/auditor in this membership inference scenario has full access to the trained model f_θ, its architecture, and the graph, including its ground-truth labels [16]. We trained 128 shadow models to estimate the models' output logit distributions and create a classifier that predicts whether a specific example was used as training data for the model f_θ. In Figure 1, we report the log-scale receiver operating characteristic (ROC) curve of the attacks and report the true positive rate (TPR) at three fixed, low false positive rates (FPR) $(0.1\%, 0.5\%, 1\%)$. The attack's success rates are also summarised in Table 3. Furthermore, we derive the maximum TPR (i.e. power) that is theoretically achievable for a given (ε, δ) setting through the duality between (ε, δ)-DP and hypothesis testing DP. We will refer to this maximum achievable TPR as the adversary's *supremum power* $\mathcal{P}$. As seen, for FPR-values < 0.001, the MIA is unsuccessful. As the FPR tolerance is increased, models trained with weaker privacy guarantees ($\varepsilon \in \{20, 15\}$) yield positive TPR when attacked, with TPR values approaching these of models trained without DP guarantees (*Non-DP* and *Sub-graphing* variants) in case of $\varepsilon = 20$. Interestingly, the model trained at $\varepsilon = 5$ successfully resists membership inference even at an FPR value of 0.01. Moreover, we observe that the GNN trained with clipped gradients is less vulnerable to membership inference than the GNNs trained without gradient clipping. This is in line with the findings in [16] that clipping the gradients during training offers some (empirical) protection against MIAs.

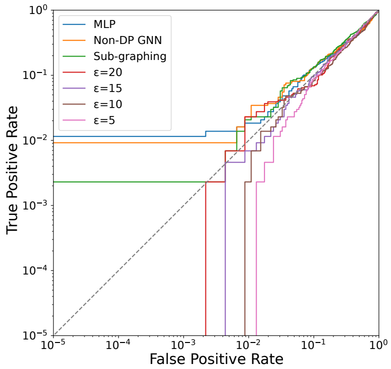


Fig. 1: Empirical ROC curves for MIA on TADPOLE.

Table 3: MIA results at different privacy budgets (TADPOLE).

Model	**Variant**	$\leq$ **0.001 FPR**		$\leq$ **0.005 FPR**		$\leq$ **0.01 FPR**	
		TPR	$\mathcal{P}$	**TPR**	$\mathcal{P}$	**TPR**	$\mathcal{P}$
MLP	-	0.0115	-	0.0138	-	0.0184	-
GNN	Non-DP	0.0092	-	0.0092	-	0.0230	-
	Sub-graphing	0.0023	-	0.0069	-	0.0207	-
	DP ($\varepsilon = 20$)	0.0000	1.0	0.0069	1.0	0.0230	1.0
	DP ($\varepsilon = 15$)	0.0000	1.0	0.0046	1.0	0.0069	1.0
	DP ($\varepsilon = 10$)	0.0000	1.0	0.0000	1.0	0.0023	1.0
	DP ($\varepsilon = 5$)	0.0000	0.1485	0.0000	0.7422	0.0000	1.0

2.3. Impact of Graph Structure on Performance

The interaction between memorisation and generalisation in neural networks is of particular interest to the privacy community. Feldman [21] hypothesises that atypical data from the long tail of the data distribution requires memorisation. This increases the negative impact of DP training for those samples. To investigate the applicability of the long-tail hypothesis to graphs, we evaluate the impact of the graph structure, measured by homophily, on model performance. Concretely, we hypothesise that graphs with low homophily are "noisier" and

Table 4: Results of the experiments on the synthetic dataset at different homophily values; All results refer to test accuracy (%), with the best ones highlighted in bold. Compare Section 2 for details. Hom.: Homophily, Subg.: Sub-graphing.

Hom.	**Non-DP**	**Clipping**	**Sub-graphing**	**Subg. + Clip.**	**DP** ($\varepsilon = 20$)	**DP** ($\varepsilon = 15$)	**DP** ($\varepsilon = 10$)	**DP** ($\varepsilon = 5$)
0.9	99.90 ± 2.00	**100.0 ± 0.00**	99.80 ± 0.40	99.90 ± 0.00	93.00 ± 4.17	96.00 ± 2.49	93.10 ± 1.36	88.70 ± 3.06
0.8	99.62 ± 0.37	99.90 ± 0.00	**100.0 ± 0.00**	99.80 ± 2.45	82.80 ± 7.83	80.60 ± 10.8	81.30 ± 6.40	80.00 ± 8.76
0.7	96.50 ± 1.23	**99.70 ± 0.40**	98.42 ± 0.51	98.20 ± 0.50	79.10 ± 3.00	76.30 ± 3.10	74.80 ± 4.55	70.30 ± 5.64
0.6	73.60 ± 0.97	**91.00 ± 2.24**	83.10 ± 3.10	85.40 ± 3.54	57.72 ± 2.38	55.50 ± 1.45	54.80 ± 3.87	57.90 ± 1.32
0.5	**66.10 ± 1.361**	53.60 ± 1.16	57.00 ± 1.76	58.10 ± 2.03	52.71 ± 3.34	51.82 ± 3.43	51.70 ± 2.38	50.70 ± 3.87

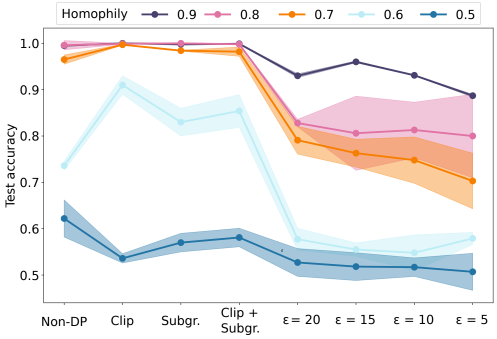


Fig. 2: Impact of sub-graphing/noise addition at different ε. *Clip*: clipping only, *Subgr.*: sub-graphing only.

therefore suffer more from DP training. For example, at homophily of 0.5 on a binary classification dataset, on average, only half of the neighbours have the same label. Thus, when applying message-passing on such a graph, the node features will get averaged over an approximately equal number of nodes from both labels. This makes it nearly impossible to learn meaningful node feature embeddings, such that learning likely relies on memorisation. The results using a synthetic dataset with different levels of homophily are summarised in Table 4 and visualised in Figure 2. The generalisation gap is especially large on low-homophily graphs, indicating over-fitting in the non-DP setting. Model accuracy in the non-DP setting profits more from the regularising effects of clipping and sub-graphing in lower-homophily graphs (0.6) compared to ones with high homophily (0.9). We note that, in a binary classification task, homophily values are symmetric about 0.5. As expected, at the lowest homophily of 0.5, learning is severely compromised without DP, and the regularising effect of clipping and sub-graphing harms accuracy. Moreover, learning is nearly impossible with DP, corroborating that, in this setting, the accuracy benefit of non-DP learning is mostly due to memorisation. Under DP, graphs with high homophily (0.9) suffer a lower performance decrease compared to low-homophily graphs, likely due to the favourable graph structure for the learning task, i.e. not requiring strong memorisation.

3. DISCUSSION, CONCLUSION, FUTURE WORK

We investigate the practicality and challenges of differentially private (DP) graph neural networks (GNNs) for medical population graphs. The utilisation of population graphs in medicine has shown promising results in performance for disease prediction [4]. However, it comes with the additional challenge of an explicit graph construction step. This can lead to poor graph structures, regarding the homogeneity of neighbourhoods, which can be measured by the homophily metric. Population graphs contain sensitive medical data of several subjects, which requires protection when applying DL methods to these graphs. Applying DP to GNNs requires special formulations of DP concepts like privacy amplification techniques and DP-SGD methods [17]. We here evaluate privacy-utility trade-offs of DP GNNs trained on medical population graphs and reveal interesting correlations between the graph structure and performance of DP GNNs. When the underlying graph structure of a dataset has low homophily (indicating diverse neighbourhoods with different labels), DP has a stronger negative impact on model performance compared to datasets with high homophily. This finding and its possible connection to the long-tail hypothesis [21] is a promising direction for future work to potentially improve DP methods for GNNs by improving the underlying graph structure. Moreover, homophily is not the only measure for the “quality” of a graph structure. Homophily is not the only measure for the “quality” of a graph structure. Further metrics such as cross-class neighbourhood similarity [22] could be evaluated, which may shed more light on the impact of graph structure on the performance of DP GNNs.

4. COMPLIANCE WITH ETHICAL STANDARDS

This study was conducted (a) retrospectively using human subject data made available in open access by [20], where ethical approval was not required and (b) in line with the principles of the Declaration of Helsinki. Approval was granted by Ethics Committee of Technical University Munich.

5. ACKNOWLEDGEMENTS

This project was funded by the German Ministry of Education and Research as part of the PrivateAIM Project and by the Medical Informatics Initiative. TM was supported by the ERC (Deep4MI-884622). Data collection of TADPOLE was funded by the Alzheimer’s Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (award number W81XWH-12-2-0012).

6. REFERENCES

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst, “Geometric deep learning: going beyond euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[2] Thomas N Kipf and Max Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv:1609.02907*, 2016.

[3] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh, “Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks,” in *25th ACM SIGKDD*, 2019, pp. 257–266.

[4] Sarah et al. Parisot, “Spectral graph convolutions for population-based disease prediction,” in *MICCAI*. Springer, 2017, pp. 177–185.

[5] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, and Doina Precup, “When do we need gnn for node classification?,” *arXiv:2210.16979*, 2022.

[6] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra, “Beyond homophily in graph neural networks: Current limitations and effective designs,” *NeurIPS*, vol. 33, pp. 7793–7804, 2020.

[7] Borja Balle, Giovanni Cherubin, and Jamie Hayes, “Reconstructing training data with informed adversaries,” in *IEEE Symposium on Security and Privacy*, 2022, pp. 1138–1156.

[8] Jamie Hayes, Saeed Mahloujifar, and Borja Balle, “Bounding training data reconstruction in dp-sgd,” *arXiv:2302.07225*, 2023.

[9] Cynthia Dwork, Aaron Roth, et al., “The algorithmic foundations of differential privacy.,” *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[10] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[11] Tamara T Mueller, Dmitrii Usynin, Johannes C Paetzold, Daniel Rueckert, and Georgios Kaissis, “Differential privacy guarantees for analytics and machine learning on graphs: A survey of results,” *JPC (Accepted)*, 2023.

[12] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang, “Node-level membership inference attacks against graph neural networks,” *arXiv:2102.05429*, 2021.

[13] Iyiola E Olatunji, Wolfgang Nejdl, and Megha Khosla, “Membership inference attack on gnns,” in *IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications*, 2021, pp. 11–20.

[14] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li, “Linkteller: Recovering private edges from graph neural networks via influence analysis,” in *IEEE Symposium on Security and Privacy*. IEEE, 2022, pp. 2005–2024.

[15] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang, “Inference attacks against graph neural networks,” in *USENIX Security Symposium*, 2022, pp. 4543–4560.

[16] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr, “Membership inference attacks from first principles,” *IEEE Symposium on Security and Privacy*, pp. 1897–1914, 2021.

[17] Ameya Daigavane, Gagan Madan, Aditya Sinha, Abhradeep Guha Thakurta, Gaurav Aggarwal, and Prateek Jain, “Node-level differentially private graph neural networks,” in *ICLR Workshop on PAIR2Struct*, 2022.

[18] Eli et al. Chien, “Differentially private decoupled graph convolutions for multigranular topology protection,” *arXiv:2307.06422*, 2023.

[19] Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein, “Latent-graph learning for disease prediction,” in *MICCAI*. Springer, 2020, pp. 643–653.

[20] Adriana et al. Di Martino, “The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism,” *Molecular psychiatry*, vol. 19, no. 6, pp. 659–667, 2014.

[21] Vitaly Feldman, “Does learning require memorization? a short tale about a long tail,” in *ACM SIGACT Symposium on Theory of Computing*, 2020, pp. 954–959.

[22] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang, “Is homophily a necessity for graph neural networks?,” *arXiv:2106.06134*, 2021.

Concluding Remarks

Applications of artificial intelligence (AI) in medicine have shown highly promising results and contributions to medical research and procedures, such as disease prediction [17], [60], image segmentation [1], or text analysis [98]. Hereby, different types of (medical) data require different methods that are adapted to the data space and the use case. Convolutional neural networks (CNNs) are currently among the most successful deep learning methods on images, using the underlying geometry of image data to extract global representations from local (pixel or voxel) information. With this, they have become very popular in supporting diverse medical tasks, such as organ segmentation [99], tumour classification [100], or concluding diagnoses [101], to only name a few.

When operating on non-Euclidean data –such as graphs or manifolds– different methods have to be developed that can directly learn from graph-structured data. These methods are summarised as graph neural networks (GNNs). GNNs show two strong advantages compared to CNNs. They are (a) faster during training and inference than CNNs and (b) have the methodological advantage of directly leveraging the underlying structure of non-Euclidean data, without the need to transform a graph dataset to another space. In this dissertation, we address parts of a broad research question of *when and how to best use GNNs for medical applications and research*. In order to address this question we target three directions: We (1) explore the limitations of the utility of GNNs, (2) investigate their strengths and advantages compared to other DL techniques, and (3) investigate privacy-preserving techniques. For this, we analyse different application areas of GNNs in medicine, analyse the impact of the graph structure on model performance, and formulate recommendations for the utilisation of GNNs in the medical domain. We extend existing methods to a wider range of use cases, discuss the general utility of GNNs for specific datasets, and design and evaluate methods for privacy-preserving graph learning on medical data. In the following sections, we discuss the individual chapters of this dissertation, highlighting the respective challenges, discussing their implications on general research in the domain of AI on non-Euclidean data, and spotlighting future directions.

11.1 Discussion

Chapter 4 - A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations

GNNs can be used to integrate data from highly different modalities by combining them in different parts of the graph structure. This can be a great opportunity for multi-modal data integration, where the information of different modalities can be represented in different parts of the graph. However, the graph construction can also be a challenge for datasets that do not come with an inherent graph structure, which is often the case in medicine. Here, graph construction methods are required to transform the datasets into graph-like structures, to render them suitable for graph deep learning. Over the years, a variety of such methods have been developed that are often highly specific to the data type at hand. We categorise relevant works for graph construction, systematise the taxonomy for static and adaptive graph learning in the medical domain, and formulate recommendations for graph construction for future research. The multitude of available methods, even for the same type of dataset, shows the complexity of this task and the requirement for standard methods and guidelines. It is important to note that the graph structure can have a severe impact on the performance of the GNN, which makes it even more important to select suitable graph construction methods that yield a “good” graph structure for subsequent deep learning. This work provides an overview of graph construction methods for graph deep learning in medicine and provides relevant recommendations for selecting suitable methods for different medical datasets and use cases. We hope for this work to function as a foundation for future work on GNNs in medicine and to facilitate future research in this area. Apart from the selection of appropriate methods, we believe it is equally important to investigate in which settings GNNs are the most beneficial method and in which other methods are more successful. We believe that our work is an important step towards further work on (a) whether it is useful to transform a dataset into graph-like structures for graph learning and (b) if so, which methods are the most promising ones. We also see a deeper investigation of different graph convolutions for different medical use cases as an important future step. This might shed more light on an optimal link between GNN methods and medical applications with the hope of improved utility.

Chapter 5 - Extended Graph Assessment Metrics for Regression and Weighted Graphs

As discussed in the previous chapter, the selection of a suitable graph construction method is important since the resulting quality of the graph can highly impact the performance of GNNs. This is, for example, relevant when working with population graphs, where initially independent data points (subjects) need to be connected in a meaningful way to construct an informative network that can be used for graph learning. In this context, the assessment of the graph structure can provide important insights into the quality of a constructed graph and hint towards potential implications on the GNN performance. Based on this, the graph construction method or the GNN architecture could be adjusted if necessary. We identify a major shortcoming of currently used graph assessment metrics as them being only applicable for unweighted graphs and classification tasks. We therefore extend two commonly used graph assessment metrics –namely homophily and cross-class neighbourhood similarity (CCNs)– to weighted graphs and if possible regression tasks. With this, we can address a wider range of applications and use cases, which require regression tasks and can potentially benefit from the utilisation of weighted graphs. We believe that the evaluation of graph assessment metrics is an important step in understanding and interpreting the results of graph learning as they allow for a better comprehension of the graph structure as well as its impact on the downstream task. Therefore, we deem the extensions addressed in this work as an important step towards having widely applicable, comparable, and comprehensive metrics for graph assessment. We hope to see more of this development in future research by extending more metrics to regression tasks and weighted graphs. Furthermore, most such metrics only investigate 1-hop neighbourhoods but could be extended to several hops, which would align well with the notion of n-hop graph neural networks and might provide more applicable insights for graph learning.

Chapter 6 - Body Fat Estimation from Surface Meshes using Graph Neural Networks

One application of graph deep learning in medicine is the usage of triangulated surface meshes. They can efficiently capture structural information and can be used to represent various forms and shapes. Here, we highlight one big advantage of GNNs compared to CNN on image data: GNNs require fewer resources and are faster to train. This can yield important utility improvements and increase the willingness to integrate such methods into medical workflows. We see this as one major promising aspect of the usage of GNNs in medical workflows since short training cycles and low resource requirements during training can reduce some of the obstacles involved in using DL methods in medicine. In this work, we use graph neural networks to estimate the quantity of body fat of two types of abdominal fatty tissue: visceral and subcutaneous fat. The distribution of body fat can be a strong indication for health-related risk factors such as the development of diseases [102], [103]. We compare the performance and run time of our GNNs trained on surface meshes to comparable approaches using CNNs on 3D magnetic resonance image (MRI) data and observe similar performance at significantly lower resource requirements. We use different mesh decimation rates to adjust the size of the meshes and investigate their impact on model performance. We note that other works [104] have shown that 2D images might be sufficient to accurately estimate different body composition values such as fat mass. This could highly reduce training time and resource requirements for CNN approaches as well and it would be interesting to also compare a 2D image-based approach with 2D and 3D meshes. In this work, we generate the surface meshes from 3D MRIs. However, we see high potential for our method to work on much cheaper and easier-to-acquire data sources, such as surface scans. They could be more easily integrated into standard medical workflows and would be more comparable to the acquisition of photographs.

Chapter 7 - Are Population Graphs Really as Powerful as Believed?

One research branch of deep learning in medicine has investigated the usage of population graphs. Here, a cohort of subjects is restructured to build a network –similar to a social network–, where subjects are connected if they show similar medical features. Related works [17], [61]–[63], [105] have shown improved prediction performance on several medical downstream tasks. However, we question said utility by showing on-par performance of well-tuned graph-agnostic methods compared to more complex GNNs. We identify the graph construction method as the main bottleneck for good performance of GNNs on population graphs and lay the ground for future theoretical analyses in this direction. We believe the latter to be highly relevant for gaining a good understanding of when and how GNNs are best used and to shed light on still unanswered questions regarding the power of GNNs. Furthermore, this work does not only question the utility of population graphs in the way they are currently used but also highlights the importance of proper usage of baseline methods in DL research in general. We believe this work to be an important step towards a better understanding of the power of GNNs and that more research needs to be done in order to render population graph studies useful. We see four directions population graphs might take in the future: (1) they render themselves to be equally useful as graph-agnostic methods in general, (2) novel and better graph construction methods will be developed, (3) different graph learning techniques are developed or used in this context, or (4) more complex data integration methods or novel use cases are employed that render population graphs more useful.

Chapter 8 - Differentially Private Guarantees for Analytics and Machine Learning on Graphs: A Survey of Results

Differential privacy (DP) is the gold standard for training deep learning models while providing formal privacy guarantees. Ensuring the privacy of sensitive data is especially relevant in medical settings, where the utilised patient data can reveal highly sensitive information. Many DP methods have been designed for datasets with rows and columns (such as image datasets or tables) and their application to graph-structured data raises additional challenges. We discuss these challenges and solutions for them by summarising, categorising, and systematising the works in this area. We specifically focus on graph learning tasks in comparison to graph analytics. We summarise different notions of DP on graph data, categorise related works in this area, show promising use cases and applications, and discuss future works. We especially highlight the variety of different notions of DP on graph-structured data, which can complicate the comparison of different methods as well as the most suitable choice of DP notion for a problem at hand. With this work, we aim to facilitate future work in the direction of privacy-preserving DL, which is of high relevance for medical applications. The trade-off between high-performing DL methods and the protection of sensitive medical data are complementary goals of high relevance and raise ethical questions since a well-performing method could potentially impact the health of patients. We believe DP methods on graph-structured data to still be under-explored and hope to encourage future research in this area with this work.

Chapter 9 - Differentially Private Graph Neural Networks for Whole-Graph Classification

One desired task of graph learning is whole-graph classification, where a network learns to predict a property for each individual graph. This can, for example, be used for molecule classification, the previously mentioned surface mesh application, or fingerprint analyses. We introduce a method for DP whole-graph classification with GNNs, allowing the usage of highly sensitive graph data while preserving privacy. Our method is based on DP-stochastic gradient descent and extends previous DP methods to graph deep learning tasks providing graph-level DP. We are able to achieve high privacy guarantees while keeping reasonable performance on several datasets. Furthermore, we utilise interpretability methods to show that DP and non-DP models consider similar components of the graph as important. This supports the hope that even though the DP models protect the whole graph structure and satisfy graph-level DP, the model performance is similar to non-DP models both in terms of predictive performance and internal decision processes. We deem a further investigation with different interpretability methods as an interesting future direction. For example, attention-based graph learning techniques could be used in addition to post-hoc methods to investigate the impact of different neighbouring nodes between private and non-private learning. With the extension of DP methods to graph-level tasks, even sensitive multi-graph datasets can be used for DL while considering privacy concerns. We believe the still present performance gap between private and non-private DL models for graph classification to be further reduced in the future and hope to have contributed towards this development with this work.

Chapter 10 - Differentially Private Graph Neural Networks for Medical Population Graphs and the Impact of the Graph Structure

Applying DP to node- or edge-level predictions is inherently more complex than for graph-level predictions since a disentanglement of individuals is impossible. In this chapter, we combine the application of differential privacy (DP) for node-level predictions with medical population graphs and investigate the impact of the graph structure on the performance of DP-GNNs. We find that the "quality" –measured by homophily in this work– is correlated with larger performance gaps on DP-GNNs. This is especially critical in settings such as population graphs, which tend to result in rather low-homophily graphs [73], [105]. We believe this to be an interesting future direction and that the comparison to other methods for node-level DP GNNs such as [94] would be valuable next steps. Furthermore, we see a potential to tie graph structure properties to the long-tail hypothesis on memorisation [106], which could give more insight into how and when GNNs memorise data and whether the graph structure has an impact on this. In the previous chapters, we have highlighted that GNNs do not out-perform well-tuned baseline methods, such as random forests, on population graph datasets. An investigation in the context of DP-trained population graphs would be highly interesting for future work. It is still an open question whether this also holds under DP guarantees.

The Power of Graph Neural Networks

Over the past years, graph deep learning techniques have become highly popular and have shown to be powerful tools that can solve a multitude of learning tasks [107]–[109]. They are promising methods to perform deep learning on a variety of datasets and data structures, often showing improved performance compared to other, graph-agnostic DL approaches. However, it has been shown that GNNs out-perform graph-agnostic models only under certain circumstances [62], [73], [108]. Therefore, several works have investigated the interplay between graph structure and GNN performance for different graph convolutions [74]. Zhu et al. [45], e.g., showed that not only the graph structure –measured by homophily or similar metrics– have an impact on the performance of GNNs, but that a separation of the ego-node features of the node of interest from the node features of each k-hop neighbourhood can improve performance. This and other works investigating the performance of GNNs on low-homophily graphs have shown that GNNs should not be used without care and that for some datasets and settings, graph-agnostic methods might even out-perform GNNs. We investigate the power of GNNs in the medical domain and show that well-tuned baselines, such as linear regression and random forest, are able to perform on par with much more complex GNNs on a variety of medical population graph datasets (Chapter 7). We deem this an important subject that not only raises the question of how and when GNNs should be utilised but also highlights the need for well-tuned baselines in (AI) research in general. However, we do not argue that GNNs are not as powerful as other DL techniques but highlight that the use case and a suitable graph structure need to be considered in order to achieve appropriate results. We believe an in-depth analysis, both empirically and theoretically, of when GNNs are most beneficial and when other DL methods should be favoured to be highly important future steps in the direction of understanding the power of GNNs and ensuring their most effective usage.

11.2 Outlook

This dissertation covers different aspects and challenges of graph deep learning in medicine. We want to highlight three main aspects of this work and how we believe this can impact future directions of research in this area:

1. **Graph deep learning can lead to faster training and fewer resource requirements than CNNs on the same task.** The mesh representation of a dataset is more memory efficient than medical images and if the mesh contains all relevant information, using them for DL speeds up the training process. This can render GNNs as highly suitable methods for routine medical examination and clinical translation. This can lead to an easier integration of DL methods into medical workflows and therefore strongly benefit medical tasks such as diagnosis or treatment planning. We believe this to be one reason for seeing more applications of GNNs in the medical domain in the future.

2. **A clear understanding of the power of GNNs will increase their impact.** We showed that medical population graphs in combination with GNNs do not outperform graph-agnostic methods in the currently used setups –if they are tuned appropriately. We identify the graph construction step as the major bottleneck for these applications and believe that the development of novel and more suitable graph construction methods is an important next step for future research in this area. We furthermore believe that this work lays the foundation for future theoretical analyses of the power of GNNs, which would be highly beneficial for the research community and increase the impact of GNNs on all downstream tasks.

3. **Differentially private GNNs are becoming more powerful.** The research area of differentially private GNNs has plenty of open questions to investigate. Node- and edge-level DP methods still under-perform non-private GNNs more drastically than privacy-preserving ML in the imaging domain. We believe DP GNNs to become more powerful in the future and the performance gap between private and non-private networks to be reduced further, also for edge- and node-level DP.

In this dissertation, we investigated several different aspects of the utilisation of GNNs in medicine to which we see many subsequent connection points and promising possibilities to further improve the understanding and effectiveness of GNNs in healthcare applications. It would, for instance, be interesting to investigate additional graph convolutions on medical datasets, ranging from heterogeneous GNNs [110] to high-order message-passing schemes [111]. Heterogeneous GNNs can be useful for more complex graphs, where several different entities are combined, such as knowledge graphs [112]. Furthermore, the investigation of unsupervised learning techniques for medical applications, such as anomaly detection, would be interesting. Graph clustering methods or learning-based methods for anomaly detection of graph data in medicine could be highly valuable in identifying diseases or pathologies. We see high value in a future investigation of GNNs compared to other types of DL, such as natural language processing and transformer networks, which have recently gained a lot of attention in the DL community. Comparisons between these methods and GNNs in the medical domain could provide more insights into the power of GNNs and their applicability to medical tasks and use cases. The definition of clear (theoretical) boundaries of when GNNs are a suitable choice for solving a problem at hand would (a) give strong evidence for their applicability and expressiveness and (b) guide future research in the direction where GNNs can solve previously unsolved tasks. Finally, we believe that there is high potential in further investigating the combination of differential privacy (DP) and graph neural networks, especially for node- and edge-level DP. The authors of [94], for instance, argue that previously used methods such as [113] are not able to provide satisfying privacy-utility trade-offs and propose a novel method for DP-GNN training for node-level DP. A combination of these methods with medical data and the impact of the graph structure under different privacy guarantees would be interesting to explore. Furthermore, an investigation of the interplay between different graph convolutions and DP guarantees could improve the performance of DP GNNs by constructing graph convolutions that are especially insensitive to the addition of DP methods.

Overall, with the works summarised in this dissertation, we believe to contribute to a better understanding of how and when GNNs can be used for medical tasks and aim to guide future research with the goal of obtaining the most beneficial and well-targeted application of GNNs for medical research and workflows. By showcasing the applicability of differential privacy to different graph-learning tasks in medicine, we provide more insights into the important subject of privacy-preserving graph-learning techniques. We believe privacy-preserving DL on sensitive medical data to become more ubiquitous in the future and therefore hope to advance the research in this domain.

Appendices

Bibliography

[1] D. D. Patil and S. G. Deore. “Medical Image Segmentation: A Review.” In: *International Journal of Computer Science and Mobile Computing* 2.1 (2013), pp. 22–27.

[2] R. T. Sutton, D. Pincock, D. C. Baumgart, D. C. Sadowski, R. N. Fedorak, and K. I. Kroeker. “An Overview of Clinical Decision Support Systems: Benefits, Risks, and Strategies for Success.” In: *NPJ Digital Medicine* 3.1 (2020), p. 17.

[3] S. Liang, F. Tang, X. Huang, K. Yang, T. Zhong, R. Hu, S. Liu, X. Yuan, and Y. Zhang. “Deep-learning-based Detection and Segmentation of Organs at Risk in Nasopharyngeal Carcinoma Computed Tomographic Images for Radiotherapy Planning.” In: *European Radiology* 29 (2019), pp. 1961–1967.

[4] D. Assefa, H. Keller, C. Ménard, N. Laperriere, R. J. Ferrari, and I. Yeung. “Robust Texture Features for Response Monitoring of Glioblastoma Multiforme on Weighted and FLAIR MR Images: A Preliminary Investigation in Terms of Identification and Segmentation.” In: *Medical Physics* 37.4 (2010), pp. 1722–1736.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. “Geometric Deep Learning: Going Beyond Euclidean Data.” In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[6] S. Mallat. “Group Invariant Scattering.” In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398.

[7] J. Bruna and S. Mallat. “Invariant Scattering Convolution Networks.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1872–1886.

[8] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam. “A Mathematical Motivation for Complex-valued Convolutional Networks.” In: *Neural computation* 28.5 (2016), pp. 815–825.

[9] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. “Graph Neural Networks for Social Recommendation.” In: *The World Wide Web Conference*. 2019, pp. 417–426.

[10] Q. Liu, M. Nickel, and D. Kiela. “Hyperbolic Graph Neural Networks.” In: *Advances in Neural Information Processing Systems* 32 (2019).

[11] B. Shakibajahromi, E. Kim, and D. E. Breen. “RIMeshGNN: A Rotation-Invariant Graph Neural Network for Mesh Classification.” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 3150–3160.

[12] K. Lin, L. Wang, and Z. Liu. “Mesh Graphormer.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 12939–12948.

[13] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui. “Graph Neural Networks in Recommender Systems: A Survey.” In: *ACM Computing Surveys* 55.5 (2022), pp. 1–37.

[14] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, and J. Wang. “A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs.” In: *IEEE Access* 10 (2022), pp. 75729–75741.

[15] J. T. Moreira-Filho, M. F. B. da Silva, J. V. V. B. Borba, A. R. Galvão Filho, E. Muratov, C. H. Andrade, R. de Campos Braga, and B. J. Neves. “Artificial Intelligence Systems for the Design of Magic Shotgun Drugs.” In: *Artificial Intelligence in the Life Sciences* (2022), p. 100055.

[16] P. Pham, L. T. Nguyen, W. Pedrycz, and B. Vo. “Deep Learning, Graph-based Text Representation and Classification: A Survey, Perspectives and Challenges.” In: *Artificial Intelligence Review* (2022), pp. 1–35.

[17] S. Parisot, S. I. Ktena, E. Ferrante, M. Lee, R. G. Moreno, B. Glocker, and D. Rueckert. “Spectral Graph Convolutions for Population-based Disease Prediction.” In: *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer. 2017, pp. 177–185.

[18] D. Klepl, F. He, M. Wu, D. J. Blackburn, and P. Sarrigiannis. “EEG-Based Graph Neural Network Classification of Alzheimer’s Disease: An Empirical Evaluation of Functional Connectivity Methods.” In: *IEEE Trans. on Neural Systems and Rehabilitation Engineering* 30 (2022), pp. 2651–2660.

[19] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller. "Inverting Gradients-How Easy is it to Break Privacy in Federated Learning?" In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16937–16947.

[20] G. Buzaglo, N. Haim, G. Yehudai, G. Vardi, Y. Oz, Y. Nikankin, and M. Irani. "Deconstructing Data Reconstruction: Multiclass, Weight Decay and General Losses." In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine. Vol. 36. Curran Associates, Inc., 2023, pp. 51515–51535.

[21] N. Haim, G. Vardi, G. Yehudai, O. Shamir, and M. Irani. "Reconstructing Training Data from Trained Neural Networks." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 22911–22924.

[22] K.-C. Wang, Y. Fu, K. Li, A. Khisti, R. Zemel, and A. Makhzani. "Variational Model Inversion Attacks." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 9706–9719.

[23] K. Ramesh, G. K. Kumar, K. Swapna, D. Datta, and S. S. Rajest. "A Review of Medical Image Segmentation Algorithms." In: *EAI Endorsed Transactions on Pervasive Health and Technology* 7.27 (2021), e6–e6.

[24] P. Aggarwal, R. Vig, S. Bhadoria, and C. Dethe. "Role of Segmentation in Medical Imaging: A Comparative Study." In: *International Journal of Computer Applications* 29.1 (2011), pp. 54–61.

[25] M. Baumgartner, P. F. Jäger, F. Isensee, and K. H. Maier-Hein. "nnDetection: A Self-configuring Method for Medical Object Detection." In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24*. Springer. 2021, pp. 530–539.

[26] R. Yang and Y. Yu. "Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis." In: *Frontiers in Oncology* 11 (2021), p. 638182.

[27] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. "Deep Learning for Medical Anomaly Detection–A Survey." In: *ACM Computing Surveys (CSUR)* 54.7 (2021), pp. 1–37.

[28] M. Zhang, A. Raghunathan, and N. K. Jha. "MedMon: Securing Medical Devices Through Wireless Monitoring and Anomaly Detection." In: *IEEE Transactions on Biomedical Circuits and Systems* 7.6 (2013), pp. 871–881.

[29] R. Hartshorne. *Geometry: Euclid and Beyond*. Springer Science & Business Media, 2013.

[30] R. A. Johnson. *Advanced Euclidean Geometry*. Courier Corporation, 2013.

[31] C. Cagniart, E. Boyer, and S. Ilic. “Free-form Mesh Tracking: A Patch-based Approach.” In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2010, pp. 1339–1346.

[32] J.-P. Thirion. “The Extremal Mesh and the Understanding of 3D Surfaces.” In: *International Journal of Computer Vision* 19 (1996), pp. 115–128.

[33] T. T. Mueller*, S. Zhou*, S. Starck, F. Jungmann, A. Ziller, O. Aksoy, D. Movchan, R. Braren, G. Kaissis, and D. Rueckert. “Body Fat Estimation from Surface Meshes Using Graph Neural Networks.” In: *Lecture Notes in Computer Science; Presented at Shape in Medical Imaging: International Workshop, ShapeMI 2023, Held in Conjunction with MICCAI 2023, Vancouver, BC, Canada, October 8, 2023, Proceedings*. Springer Nature Switzerland, 2023, pp. 105–117.

[34] L. Lebrat, R. Santa Cruz, F. de Gournay, D. Fu, P. Bourgeat, J. Fripp, C. Fookes, and O. Salvado. “CorticalFlow: A Diffeomorphic Mesh Transformer Network for Cortical Surface Reconstruction.” In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29491–29505.

[35] F. Bongratz, A.-M. Rickmann, and C. Wachinger. “Abdominal Organ Segmentation via Deep Diffeomorphic Mesh Deformations.” In: *Scientific Reports* 13.1 (2023), p. 18270.

[36] L. Cai, J. Gao, and D. Zhao. “A Review of the Application of Deep Learning in Medical Image Classification and Segmentation.” In: *Annals of Translational Medicine* 8.11 (2020).

[37] L. Yang, B. Georgescu, Y. Zheng, Y. Wang, P. Meer, and D. Comaniciu. “Prediction Based Collaborative Trackers (PCT): A Robust and Accurate Approach Toward 3D Medical Object Tracking.” In: *IEEE Transaction on Medical Imaging* 30.11 (2011), pp. 1921–1932.

[38] Y. LeCun, Y. Bengio, and G. Hinton. “Deep Learning.” In: *Nature* 521.7553 (2015), pp. 436–444.

[39] L. Deng, D. Yu, et al. “Deep Learning: Methods and Applications.” In: *Foundations and Trends in Signal Processing* 7.3–4 (2014), pp. 197–387.

[40] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016.

[41] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. "Neural Message Passing for Quantum Chemistry." In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1263–1272.

[42] T. N. Kipf and M. Welling. "Semi-Supervised Classification with Graph Convolutional Networks." In: *International Conference on Learning Representations*. 2017.

[43] W. Hamilton, Z. Ying, and J. Leskovec. "Inductive Representation Learning on Large Graphs." In: *Advances in Neural Information Processing Systems* 30 (2017).

[44] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al. "Graph Attention Networks." In: *Stat* 1050.20 (2017), pp. 10–48550.

[45] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. "Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 7793–7804.

[46] J.-W. Hu and W.-W. Zhang. "Mesh-Conv: Convolution Operator with Mesh Resolution Independence for Flow Field Modeling." In: *Journal of Computational Physics* 452 (2022), p. 110896.

[47] S. Brody, U. Alon, and E. Yahav. "How Attentive are Graph Attention Networks?" In: *International Conference on Learning Representations*. 2022.

[48] J. Hansen and T. Gebhart. "Sheaf Neural Networks." In: *TDA & Beyond*. 2020.

[49] C. Bodnar, F. Di Giovanni, B. Chamberlain, P. Liò, and M. Bronstein. "Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18527–18541.

[50] T. T. Mueller, S. Starck, A. Dima, S. Wunderlich, K.-M. Bintsi, K. Zaripova, R. Braren, D. Rueckert, A. Kazi, and G. Kaissis. "A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations." In: *Transactions on Machine Learning Research* (2024).

[51] M. A. Gharsallaoui, F. Tornaci, and I. Rekik. "Investigating and Quantifying the Reproducibility of Graph Neural Networks in Predictive Medicine." In: *PRIMEMICCAI*. 2021, pp. 104–116.

[52] R. Yu, C. Pan, X. Fei, M. Chen, and D. Shen. “Multi-Graph Attention Networks with Bilinear Convolution for Diagnosis of Schizophrenia.” In: *IEEE Journal of Biomedical and Health Informatics* (2023).

[53] A. Bessadok, M. A. Mahjoub, and I. Rekik. “Graph Neural Networks in Network Neuroscience.” In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2022).

[54] J. C. Paetzold, J. McGinnis, S. Shit, I. Ezhov, P. Büschl, C. Prabhakar, A. Sekuboyina, M. Todorov, G. Kaissis, A. Ertürk, et al. “Whole Brain Vessel Graphs: A Dataset and Benchmark for Graph Learning and Neuroscience.” In: *NeurIPS Datasets and Benchmarks*. 2021.

[55] T. Zhao and Z. Yin. “Airway Anomaly Detection by Prototype-Based Graph Neural Network.” In: *MICCAI*. 2021, pp. 195–204.

[56] R. Selvan, T. Kipf, M. Welling, A. G.-U. Juarez, J. H. Pedersen, J. Petersen, and M. de Bruijne. “Graph Refinement Based Airway Extraction Using Mean-Field Networks and Graph Neural Networks.” In: *Medical Image Analysis* 64 (2020), p. 101751.

[57] S. Bonner, I. P. Barrett, C. Ye, R. Swiers, O. Engkvist, A. Bender, C. T. Hoyt, and W. L. Hamilton. “A Review of Biomedical Datasets Relating to Drug Discovery: A Knowledge Graph Perspective.” In: *Briefings in Bioinformatics* 23.6 (2022).

[58] P. Chandak, K. Huang, and M. Zitnik. “Building a Knowledge Graph to Enable Precision Medicine.” In: *Scientific Data* 10.1 (2023), p. 67.

[59] T. T. Mueller, J. C. Paetzold, C. Prabhakar, D. Usynin, D. Rueckert, and G. Kaissis. “Differentially Private Graph Neural Networks for Whole-Graph Classification.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.6 (2023), pp. 7308–7318.

[60] A. Kazi, L. Cosmo, S.-A. Ahmadi, N. Navab, and M. M. Bronstein. “Differentiable Graph Module (DGM) for Graph Convolutional Networks.” In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 45.2 (2022), pp. 1606–1617.

[61] K.-M. Bintsi, V. Baltatzis, R. A. Potamias, A. Hammers, and D. Rueckert. “Multimodal Brain Age Estimation Using Interpretable Adaptive Population-Graph Learning.” In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2023, pp. 195–204.

[62] L. Cosmo, A. Kazi, S.-A. Ahmadi, N. Navab, and M. Bronstein. “Latent-Graph Learning for Disease Prediction.” In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 643–653.

[63] A. Kazi, S. Shekarforoush, S. Arvind Krishna, H. Burwinkel, G. Vivar, K. Kortüm, S.-A. Ahmadi, S. Albarqouni, and N. Navab. “InceptionGCN: Receptive Field Aware Graph Convolutional Network for Disease Prediction.” In: *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*. Springer. 2019, pp. 73–85.

[64] W. E. Lorensen and H. E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm.” In: *Seminal graphics: pioneering efforts that shaped the field*. 1998, pp. 347–353.

[65] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. “Geom-GCN: Geometric Graph Convolutional Networks.” In: *International Conference on Learning Representations*. 2020.

[66] Y. Ma, X. Liu, N. Shah, and J. Tang. “Is Homophily a Necessity for Graph Neural Networks?” In: *International Conference on Learning Representations*. 2022.

[67] D. Lim, X. Li, F. Hohne, and S.-N. Lim. “New Benchmarks for Learning on Non-Homophilous Graphs.” In: *arXiv:2104.01404* (2021).

[68] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. “Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification?” In: *arXiv:2109.05641* (2021).

[69] Z. Hu, Y. Dong, K. Wang, and Y. Sun. “Heterogeneous Graph Transformer.” In: *Proceedings of the Web Conference*. 2020, pp. 2704–2710.

[70] X. Mo, Y. Xing, and C. Lv. “Heterogeneous Edge-Enhanced Graph Attention Network For Multi-Agent Trajectory Prediction.” In: *CoRR* abs/2106.07161 (2021).

[71] S. Luan, C. Hua, M. Xu, Q. Lu, J. Zhu, X.-W. Chang, J. Fu, J. Leskovec, and D. Precup. “When Do Graph Neural Networks Help with Node Classification? Investigating the Homophily Principle on Node Distinguishability.” In: *Advances in Neural Information Processing Systems* 36 (2024).

[72] O. Platonov, D. Kuznedelev, A. Babenko, and L. Prokhorenkova. “Characterizing Graph Datasets for Node Classification: Beyond Homophily-Heterophily Dichotomy.” In: *arXiv preprint arXiv:2209.06177* (2022).

[73] T. T. Mueller, S. Starck, K.-M. Bintsi, A. Ziller, R. Braren, G. Kaissis, and D. Rueckert. “Are Population Graphs Really as Powerful as Believed?” In: *Transactions on Machine Learning Research* (2024).

[74] S. Luan, C. Hua, Q. Lu, J. Zhu, X.-W. Chang, and D. Precup. “When Do We Need GNN for Node Classification?” In: *arXiv:2210.16979* (2022).

[75] H. Yuan, H. Yu, S. Gui, and S. Ji. “Explainability in Graph Neural Networks: A Taxonomic Survey.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.5 (2022), pp. 5782–5799.

[76] C. Agarwal, M. Zitnik, and H. Lakkaraju. “Probing GNN Explainers: A Rigorous Theoretical and Empirical Analysis of GNN Explanation Methods.” In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 8969–8996.

[77] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. “Grad-cam: Visual Explanations from Deep Networks via Gradient-based Localization.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 618–626.

[78] Z. Salahuddin, H. C. Woodruff, A. Chatterjee, and P. Lambin. “Transparency of Deep Neural Networks for Medical Image Analysis: A Review of Interpretability Methods.” In: *Computers in Biology and Medicine* 140 (2022), p. 105111.

[79] S. Starck*, Y. V. Kini*, J. J. M. Ritter, R. Braren, D. Rueckert, and T. T. Mueller. “Atlas-Based Interpretable Age Prediction In Whole-Body MR Images.” In: *arXiv preprint arXiv:2307.07439* (2023). Accepted at iMIMIC Workshop at MICCAI 2023.

[80] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. “GNNExplainer: Generating Explanations for Graph Neural Networks.” In: *Advances in Neural Information Processing Systems* 32 (2019).

[81] M. Al-Rubaie and J. M. Chang. “Privacy-Preserving Machine Learning: Threats and Solutions.” In: *IEEE Security & Privacy* 17.2 (2019), pp. 49–58.

[82] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer. “Membership Inference Attacks From First Principles.” In: *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2022, pp. 1897–1914.

[83] C. Dwork, A. Roth, et al. "The Algorithmic Foundations of Differential Privacy." In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407.

[84] A. Narayanan and V. Shmatikov. "Robust De-Anonymization of Large Sparse Datasets." In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE. 2008, pp. 111–125.

[85] A. Cohen and K. Nissim. "Towards Formalizing the GDPR's Notion of Singling Out." In: *Proceedings of the National Academy of Sciences* 117.15 (2020), pp. 8344–8352.

[86] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov. "See Through Gradients: Image Batch Recovery via Gradinversion." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16337–16346.

[87] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot. "When the Curious Abandon Honesty: Federated Learning is not Private." In: *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2023, pp. 175–199.

[88] L. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein. "Robbing the Fed: Directly Obtaining Private Data in Federated Learning with Modified Models." In: *Tenth International Conference on Learning Representations* (2022).

[89] G. Buzaglo, N. Haim, G. Yehudai, G. Vardi, Y. Oz, Y. Nikankin, and M. Irani. "Deconstructing Data Reconstruction: Multiclass, Weight Decay and General Losses." In: *Advances in Neural Information Processing Systems* 36 (2024).

[90] A. Ziller, T. T. Mueller, R. Braren, D. Rueckert, and G. Kaissis. "Privacy: An Axiomatic Approach." In: *Entropy* 24.5 (2022), p. 714.

[91] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. "Deep Learning with Differential Privacy." In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 308–318.

[92] B. Balle, G. Barthe, and M. Gaboardi. "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences." In: *Advances in neural information processing systems* 31 (2018).

[93] A. Daigavane, G. Madan, A. Sinha, A. G. Thakurta, G. Aggarwal, and P. Jain. “Node-Level Differentially Private Graph Neural Networks.” In: *ICLR 2022 Workshop on PAIR2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data.* 2022.

[94] Z. Xiang, T. Wang, and D. Wang. “Preserving Node-level Privacy in Graph Neural Networks.” In: *arXiv preprint arXiv:2311.06888* (2023).

[95] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar. “Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data.” In: *International Conference on Learning Representations.* 2017.

[96] I. E. Olatunji, T. Funke, and M. Khosla. “Releasing Graph Neural Networks with Differential Privacy Guarantees.” In: *Transactions on Machine Learning Research* (2023).

[97] T. T. Mueller*, M. Chevli*, A. Daigavane, D. Rueckert, and G. Kaissis. “Differentially Private Graph Neural Networks for Medical Population Graphs and the Impact of the Graph Structure.” In: *IEEE 21st International Symposium on Biomedical Imaging (ISBI).* Accepted. 2024.

[98] S. Locke, A. Bashall, S. Al-Adely, J. Moore, A. Wilson, and G. B. Kitchen. “Natural Language Processing in Medicine: A Review.” In: *Trends in Anaesthesia and Critical Care* 38 (2021), pp. 4–9.

[99] W. Bai, M. Sinclair, G. Tarroni, O. Oktay, M. Rajchl, G. Vaillant, A. M. Lee, N. Aung, E. Lukaschuk, M. M. Sanghvi, et al. “Automated Cardiovascular Magnetic Resonance Image Analysis with Fully Convolutional Networks.” In: *Journal of Cardiovascular Magnetic Resonance* 20.1 (2018), pp. 1–12.

[100] O. Iizuka, F. Kanavati, K. Kato, M. Rambeau, K. Arihiro, and M. Tsuneki. “Deep Learning Models for Histopathological Classification of Gastric and Colonic Epithelial Tumours.” In: *Scientific reports* 10.1 (2020), p. 1504.

[101] S. Dabeer, M. M. Khan, and S. Islam. “Cancer Diagnosis in Histopathological Image: CNN Based Approach.” In: *Informatics in Medicine Unlocked* 16 (2019), p. 100231.

[102] G. 2. O. Collaborators. “Health Effects of Overweight and Obesity in 195 Countries over 25 Years.” In: *New England journal of medicine* 377.1 (2017), pp. 13–27.

[103] E. E. Calle, C. Rodriguez, K. Walker-Thurmond, and M. J. Thun. “Overweight, Obesity, and Mortality from Cancer in a Prospectively Studied Cohort of US Adults.” In: *New England Journal of Medicine* 348.17 (2003), pp. 1625–1638.

[104] I. Y. Tian, B. K. Ng, M. C. Wong, S. Kennedy, P. Hwaung, N. Kelly, E. Liu, A. K. Garber, B. Curless, S. B. Heymsfield, et al. “Predicting 3D Body Shape and Body Composition from Conventional 2D Photography.” In: *Medical Physics* 47.12 (2020), pp. 6232–6245.

[105] K.-M. Bintsi, T. T. Mueller, S. Starck, V. Baltatzis, A. Hammers, and D. Rueckert. “A Comparative Study of Population-Graph Construction Methods and Graph Neural Networks for Brain Age Regression.” In: *Graphs in Biomedical Image Analysis, and Overlapped Cell on Tissue Dataset for Histopathology.* Ed. by S.-A. Ahmadi and S. Pereira. Cham: Springer Nature Switzerland, 2024, pp. 64–73.

[106] V. Feldman. “Does Learning Require Memorization? A Short Tale About a Long Tail.” In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 2020, pp. 954–959.

[107] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. “Graph Neural Networks: A Review of Methods and Applications.” In: *AI open* 1 (2020), pp. 57–81.

[108] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson. “Graph-based Deep Learning for Medical Diagnosis and Analysis: Past, Present and Future.” In: *Sensors* 21.14 (2021), p. 4758.

[109] A. Gupta, P. Matta, and B. Pant. “Graph Neural Network: Current state of Art, Challenges and Applications.” In: *Materials Today: Proceedings* 46 (2021). International Conference on Technological Advancements in Materials Science and Manufacturing, pp. 10927–10932.

[110] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, and Y. Ye. “Heterogeneous Graph Structure Learning for Graph Neural Networks.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 5. 2021, pp. 4697–4705.

[111] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. “Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 4602–4609.

[112] Z. Li, H. Liu, Z. Zhang, T. Liu, and N. N. Xiong. “Learning Knowledge Graph Embedding with Heterogeneous Relation Attention Networks.” In: *IEEE Transactions on Neural Networks and Learning Systems* 33.8 (2021), pp. 3961–3973.

[113] A. Daigavane, B. Ravindran, and G. Aggarwal. “Understanding Convolutions on Graphs.” In: *Distill* (2021).

Supplementary Material: Extended Graph Assessment Metrics for Regression and Weighted Graphs

A Further Information on Extended Graph Assessment Metrics

A.1 *K*-hop metrics

We here formally define k-hop node homophily and k-hop CCNs.

Definition 8 (k-hop node homophily). *A graph $G := (V, E)$ with the set of node labels $Y := \{y_u; u \in V\}$ has the following k-hop node homophily:*

$$h^{(k)}(G, Y) := \frac{1}{|V|} \sum_{v \in V} \frac{\left|\{u | u \in \mathcal{N}_v^{(k)}, y_u = y_v\}\right|}{|\mathcal{N}_v^{(k)}|}, \tag{8}$$

where $\mathcal{N}_v^{(k)}$ is the set of nodes in the k-hop neighbourhood of v.

Definition 9 (k-hop CCNS). *A graph $G := (V, E)$ with the set of node labels C has the following k-hop CCNS for two classes c and c':*

$$\mathrm{CCNS}(c, c') = \frac{1}{|\mathcal{V}_c||\mathcal{V}_{c'}|} \sum_{u \in \mathcal{V}, v \in \mathcal{V}'} \mathrm{cossim}(d^{(k)}(u), d^{(k)}(v)), \tag{9}$$

where $d^{(k)}(v)$ indicates the empirical histogram of the labels of the k-hop neighbours of node v and $\mathrm{cossim}(\cdot, \cdot)$ *the cosine similarity.*

Table 3 summarises model performances of $\{1, 2, 3\}$-hop GCNs on the different benchmark datasets and the corresponing MLP performance on the node features only. We can see that even though 3-hop homophily of the datasets Computers and Photo is very low, the GCNs with 3 hops perform best on these datasets. This does not align with our initial intuition about these metrics and we believe this finding to be interesting to investigate further.

Table 3. Graph metrics of benchmark node classification datasets with corresponding performances of an MLP and 1,2, and 3-hop GCNs, reported in accuracy in %. Nodes: number of nodes, Cl.: number of classes in the dataset.

Dataset	**Nodes**	**Cl.**	**Node homophily**			**MLP**	**GCN**		
			1-hop	**2-hop**	**3-hop**		**1-hop**	**2-hop**	**3-hop**
Cora	1 433	7	0.825 ± 0.29	0.775 ± 0.26	0.663 ± 0.29	60.41	76.33	**81.70**	78.90
Citeseer	3 703	6	0.706 ± 0.40	0.754 ± 0.28	0.712 ± 0.29	61.19	71.20	**72.10**	67.10
Pubmed	19 717	3	0.792 ± 0.35	0.761 ± 0.26	0.687 ± 0.26	74.00	76.60	**79.10**	77.70
Computers	13 752	10	0.785 ± 0.26	0.569 ± 0.27	0.303 ± 0.20	79.35	39.27	67.56	**83.13**
Photo	7 650	8	0.837 ± 0.25	0.660 ± 0.30	0.447 ± 0.28	82.09	48.10	82.88	**88.37**
Coauthor CS	18 333	15	0.832 ± 0.24	0.698 ± 0.25	0.520 ± 0.25	88.93	**93.13**	89.31	92.09

A.2 Node-wise metrics

The k-hop homophily for regression can also defined for every node individually and then combined in the full homophily over the entire graph as defined in the main part of this work.

Definition 10 (Homophily for regression). *Let $G = (V, E)$ and $\mathcal{N}_v^k$ be defined as above and Y be the vector of node labels, which is normalised between 0 and 1. Then the k-hop homophily of a node $v \in V_c$ in a node regression task is defined as the mean label distance between the node v and all it's neighbours.*

$$\mathrm{HReg}_v^{(k)} := 1 - \left(\frac{1}{|\mathcal{N}_v^{(k)}|} \sum_{n \in \mathcal{N}_v^{(k)}} \|y_v - y_n\| \right), \tag{10}$$

where $|\cdot|$ is the cardinality of a set and $\|x\|$ the absolute value of x.

The k-hop homophily for regression of the whole graph G can then be extracted as follows:

$$\begin{aligned} \mathrm{HReg}_G^{(k)} &:= 1 - \left(\frac{1}{|V|} \sum_{v \in V} \mathrm{HReg}_v^{(k)} \right) \\ &= 1 - \left(\frac{1}{|V|} \sum_{v \in V} \left(\frac{1}{|\mathcal{N}_v^{(k)}|} \sum_{n \in \mathcal{N}_v^{(k)}} \|y_v - y_n\| \right) \right). \end{aligned} \tag{11}$$

B Experiments

In this section we give more details on training parameters and setups of the experiments performed in this work.

B.1 Synthetic dataset

The synthetic datasets are generated using *sklearn* [4]. Each dataset consist of either 1 000 or 2 000 nodes, with 50 node features of which 5 are informative. For all experiments on the synthetic dataset we utilise early stopping and the initial graph structure is generated using the k-nearest neighbours approach with 5 neighbours and the Euclidean distance.

B.2 TADPOLE dataset

We use the same TADPOLE dataset as in [9], which consists of 564 subjects. The task is the classification of Alzheimer's disease, mild cognitive impairment and control normal. For all experiments on the TADPOLE dataset, we use early stopping and generate the initial graph structure using the k-nearest neighbours approach and the Euclidean distance.

B.3 UKBB dataset

The dDGM experiments on the UKBB dataset are performed using no initial graph structure, since this resulted in better model performance. For the cDGM experiments we use the k-nearest neighbours approach with x neighbours. We utilised no early stopping and the Euclidean distance for the graph construction for the cDGM experiments.

B.4 Baseline results

Table 4 summarises the baseline results on the population graph datasets using a random forest and the implementation from sklearn [4].

Table 4. Baseline results using random forests on the different datasets. For classification tasks, we report accuracy in % and for regression MAE. We report the mean and standard deviation of a 5-fold cross validation.

Dataset	**Nr. nodes**	**Task**	**Test Score**
Synthetic	1000	Binary classification Regression	78.00 ± 0.07 0.0529 ± 0.01
	2000	Binary classification Regression	88.10 ± 0.02 0.0081 ± 0.00
Tadpole	564	Classification	94.15 ± 0.01
UKBB	6406	Regression	4.2644 ± 0.05

Supplementary Material: Are Population Graphs Really as Powerful as Believed?

A Additional Information on the Datasets

We here provide some additional information on some of the population graph datasets.

TADPOLE For the TADPOLE dataset, we follow the approach from Kazi et al. (2022) and use the same features as in their work.

ABIDE For the ABIDE dataset, we follow the approach from Parisot et al. (2017) and use the following non-imaging features: Sex and site. The imaging features are extracted in the same way as in their work.

UKBB cardiac We use the following non-imaging features from the UKBB: Age, sex, body fat percentage, smoking status, body mass index, and the frequency of exercises in the last four weeks. The imaging features are extracted from these subjects' cardiac magnetic resonance images (MRIs) and contain information such as end-diastolic, end-systolic volume, stroke volume, and ejection fraction for both ventricles and myocardial-wall thickness. More information about the imaging features can be found in Bai et al. (2020).

COVID The COVID dataset is an in-house dataset, with the task of predicting whether a CoViD patient will require an intensive care unit. The non-imaging features are demographics, blood values, and prior diseases such as age, sex, fever, coughing, the loss of taste or smell, other symptoms, immunosuppressors, duration of symptomatic, shortness of breath, GIT symptoms, neurological symptoms, acute, prior diseases, temperature, oxygen saturation.

UKBB brain age For this dataset, we follow the approach from Bintsi et al. (2023a) for both imaging and non-imaging features. We use the same non-imaging features as in the original work: Sex, weight, height, body mass index, systolic blood pressure, diastolic blood pressure, college education, smoking status, alcohol intake frequency, stroke, diabetes, walking per week, moderate exercising per week, vigorous exercising per week, fluid intelligence, tower rearranging: number of puzzles correct, trail making task: duration to complete numeric path trail 1, trail making task: duration to complete alphanumeric path trail 2, matrix pattern completion: number of puzzles correctly solved, matrix pattern completion: duration spent answering each puzzle.

Synthetic dataset We generate a synthetic dataset using `sklearn` with 4 classes and a varying number of nodes to investigate the impact of the dataset size on the GNN performance. We use 50 node features, of which 10 are informative.

B Hyperparameters and Model Architectures

We summarise the hyperparameter ranges used for the sweeps for our experiments in Table 10. We distinguish between experiments using static graph construction, dynamic graph construction, and baseline tuning.

C Additional Results

We here summarise the results of additional experiments to the ones reported in the main text on the datasets CORA (Table 11), TADPOLE (Table 12), UKBB brain age (Table 13), and UKBB cardiac (Table 14). For example, the performance of the GNNs on an imitated graph structure that only contains self-loops simulates transductive learning without a meaningful graph structure and a graph construction using k-NN with the cosine distance. All here summarised experiments follow the same setup as introduced in Section 4.4. Figure 6 visualises more results following the same approach as in Section 5.4 with the additional dataset UKBB cardiac and larger.

C.1 Benchmark Datasets

With the experiments on the CORA dataset (Table 11), we observe that only the GNNs that utilise the "ground truth" edges out-perform our baseline methods, while the commonly used graph construction methods

	Parameter	**Range**
All	Learning rate	`[0.00001; 0.09]`
	Dropout	`[0.0,0.1,0.2,0.3,0.4]`
	k	`[2,5,10,20]`
	Convolutions	`[GAT, GCN, GraphConv, GraphSAGE]`
St.	Nr. layers	`[1,2,3]`
	Hidden channels	32
Dyn.	FC layers	`[[32,8,1], [8,8,3]]`
	DGM layers	`[[[32,16,4]], [[32,16,4],[],[]]]`
	Conv layers	`[[[32,32]], [[32,32],[32,16],[16,8]]]`
Neural Sheaf	d	`[2,3,4]`
	Add lp	`[0,1]`
	Add hp	`[0,1]`
	Nr. layers	`[2,3,4,5,6]`
	Hidden channels	`[8,16,32]`
	Input dropout	`[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]`
	Dropout	`[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]`
	Learning rate	`[0.02, 0.01, 0.05, 0.05, 0.001]`
	Sheaf type	`[BundleSheaf, DiagSheaf, GeneralScheaf]`
RF	Max depth	`[2;20]`
	Nr. estimators	`[500;2000]`
R	Alpha	`[0.001; 50]`

Table 10: Hyperparameter ranges for static and dynamic graph construction experiments. **All GNN**: for all GNN experiments, **St.**: parameters for static experiments only, **Dyn.**: for dynamic experiments only, **Neural Sheaf**: experiments with neural sheaf diffusion models, **RF**: random forest experiments, **R**: ridge classifier/regressor experiments.

for population graphs also do not benefit performance on the CORA dataset. We observed similar results for the other benchmark datasets.

C.2 Population Graph Datasets

Tables 12, 13, and 14 show the results of additional experiments on the TADPOLE, UKBB brain age, and UKBB cardiac datasets, respectively. We here also test the performance of GNNs on a graph that only contains self-loops, which mimics a transductive learning setting without actually using a graph structure. This rules out that a potential performance increase of GNNs stems from the fact that all node features are seen during training, which is not the case for standard baseline models, such as random forests or linear regressions. However, we here also do not observe an improved performance of GNNs compared to our baseline models.

C.3 Attention Evaluation

In Section 5.4, we observed an impact of the dataset complexity on GNN performance at different homophily values. While for the CORA dataset, which has 7 classes, low-homophily graphs always resulted in poor performance, on the TADPOLE dataset, low-homophily graphs were also able to lead to good GNN performance. Similar to Figure 4, we visualise an additional dataset in Figure 6. We attribute the relatively good performance of all models at low homophily values on the TADPOLE dataset (Figure 6b) to the learning of opposite labels for specific node features. If most of the neighbouring nodes share a different label than the one the node of interest holds, but this is consistent across the graph –the graph has a low CCNS distance–, then the network can still learn to make the correct predictions. We show this by evaluating the attention values of GAT networks of four synthetic graph structures with different homophily values. All values are summarised in the appendix in Table 15. We always report the normalised sum of all attention heads of the GAT. At homophily 0.9 (where most neighbours share the same label as the node of interest), the attention from the neighbours with the same label is the highest. On the other hand, at hom = 0.5, all nodes receive the highest attention from neighbours with class label "MCI". This makes it very difficult for the network to distinguish between nodes of different labels, and therefore to make the correct predictions. At very low homophily (hom = 0.1), the attention of the neighbours with the same label is 0, which again,

Table 11: Results of the experiments on the **CORA** dataset. BL: baselines, k: number of neighbours, Transd.: transductive learning with only self-loops. GNNs out-performing the BL are underlined, and the best performances of static and dynamic graph constructions, the highest homophily and the lowest CCNS distance are **bold**.

	Initial edges	Model	k	Test acc	Homophily ↑		CCNS distance ↓	
					Train	Test	Train	Test
BL	-	Random Forest	-	0.7788 ± 0.00	-	-	-	-
		Ridge classifier	-	0.7860 ± 0.00	-	-	-	-
		MLP	-	0.6030 ± 0.00	-	-	-	-
Transd.	Self-loops	GCN	-	0.6200 ± 0.02	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		SAGE	-	0.6396 ± 0.03	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GraphConv	-	0.6504 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GAT	-	0.6848 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
Static graph construction	Random	GCN	-	0.3068 ± 0.02	0.171 ± 0.26	0.201 ± 0.29	0.373	0.356
		SAGE	-	0.6224 ± 0.02	0.171 ± 0.26	0.201 ± 0.29	0.373	0.356
		GraphConv	-	0.5388 ± 0.03	0.171 ± 0.26	0.201 ± 0.29	0.373	0.356
		GAT	-	0.3208 ± 0.02	0.171 ± 0.26	0.201 ± 0.29	0.373	0.356
	k-NN Euclidean	GCN	20	0.7336 ± 0.01	0.498 ± 0.23	0.495 ± 0.22	0.378	0.396
		SAGE	20	0.6836 ± 0.02	0.498 ± 0.23	0.495 ± 0.22	0.378	0.396
		GraphConv	20	0.7692 ± 0.01	0.498 ± 0.23	0.495 ± 0.22	0.378	0.396
		GAT	20	0.7288 ± 0.01	0.498 ± 0.23	0.495 ± 0.22	0.378	0.396
	k-NN Cosine	GCN	20	0.7332 ± 0.01	0.537 ± 0.24	0.537 ± 0.23	0.344	0.362
		SAGE	20	0.6668 ± 0.01	0.537 ± 0.24	0.537 ± 0.23	0.344	0.362
		GraphConv	20	0.7628 ± 0.01	0.537 ± 0.24	0.537 ± 0.23	0.344	0.362
		GAT	20	0.7260 ± 0.01	0.537 ± 0.24	0.537 ± 0.23	0.344	0.362
	Orig. edges	GCN	-	0.8332 ± 0.01	**0.830 ± 0.29**	**0.860 ± 0.29**	**0.101**	**0.084**
		SAGE	-	**0.8540 ± 0.01**	**0.830 ± 0.29**	**0.860 ± 0.29**	**0.101**	**0.084**
		GraphConv	-	**0.8540 ± 0.01**	**0.830 ± 0.29**	**0.860 ± 0.29**	**0.101**	**0.084**
		GAT	-	0.8420 ± 0.00	**0.830 ± 0.29**	**0.860 ± 0.29**	**0.101**	**0.084**
Dynamic graph construction	No edges	GCN	2	0.6900 ± 0.03	**0.987 ± 0.10**	0.749 ± 0.42	0.072	0.181
		SAGE	2	0.7000 ± 0.02	0.589 ± 0.38	0.510 ± 0.37	0.232	0.267
		GraphConv	2	0.6904 ± 0.01	0.880 ± 0.21	0.769 ± 0.25	0.085	0.144
		GAT	2	0.6532 ± 0.03	0.921 ± 0.20	0.652 ± 0.43	**0.050**	0.208
	Self-loops	GCN	5	0.5932 ± 0.13	0.737 ± 0.31	0.612 ± 0.37	0.176	0.244
		SAGE	20	0.6900 ± 0.01	0.857 ± 0.23	0.751 ± 0.25	0.092	0.160
		GraphConv	2	0.7024 ± 0.01	0.696 ± 0.27	0.586 ± 0.32	0.185	0.273
		GAT	5	0.6492 ± 0.01	0.796 ± 0.28	0.584 ± 0.39	0.139	0.259
	Random	GCN	2	0.3240 ± 0.02	0.663 ± 0.28	0.230 ± 0.38	0.201	0.351
		SAGE	10	0.6960 ± 0.01	0.674 ± 0.25	0.534 ± 0.32	0.206	0.323
		GraphConv	2	0.7052 ± 0.01	0.831 ± 0.24	0.719 ± 0.25	0.101	0.180
		GAT	10	0.4252 ± 0.02	0.405 ± 0.23	0.252 ± 0.23	0.436	0.544
	k-NN Euclidean	GCN	5	0.7192 ± 0.01	0.581 ± 0.31	0.533 ± 0.30	0.314	0.363
		SAGE	5	0.7264 ± 0.01	0.838 ± 0.23	0.676 ± 0.35	0.097	0.222
		GraphConv	5	0.7284 ± 0.01	0.884 ± 0.21	0.801 ± 0.24	0.073	**0.129**
		GAT	20	0.6388 ± 0.06	0.419 ± 0.27	0.415 ± 0.28	0.429	0.446
	k-NN Cosine	GCN	5	0.7424 ± 0.00	0.611 ± 0.33	0.570 ± 0.32	0.299	0.349
		SAGE	5	0.7216 ± 0.01	0.774 ± 0.26	0.663 ± 0.35	0.153	0.234
		GraphConv	5	0.7304 ± 0.01	0.890 ± 0.21	0.778 ± 0.25	0.070	0.143
		GAT	20	0.6716 ± 0.01	0.662 ± 0.30	0.634 ± 0.37	0.216	0.236
	Orig. edges	GCN	20	**0.8372 ± 0.01**	0.861 ± 0.24	**0.813 ± 0.31**	0.086	0.133
		SAGE	10	0.7832 ± 0.01	0.958 ± 0.10	0.780 ± 0.32	0.019	0.138
		GraphConv	2	0.7576 ± 0.02	0.819 ± 0.25	0.780 ± 0.29	0.115	0.149
		GAT	2	0.8388 ± 0.04	0.885 ± 0.21	0.807 ± 0.29	0.071	0.131

makes it possible for the network to distinguish nodes by their neighbourhood, enabling correct predictions. Three examples of 2-hop neighbourhoods at the different homophily values are visualised in Figure 7. The label is indicated by the node colour and the distance between two nodes indicates the attention value of this edge. While at hom = 0.9, most neighbours share the same label, at a low homophily value of 0.1 (c), most neighbours have a different label and the attention values are similar across them. At an in-between homophily of 0.4, several nodes share the same label, while others do not.

Table 12: Results of the experiments on the **TADPOLE** dataset. BL: baselines, k: number of neighbours, Transd.: transductive learning with only self-loops. Overall, the best performance for static and dynamic graph construction is underlined, the best performance for static and dynamic graph construction, highest homophily and lowest DNNS distance are **bold**.

	Initial edges	**Model**	k	**Test acc ↑**	**Homophily ↑**		**CCNS distance ↓**	
					Train	Test	Train	Test
BL	-	Majority vote	-	0.5674 ± 0.00	-	-	-	-
	-	Random forest	-	**0.9474 ± 0.00**	-	-	-	-
	-	Logistic regression	-	0.8597 ± 0.00	-	-	-	-
Transd.	Self-loops	GCN	-	0.9018 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		SAGE	-	0.8772 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GraphConv	-	0.8912 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GAT	-	0.6386 ± 0.07	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
Static graph construction	Random	GCN	-	0.7965 ± 0.04	0.457 ± 0.49	0.426 ± 0.49	0.350	0.348
		SAGE	-	0.8877 ± 0.01	0.457 ± 0.49	0.426 ± 0.49	0.350	0.348
		GraphConv	-	0.8842 ± 0.01	0.457 ± 0.49	0.426 ± 0.49	0.350	0.348
		GAT	-	0.7930 ± 0.04	0.457 ± 0.49	0.426 ± 0.49	0.350	0.348
	k-NN Euclidean	GCN	5	0.7439 ± 0.03	0.754 ± 0.23	0.775 ± 0.24	0.283	0.213
		SAGE	5	0.8982 ± 0.03	0.754 ± 0.23	0.775 ± 0.24	0.283	0.213
		GraphConv	5	0.9088 ± 0.01	0.754 ± 0.23	0.775 ± 0.24	0.283	0.213
		GAT	2	0.7895 ± 0.04	**0.857 ± 0.23**	**0.904 ± 0.20**	**0.184**	**0.094**
	k-NN Cosine	GCN	5	0.7789 ± 0.02	0.760 ± 0.23	0.754 ± 0.25	0.276	0.221
		SAGE	5	0.8877 ± 0.02	0.760 ± 0.23	0.754 ± 0.25	0.276	0.221
		GraphConv	5	**0.9333 ± 0.01**	0.760 ± 0.23	0.754 ± 0.25	0.276	0.221
		GAT	2	0.8105 ± 0.02	0.855 ± 0.23	0.895 ± 0.21	0.192	0.105
Dynamic graph construction	No edges	GCN	20	0.9263 ± 0.03	0.899 ± 0.19	0.919 ± 0.19	0.143	**0.073**
		SAGE	20	0.9053 ± 0.02	0.867 ± 0.20	0.806 ± 0.21	0.183	0.183
		GraphConv	2	0.9228 ± 0.02	0.919 ± 0.18	0.798 ± 0.34	**0.107**	0.190
		GAT	20	0.9018 ± 0.06	0.739 ± 0.24	0.908 ± 0.15	0.280	0.101
	Self-loops	GCN	10	0.9298 ± 0.02	0.891 ± 0.21	0.902 ± 0.16	0.150	0.085
		SAGE	5	0.9088 ± 0.02	0.900 ± 0.19	0.614 ± 0.29	0.140	0.441
		GraphConv	5	0.9228 ± 0.02	**0.920 ± 0.18**	**0.937 ± 0.15**	0.113	0.051
		GAT	20	0.9123 ± 0.05	0.826 ± 0.24	0.784 ± 0.21	0.236	0.204
	Random	GCN	2	0.8421 ± 0.06	0.912 ± 0.20	0.851 ± 0.27	0.132	0.177
		SAGE	10	0.9228 ± 0.02	0.834 ± 0.23	0.423 ± 0.22	0.205	0.616
		GraphConv	5	0.8947 ± 0.03	0.775 ± 0.24	0.411 ± 0.25	0.273	0.594
		GAT	5	0.8632 ± 0.02	0.903 ± 0.20	0.895 ± 0.20	0.145	0.119
	k-NN Euclidean	GCN	2	0.9333 ± 0.01	0.811 ± 0.25	0.793 ± 0.28	0.229	0.204
		SAGE	20	0.9368 ± 0.01	0.896 ± 0.19	0.461 ± 0.63	0.138	0.632
		GraphConv	10	0.8947 ± 0.02	0.736 ± 0.23	0.777 ± 0.29	0.302	0.219
		GAT	10	0.9123 ± 0.03	0.826 ± 0.24	0.775 ± 0.29	0.223	0.206
	k-NN Cosine	GCN	2	0.8421 ± 0.02	0.833 ± 0.24	0.786 ± 0.30	0.210	0.199
		SAGE	20	**0.9404 ± 0.02**	0.822 ± 0.23	0.899 ± 0.21	0.220	0.084
		GraphConv	10	0.8982 ± 0.02	0.740 ± 0.25	0.761 ± 0.28	0.304	0.213
		GAT	10	0.8316 ± 0.04	0.846 ± 0.23	0.828 ± 0.27	0.201	0.187

Table 13: Results of the experiments on the **UKBB Brain Age** dataset. BL: baselines, k: number of neighbours, Transd.: transductive training with only self-loops. The best performance and highest homophily for static and dynamic graph construction are **bold**. For all methods, homophily is evaluated on the train and test set.

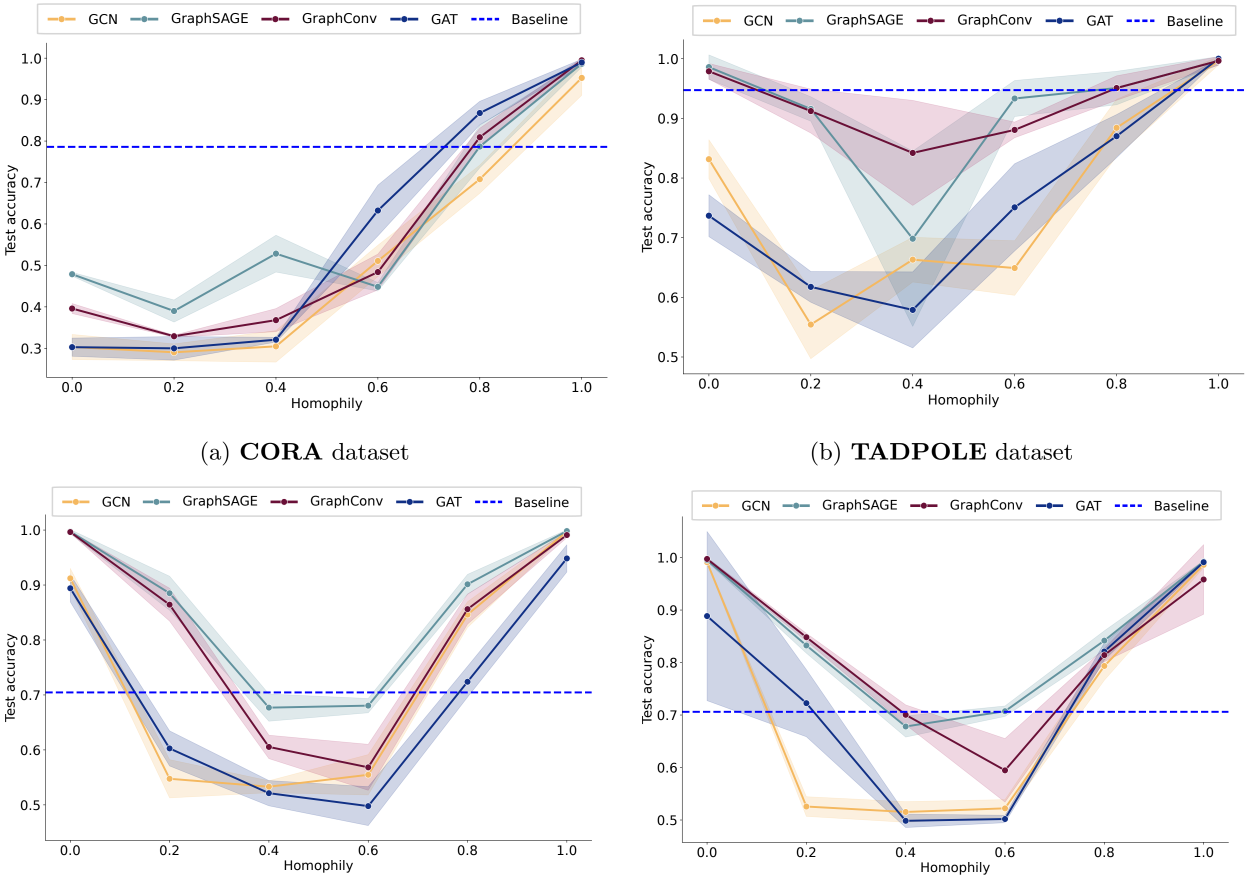
	Initial edges	**Model**	k	**Test MAE ↓**	**Homophily ↑**	
					Train	Test
BL	-	Mean prediction	-	6.4090 ± 0.00	-	-
	-	Random Forest	-	4.1424 ± 0.01	-	-
	-	Linear Regression	-	3.7545 ± 0.00	-	-
Transd.	Self-loops	GCN	-	4.0236 ± 0.12	1.000 ± 0.00	1.000 ± 0.00
		SAGE	-	4.0339 ± 0.05	1.000 ± 0.00	1.000 ± 0.00
		GraphConv	-	3.9750 ± 0.06	1.000 ± 0.00	1.000 ± 0.00
		GAT		3.9477 ± 0.04	1.000 ± 0.00	1.000 ± 0.00
Static graph construction	Random	GCN	-	6.2158 ± 0.07	0.750 ± 0.10	0.742 ± 0.10
		SAGE	-	**3.8764 ± 0.08**	0.750 ± 0.10	0.742 ± 0.10
		GraphConv	-	4.2029 ± 0.16	0.750 ± 0.10	0.742 ± 0.10
		GAT	-	6.4034 ± 0.07	0.750 ± 0.10	0.742 ± 0.10
	k-NN Euclidean	GCN	2	4.3351 ± 0.07	**0.915 ± 0.07**	**0.916 ± 0.07**
		SAGE	10	4.1780 ± 0.17	0.843 ± 0.06	0.844 ± 0.06
		GraphConv	2	4.1979 ± 0.04	**0.915 ± 0.07**	**0.916 ± 0.07**
		GAT	20	4.2888 ± 0.01	0.832 ± 0.06	0.834 ± 0.06
	k-NN Cosine	GCN	2	4.3808 ± 0.08	**0.915 ± 0.07**	**0.919 ± 0.06**
		SAGE	10	4.2302 ± 0.21	0.843 ± 0.06	0.844 ± 0.06
		GraphConv	2	4.2260 ± 0.06	**0.915 ± 0.07**	**0.919 ± 0.06**
		GAT	20	4.3182 ± 0.03	0.833 ± 0.06	0.833 ± 0.06
Dynamic graph construction	No edges	GCN	2	4.0257 ± 0.06	**0.886 ± 0.09**	**0.865 ± 0.10**
		SAGE	5	3.8882 ± 0.03	0.752 ± 0.10	0.754 ± 0.10
		GraphConv	5	3.9741 ± 0.05	0.845 ± 0.08	0.840 ± 0.08
		GAT	2	4.1071 ± 0.07	0.840 ± 0.10	0.843 ± 0.11
	Self-loops	GCN	2	3.9869 ± 0.06	0.844 ± 0.10	0.841 ± 0.10
		SAGE	20	3.9496 ± 0.16	0.781 ± 0.07	0.780 ± 0.08
		GraphConv	20	3.9422 ± 0.13	0.849 ± 0.06	0.845 ± 0.07
		GAT	2	4.0825 ± 0.07	0.844 ± 0.10	0.839 ± 0.10
	Random	GCN	2	5.1712 ± 0.20	0.837 ± 0.12	0.834 ± 0.13
		SAGE	10	**3.8811 ± 0.04**	0.769 ± 0.08	0.780 ± 0.09
		GraphConv	10	4.1248 ± 0.30	0.770 ± 0.08	0.768 ± 0.09
		GAT	2	5.7138 ± 0.10	0.852 ± 0.11	0.831 ± 0.14
	k-NN Euclidean	GCN	2	4.1109 ± 0.07	0.835 ± 0.10	0.849 ± 0.11
		SAGE	20	3.9226 ± 0.13	0.845 ± 0.06	0.842 ± 0.07
		GraphConv	2	3.9560 ± 0.09	0.843 ± 0.11	0.831 ± 0.11
		GAT	2	4.1603 ± 0.04	0.835 ± 0.10	0.837 ± 0.11
	k-NN Cosine	GCN	2	4.0975 ± 0.05	0.839 ± 0.10	0.844 ± 0.10
		SAGE	20	3.9353 ± 0.12	0.837 ± 0.06	0.837 ± 0.07
		GraphConv	2	4.0181 ± 0.13	0.848 ± 0.11	0.852 ± 0.10
		GAT	2	4.1927 ± 0.04	0.833 ± 0.10	0.835 ± 0.10

Table 14: Results of the experiments on the **UKBB Cardiac** dataset. BL: baselines, k: number of neighbours, Transd.: transductive training with only self-loops, GC: graph construction. GNNs out-performing the baselines are underlined, and the best performances of static and dynamic graph constructions are **bold**.

	Initial edges	Model	k	Test accuracy	Homophily ↑		CCNS distance ↓	
					Train	Test	Train	Test
BL	-	Majority vote	-	0.5000 ± 0.00	-	-	-	-
	-	Random Forest	-	0.7027 ± 0.00	-	-	-	-
	-	Linear Regression	-	0.6916 ± 0.00	-	-	-	-
Transd.	Self-loops	GCN	-	**0.6816 ± 0.01**	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		SAGE	-	0.5920 ± 0.08	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GraphConv	-	0.6724 ± 0.02	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
		GAT	-	0.6812 ± 0.01	1.000 ± 0.00	1.000 ± 0.00	0.000	0.000
Static GC	Random	GCN	-	0.5019 ± 0.03	0.504 ± 0.34	0.480 ± 0.34	0.500	0.499
		SAGE	-	0.6824 ± 0.02	0.504 ± 0.34	0.480 ± 0.34	0.500	0.499
		GraphConv	-	0.5169 ± 0.03	0.504 ± 0.34	0.480 ± 0.34	0.500	0.499
		GAT	-	0.5291 ± 0.02	0.504 ± 0.34	0.480 ± 0.34	0.500	0.499
	k-NN Euclidean	GCN	10	0.6632 ± 0.01	0.590 ± 0.19	0.605 ± 0.19	0.477	0.467
		SAGE	2	0.6498 ± 0.01	**0.778 ± 0.25**	**0.786 ± 0.25**	**0.345**	**0.335**
		GraphConv	5	0.6686 ± 0.01	0.640 ± 0.22	0.645 ± 0.23	0.449	0.447
		GAT	20	0.6322 ± 0.03	0.563 ± 0.16	0.572 ± 0.15	0.488	0.483
	k-NN Cosine	GCN	20	0.6517 ± 0.00	0.564 ± 0.16	0.576 ± 0.15	0.487	0.482
		SAGE	10	0.6510 ± 0.03	0.595 ± 0.19	0.601 ± 0.19	0.474	0.470
		GraphConv	10	0.6563 ± 0.02	0.595 ± 0.19	0.601 ± 0.19	0.474	0.470
Dynamic GC	No edges	GCN	2	0.6816 ± 0.01	0.644 ± 0.36	0.627 ± 0.37	0.458	0.468
		SAGE	20	0.6379 ± 0.02	0.599 ± 0.18	0.572 ± 0.19	0.489	0.484
		GraphConv	2	0.6215 ± 0.06	0.606 ± 0.37	0.636 ± 0.36	0.478	0.463
		GAT	2	0.6839 ± 0.00	0.615 ± 0.38	0.616 ± 0.38	0.474	0.473
	Self-loops	GCN	2	0.6521 ± 0.03	0.622 ± 0.37	0.602 ± 0.36	0.470	0.479
		SAGE	20	0.6444 ± 0.01	0.617 ± 0.20	0.606 ± 0.21	0.461	0.468
		GraphConv	2	0.6659 ± 0.02	0.718 ± 0.33	0.652 ± 0.36	0.405	0.454
		GAT	2	0.6812 ± 0.01	0.636 ± 0.37	0.634 ± 0.38	0.463	0.464
	Random	GCN	2	0.6360 ± 0.02	0.551 ± 0.37	0.538 ± 0.37	0.495	0.497
		SAGE	10	0.6678 ± 0.01	0.508 ± 0.16	0.556 ± 0.19	0.500	0.490
		GraphConv	2	0.6563 ± 0.02	0.542 ± 0.36	0.526 ± 0.36	0.496	0.499
		GAT	10	0.6510 ± 0.04	0.520 ± 0.18	0.516 ± 0.16	0.499	0.499
	k-NN Euclidean	GCN	2	0.6781 ± 0.01	0.611 ± 0.37	0.612 ± 0.36	0.475	0.475
		SAGE	10	**0.6970 ± 0.02**	0.499 ± 0.11	0.507 ± 0.12	0.500	0.500
		GraphConv	2	0.6860 ± 0.02	0.678 ± 0.35	0.614 ± 0.38	0.436	0.474
		GAT	5	0.6690 ± 0.03	0.541 ± 0.25	0.554 ± 0.25	0.495	0.493
	k-NN Cosine	GCN	2	0.6770 ± 0.00	0.607 ± 0.37	0.595 ± 0.37	0.477	0.482
		SAGE	10	0.6659 ± 0.03	0.682 ± 0.24	0.677 ± 0.25	0.421	0.426
		GraphConv	2	0.6862 ± 0.01	**0.767 ± 0.25**	**0.714 ± 0.30**	**0.357**	**0.409**
		GAT	2	0.6736 ± 0.01	0.589 ± 0.37	0.588 ± 0.37	0.484	0.484

Table 15: Mean and standard deviation of normalised attention values from all neighbours with respective labels of a graph structure with high and low homophily. The highest attention values for each node label class are highlighted in bold. NC: normal control, MCI: mild cognitive impairment, AD: Alzheimer's disease.

Homophily	Node label	Attention from NC	Attention from MCI	Attention from AD
0.9	NC	**1.919 ± 1.08**	0.532 ± 0.56	0.091 ± 0.21
	MCI	0.198 ± 0.31	**1.881 ± 1.06**	0.083 ± 0.22
	AD	0.158 ± 0.29	0.777 ± 0.66	**1.961 ± 1.14**
0.4	NC	0.978 ± 0.75	**2.002 ± 1.05**	0.255 ± 0.34
	MCI	0.556 ± 0.59	**0.972 ± 0.74**	0.243 ± 0.36
	AD	0.676 ± 0.68	**1.743 ± 0.97**	0.940 ± 0.71
0.1	NC	0.000 ± 0.00	**3.106 ± 1.47**	0.415 ± 0.48
	MCI	**0.985 ± 0.74**	0.000 ± 0.00	0.461 ± 0.57
	AD	1.038 ± 0.88	**3.013 ± 1.35**	0.000 ± 0.00



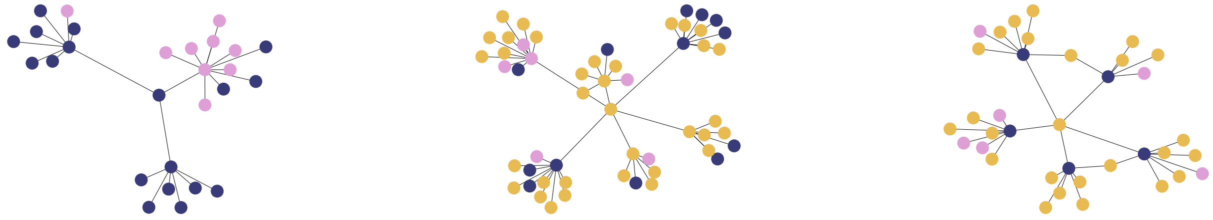
(a) **CORA** dataset

(b) **TADPOLE** dataset

(c) **ABIDE** dataset

(d) **UKBB Cardiac** dataset

Figure 6: Performance of different graph convolutions on synthetic graph structures with different homophily values on (a) the **CORA** dataset, (b) the **TADPOLE** dataset, (c) the ABIDE dataset, and (d) the UKBB cardiac dataset. The dashed blue line indicates the mean performance of the best baseline for each dataset.



(a) Homophily 0.9

(b) Homophily 0.4

(c) Homophily 0.1

Figure 7: **Visualisation of attention-based neighbourhoods** of a random node (centre node) from the TADPOLE dataset with synthetically generated graph structures and its two-hop neighbourhood. The node colours indicate node labels and the distance is proportional to the summed attention weight of the edges to the respective neighbouring node.

Supplementary Material: Differentially Private Graph Neural Networks for Whole-Graph Classification

SUPPLEMENTARY MATERIAL

APPENDIX A DATASETS

Here, we provide some additional information about the datasets that were used for the experiments presented in this work.

A.1 Synthetic Dataset

We generate $1,000$ individual Erdős-Rényi graphs, where 500 graphs belong to each of the two classes that define the binary classification problem. Each individual graph consists of twenty nodes and each node contains nine features, which are sampled from a normal distribution with the mean values of 0 and 0.1, while having the same standard deviation of 0.5. The edge connection probabilities of the graphs from the two classes are set to 0.2 and 0.3, respectively. We split the dataset into 600 training samples, 100 validation samples and 300 test samples, and perform binary graph classification.

A.2 ECG Dataset

LBBB is an insidious type of arrhythmia (that is, anomaly in the conduction system of the heart), which, when appearing suddenly, can herald acute myocardial ischemia (a lack of oxygenation of the heart muscle) or infarction. The ECG dataset we use contains ECG data of 1125 subjects.

APPENDIX B EXECUTION ENVIRONMENT

The experiments were executed on a Linux machine with a Quadro RTX 8000 GPU and the code was implemented in PyTorch [70] version 1.12.0, Pytorch Geometric [71] version 2.0.4 and Functorch [72] version 0.2.0.

APPENDIX C MODEL ARCHITECTURES AND TRAINING PARAMETERS

We here provide a brief overview of the model architectures we used for our experiments as well all corresponding hyperparameters. For each dataset, we utilised a different model architecture which we determined through hyperparameter searches. Table 4 summarises the hyperparameters used for each experiment. For SGD training on the *Molbace* dataset we used a cyclic learning rate scheduler with a defined lower and upper learning rate as described in Table 4. We did not observe increased performance when applying the cyclic learning rate scheduler to DP-SGD training on this dataset, which is why we did not utilise a learning rate scheduler for those applications. All models are implemented using *PyTorch* [73] and *PyTorch Geometric* [71]. For all binary models we used the Binary Cross Entropy Loss, for the non-binary classification task of the *Fingerprint* dataset we use the Cross Entropy Loss.

C.1 Synthetic Dataset

For the *synthetic* dataset we use a Neural Network (NN) with four layers of the corresponding graph convolutions (GCN, GAT, or GraphSAGE), followed by a Mean Pooling layer and two linear layers. Each convolutional layer, as well as the first linear layer, is followed by a ReLU activation function. The hidden channels of the graph convolutional layers are $200, 400, 800, 1600$ and the fully connected layers at the end of the network have 265 hidden channels. For non-DP training, we used a learning rate of 0.05 for the GCN network, 0.1 for GAT and 0.08 for GraphSAGE and a batch size of 24. For all DP-SGD training runs we use an L_2 clipping bound of 3.0 and noise multipliers in $\{1.0, 2.0, 2.2\}$.

C.2 Fingerprint Dataset

For the experiments on the *Fingerprint* dataset we use a NN architecture with an Instance Normalisation layer followed by three layers of the corresponding graph convolution (GCN, GAT, or GraphSAGE), a Max Pooling layer and two linear layers. Each convolutional layer, as well as the first linear layer, is followed by a ReLU activation function. The hidden channels of the graph convolutional layers are $256, 512, 1024$ and the fully connected layers at the end of the network have 265 hidden channels. For the non-DP training we used a learning rate of 0.08 for GCN, GAT and GraphSAGE networks and a batch size of 64. DP-SGD training was performed with different noise multipliers in $\{1.0, 1.8, 2.3\}$ and the same L_2 Clip of 3.0. The learning rates differ with the type of graph convolution and can be found in Table 4.

C.3 ECG Dataset

The experiments on the *ECG* Dataset were performed with a model architecture consisting of two graph convolutional layers, followed by a Max Pooling layer and three linear layers. Each convolutional layer and each linear layer is followed by a ReLU activation function. The hidden channels of the graph convolutional layers are $256, 512$ and of the linear layers $128, 56, 24$. In the non-DP training all convolutional layers and the Max Pooling layer are followed by a Dropout Layer with dropout probability 0.2. We removed all Dropout Layers for the DP-SGD training, because the added noise intrinsic to the algorithm already functions as regularisation. Learning rate and batch size of the SGD training were set to 0.05 and 24, correspondingly for all models. The learning rates for DP-SGD training runs depends on the graph convolution and can be found in Table 4.

C.4 Molbace Dataset

For the experiments on the *Molbace* dataset we utilise a NN with an Instance Normalisation layer, followed by three graph convolution layers, each followed by another Instance Normalisation layer and a ReLU activation function. The convolutional layers are followed by a Mean Pooling layer and two linear layers with 512 hidden channels. For the non-DP training we use a Batch Size of 64 and a cyclic learning rate scheduler with upper and lower learning rates noted in Table 4. The learning rates were determined using a learning rate finder. For the DP-SGD training we use a Batch Size of

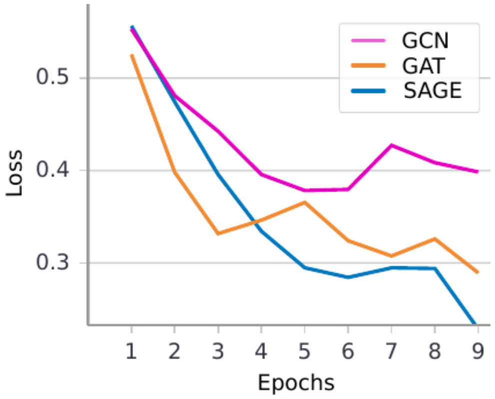


Fig. 7: Training loss curves of the three GNN models on the *ECG* dataset using DP-SGD with a noise multiplier of 0.6 and an L_2-clip of 5.0, where $\varepsilon = 10$.

24 and a learning rate of 0.1 for all models. We did not use a learning rate scheduler for DP-SGD training, since it did not improve model performance.

C.5 Organ Meshes Dataset

To train the organ mesh classification we utilise a NN with three graph convolutional layers that are either GCN, GraphSAGE, or Cheb convolutions. The Leaky ReLU activation function was used as well as a dropout layer with probability 0.3 after each non-linearity. Global mean pooling was used as a readout layer to unify the graph representation into one representation for the whole graph. The batch size was set to 24 for all experiments on this dataset. The specific learning rates for each model and DP and non-DP setting can be found in Table 4. We used chebyshev spectral graph convolutions (Cheb) instead of graph attention (GAT) layers for this dataset, due to limited resources and the fact that GAT layers need more memory for multiple attention heads.

Appendix D
Performances of Different Model Architectures

The comparable performance of the different GNN architectures we use for our experiments (GCN, GAT, Cheb, and GraphSAGE) leads to the conclusion that DP-SGD training is independent from the type of graph convolution. For all datasets, we report similar performance of all models and we show two examples of respective training loss curves in Figure 7 and 8. Figure 7 shows the training loss curves on the *ECG* dataset for the DP-SGD training with noise multiplier 0.6 and L_2 Clip 5.0.

Appendix E
Explainability using GNNExplainer

As noted in section 5, we applied the explainability technique GNNExplainer [74] to our trained networks. Figure 9 visualises four more examples of the original graph (**A**) in blue, the output of the GNNExplainer from the "normal" SGD training (**B**) in orange and the graph resulting from the GNNExplainer and the DP-SGD training (**C**) in red. All figures were created using our GCN model on the *Fingerprint* dataset. The DP model was trained with a privacy budget of $\varepsilon = 5$. In the second example, **B.2** and **C.2** are equal, indicating that in both SGD and DP-SGD training the same graph edges are considered most relevant. In the other three examples in Figure 9 there are minute differences in the GNNExplainer graphs, showing that slightly different edges have most impact on the model's prediction.

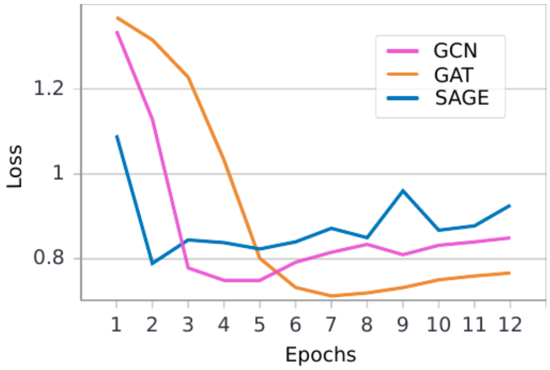


Fig. 8: Training loss curves of the three GNN models on the *Fingerprint* dataset using DP-SGD with a noise multiplier of 1.0 and an L_2-clip of 3.0, where $\varepsilon = 5$.

Dataset	Model	Optimiser	Learning Rate	Batch Size	Scheduler
Synthetic	GCN	SGD	0.05	24	-
		DP-SGD	0.1	24	-
	GAT	SGD	0.1	24	-
		DP-SGD	0.4	24	-
	SAGE	SGD	0.04	24	-
		DP-SGD	0.2	24	-
Fingerprints	GCN	SGD	0.08	64	-
		DP-SGD	0.2	64	-
	GAT	SGD	0.08	64	-
		DP-SGD	0.2	64	-
	SAGE	SGD	0.08	64	-
		DP-SGD	0.1	64	-
ECG	GCN	SGD	0.05	24	-
		DP-SGD	0.12	24	-
	GAT	SGD	0.05	24	-
		DP-SGD	0.15	24	-
	SAGE	SGD	0.05	24	-
		DP-SGD	0.1	24	-
Molbace	GCN	SGD	$7e-3$ to $7e-2$	64	cyclic
		DP-SGD	0.1	24	-
	GAT	SGD	$5e-3$ to $5e-2$	64	cyclic
		DP-SGD	0.1	24	-
	SAGE	SGD	$7e-3$ to $7e-2$	64	cyclic
		DP-SGD	0.1	24	-
Organ Meshes	GCN	SGD	0.0712	24	-
		DP-SGD ($\varepsilon = 1.0$)	0.07134	25	-
		DP-SGD ($\varepsilon = 0.5$)	0.0748	25	-
	Cheb	SGD	0.0923	24	-
		DP-SGD ($\varepsilon = 1.0$)	0.0394	24	-
		DP-SGD ($\varepsilon = 0.5$)	0.05495	24	-
	SAGE	SGD	0.0765	24	-
		DP-SGD ($\varepsilon = 1.0$)	0.07134	24	-
		DP-SGD ($\varepsilon = 0.5$)	0.02735	24	-

TABLE 4: Overview of hyperparameters for all datasets and experiments. We performed manual, grid search, and random search hyperparameter tuning.

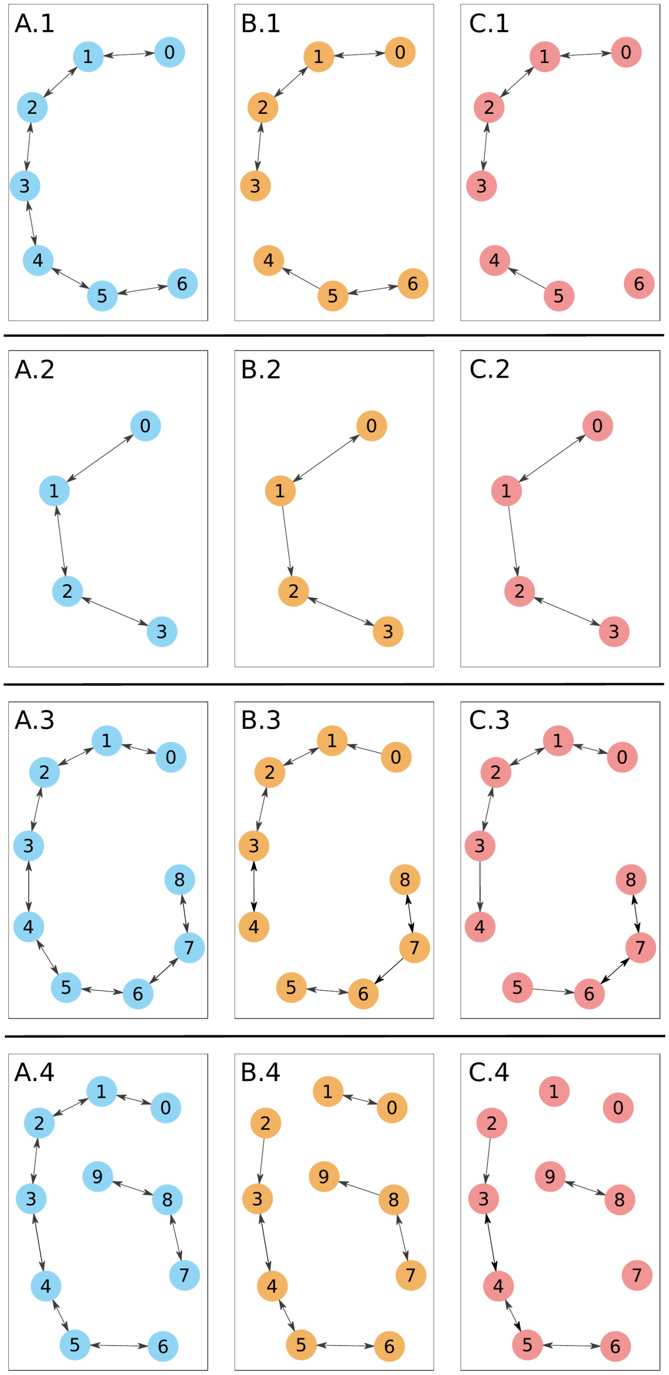
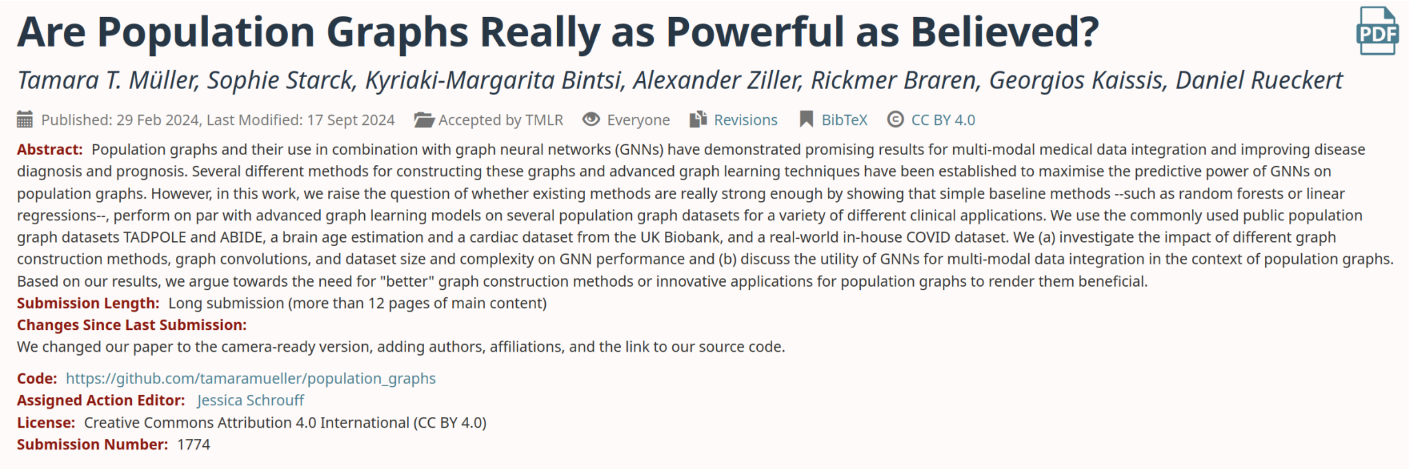


Fig. 9: Visualisation of four GNNExplainer examples on the fingerprint dataset. The original graph (**A**) is shown in blue, the resulting graph from the GNNExplainer and the model trained with SGD in orange (**B**) and with DP-SGD in red (**C**). Both models were trained on the *Fingerprint* dataset. The DP-SGD training was performed with a privacy budget of $\varepsilon = 5$.

Licences



Are Population Graphs Really as Powerful as Believed?

Tamara T. Müller, Sophie Starck, Kyriaki-Margarita Bintsi, Alexander Ziller, Rickmer Braren, Georgios Kaissis, Daniel Rueckert

Published: 29 Feb 2024, Last Modified: 17 Sept 2024 Accepted by TMLR Everyone Revisions BibTeX CC BY 4.0

Abstract: Population graphs and their use in combination with graph neural networks (GNNs) have demonstrated promising results for multi-modal medical data integration and improving disease diagnosis and prognosis. Several different methods for constructing these graphs and advanced graph learning techniques have been established to maximise the predictive power of GNNs on population graphs. However, in this work, we raise the question of whether existing methods are really strong enough by showing that simple baseline methods --such as random forests or linear regressions--, perform on par with advanced graph learning models on several population graph datasets for a variety of different clinical applications. We use the commonly used public population graph datasets TADPOLE and ABIDE, a brain age estimation and a cardiac dataset from the UK Biobank, and a real-world in-house COVID dataset. We (a) investigate the impact of different graph construction methods, graph convolutions, and dataset size and complexity on GNN performance and (b) discuss the utility of GNNs for multi-modal data integration in the context of population graphs. Based on our results, we argue towards the need for "better" graph construction methods or innovative applications for population graphs to render them beneficial.

Submission Length: Long submission (more than 12 pages of main content)

Changes Since Last Submission:

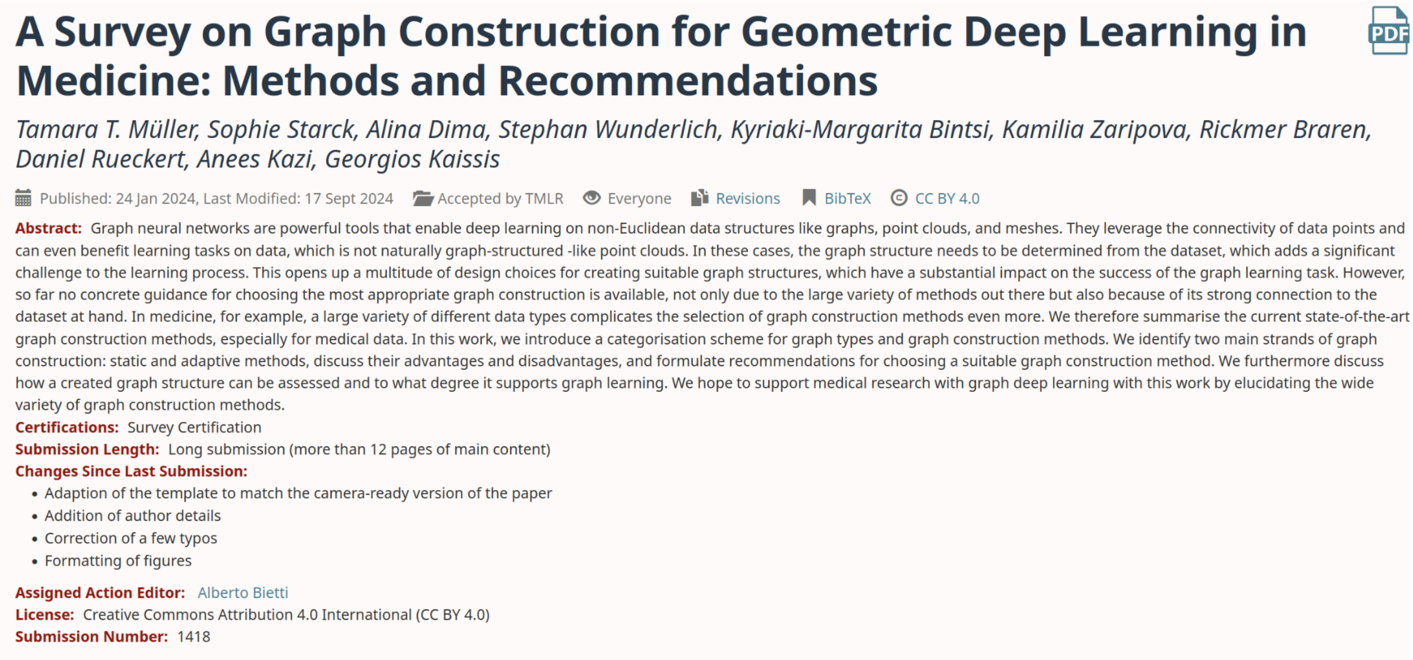
We changed our paper to the camera-ready version, adding authors, affiliations, and the link to our source code.

Code: https://github.com/tamaramueller/population_graphs

Assigned Action Editor: Jessica Schrouff

License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Submission Number: 1774



A Survey on Graph Construction for Geometric Deep Learning in Medicine: Methods and Recommendations

Tamara T. Müller, Sophie Starck, Alina Dima, Stephan Wunderlich, Kyriaki-Margarita Bintsi, Kamilia Zaripova, Rickmer Braren, Daniel Rueckert, Anees Kazi, Georgios Kaissis

Published: 24 Jan 2024, Last Modified: 17 Sept 2024 Accepted by TMLR Everyone Revisions BibTeX CC BY 4.0

Abstract: Graph neural networks are powerful tools that enable deep learning on non-Euclidean data structures like graphs, point clouds, and meshes. They leverage the connectivity of data points and can even benefit learning tasks on data, which is not naturally graph-structured -like point clouds. In these cases, the graph structure needs to be determined from the dataset, which adds a significant challenge to the learning process. This opens up a multitude of design choices for creating suitable graph structures, which have a substantial impact on the success of the graph learning task. However, so far no concrete guidance for choosing the most appropriate graph construction is available, not only due to the large variety of methods out there but also because of its strong connection to the dataset at hand. In medicine, for example, a large variety of different data types complicates the selection of graph construction methods even more. We therefore summarise the current state-of-the-art graph construction methods, especially for medical data. In this work, we introduce a categorisation scheme for graph types and graph construction methods. We identify two main strands of graph construction: static and adaptive methods, discuss their advantages and disadvantages, and formulate recommendations for choosing a suitable graph construction method. We furthermore discuss how a created graph structure can be assessed and to what degree it supports graph learning. We hope to support medical research with graph deep learning with this work by elucidating the wide variety of graph construction methods.

Certifications: Survey Certification

Submission Length: Long submission (more than 12 pages of main content)

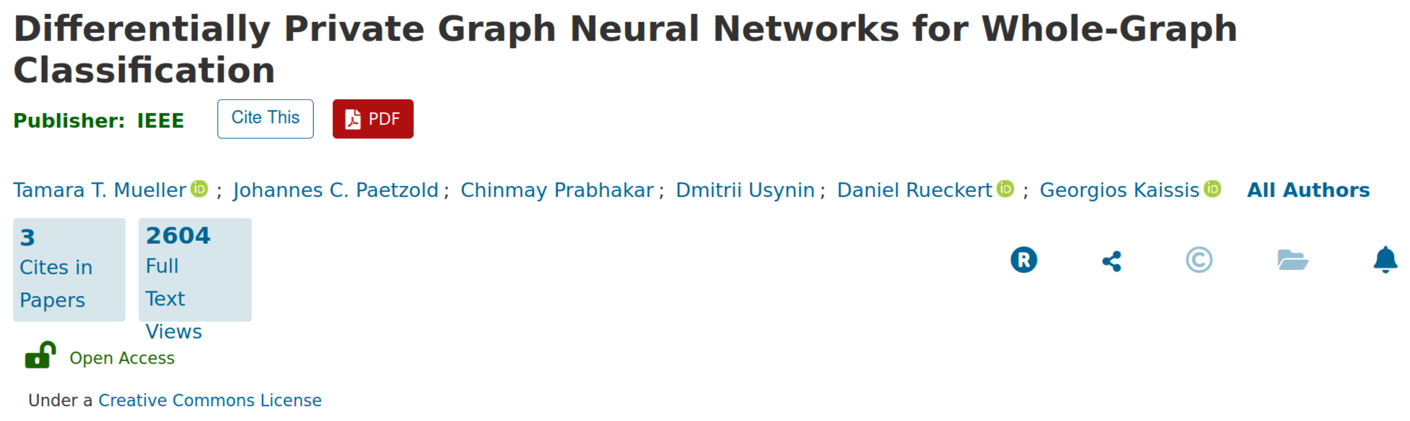
Changes Since Last Submission:

- Adaption of the template to match the camera-ready version of the paper
- Addition of author details
- Correction of a few typos
- Formatting of figures

Assigned Action Editor: Alberto Bietti

License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Submission Number: 1418



Differentially Private Graph Neural Networks for Whole-Graph Classification

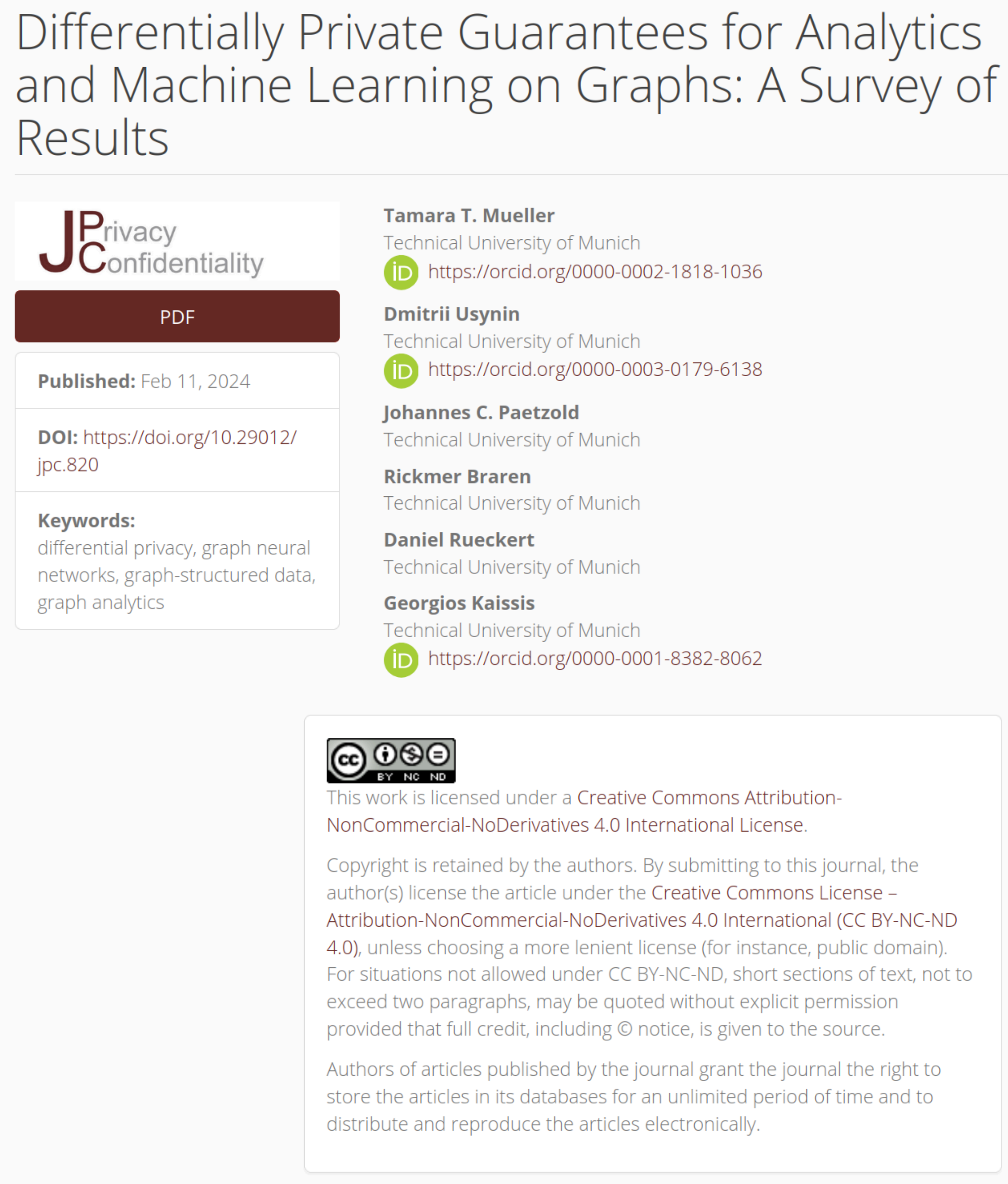
Publisher: IEEE Cite This PDF

Tamara T. Mueller ; Johannes C. Paetzold ; Chinmay Prabhakar ; Dmitrii Usynin ; Daniel Rueckert ; Georgios Kaissis **All Authors**

3 Cites in Papers 2604 Full Text Views

Open Access

Under a Creative Commons License



Differentially Private Guarantees for Analytics and Machine Learning on Graphs: A Survey of Results

JPrivacy Confidentiality

PDF

Published: Feb 11, 2024

DOI: https://doi.org/10.29012/jpc.820

Keywords:
differential privacy, graph neural networks, graph-structured data, graph analytics

Tamara T. Mueller
Technical University of Munich
https://orcid.org/0000-0002-1818-1036

Dmitrii Usynin
Technical University of Munich
https://orcid.org/0000-0003-0179-6138

Johannes C. Paetzold
Technical University of Munich

Rickmer Braren
Technical University of Munich

Daniel Rueckert
Technical University of Munich

Georgios Kaissis
Technical University of Munich
https://orcid.org/0000-0001-8382-8062

English Search Donate Explore CC

MENU

CC BY 4.0

ATTRIBUTION 4.0 INTERNATIONAL

Deed

Canonical URL : https://creativecommons.org/licenses/by/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format for any purpose, even commercially.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Oct 29, 2024

This Agreement between Mrs. Tamara Mueller ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5778040568029
License date	Apr 29, 2024
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Body Fat Estimation from Surface Meshes Using Graph Neural Networks
Licensed Content Author	Tamara T. Mueller, Siyu Zhou, Sophie Starck et al
Licensed Content Date	Jan 1, 2023
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes
Title of new work	Graph Deep Learning in Medicine - Prospects, Pitfalls, and Privacy

Institution name	Technical University of Munich
Expected presentation date	Apr 2024
Requestor Location	Mrs. Tamara Mueller Institute for AI in Medicine Technical University of Munich Einsteinstr. 25 Munchen, other 81675 Germany Attn: Mrs. Tamara Mueller
Billing Type	Invoice
Billing Address	Mrs. Tamara Mueller Institute for AI in Medicine Technical University of Munich Einsteinstr. 25 Munchen, Germany 81675 Attn: Tamara Mueller
Total	0.00 EUR

Terms and Conditions

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/ magazine	Lifetime of journal/magazine
Reuse in a book/ textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/ poster. Note: publication whether electronic or in print of presentation/ slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/ CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/ classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/ tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/ romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted

hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

Other Conditions:

Version 1.4 - Dec 2022

Questions? customercare@copyright.com.

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Oct 29, 2024

This Agreement between Mrs. Tamara Mueller ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5778040602373
License date	Apr 29, 2024
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Extended Graph Assessment Metrics for Regression and Weighted Graphs
Licensed Content Author	Tamara T. Mueller, Sophie Starck, Leonhard F. Feiner et al
Licensed Content Date	Jan 1, 2024
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes
Title of new work	Graph Deep Learning in Medicine - Prospects, Pitfalls, and Privacy

Institution name	Technical University of Munich
Expected presentation date	Apr 2024
Requestor Location	Mrs. Tamara Mueller Institute for AI in Medicine Technical University of Munich Einsteinstr. 25 Munchen, other 81675 Germany Attn: Mrs. Tamara Mueller
Billing Type	Invoice
Billing Address	Mrs. Tamara Mueller Institute for AI in Medicine Technical University of Munich Einsteinstr. 25 Munchen, Germany 81675 Attn: Tamara Mueller
Total	0.00 EUR

Terms and Conditions

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/ magazine	Lifetime of journal/magazine
Reuse in a book/ textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/ poster. Note: publication whether electronic or in print of presentation/ slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/ CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/ classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/ tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/ romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted

hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

Other Conditions:

Version 1.4 - Dec 2022

Questions? customercare@copyright.com.

CCC RightsLink

Sign in/Register

IEEE Requesting permission to reuse content from an IEEE publication

Differentially Private Graph Neural Networks for Medical Population Graphs and The Impact of The Graph Structure

Conference Proceedings: 2024 IEEE International Symposium on Biomedical Imaging (ISBI)

Author: Tamara T. Mueller

Publisher: IEEE

Date: 27 May 2024

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Privacy - Terms