

Towards 4D Human Perception with Novel Representations for Gait Recognition and Tracking

Torben Eike Teepe

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)
genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Werner Hemmert

Prüfer der Dissertation: 1. Prof. Dr.-Ing. Gerhard Rigoll
2. Prof. Dr.-Ing. Horst-Michael Groß

Die Dissertation wurde am 11.11.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 05.05.2025 angenommen.

Abstract

The rapid development of computer vision technologies has significantly influenced daily life and enabled functionalities like automated security systems. Despite these advancements, challenges remain in comprehending the 3D structure of environments and human motion. We address these challenges by incorporating the temporal dimension, emphasizing the need for a comprehensive understanding of human activities over time.

Gait recognition, the process of identifying individuals based on their walking patterns, presents unique advantages in surveillance as it allows identification from a distance without the subject's cooperation. Traditional methods rely on silhouettes extracted through background subtraction, which often include irrelevant appearance details and fail to capture temporal information accurately. We propose a novel approach using skeleton-based representations of human gait. By leveraging graph neural networks (GNNs), our method aims to capture and analyze motion patterns more effectively while being less susceptible to variations in clothing and shoes. Our key contributions to this area include the introduction of a skeleton-based gait representation using GNNs to extract meaningful gait features, the development of a spatial-temporal graph convolutional network (GCN) for constructing and analyzing human skeleton graphs, and extensive experimentation on various datasets to validate these methods, demonstrating significant improvements over state-of-the-art techniques.

Multi-Object Tracking (MOT) is crucial for associating detections across frames to understand movement over time. MOT is also a necessary step for gait recognition, as it needs to associate detections across time to understand the temporal aspect of gait. A significant challenge in MOT is occlusion, where objects are hidden from the camera's view. Utilizing multiple cameras can mitigate this issue by providing different perspectives, but it introduces the complexity of associating detections across cameras and time. We integrate modern lifting methods with early fusion approaches to aggregate camera data in 3D space, addressing occlusions and enhancing tracking accuracy and computational efficiency. Our key contributions to this area include the proposal of a novel multi-view detection and tracking approach that fuses data from multiple cameras in 3D space, addressing occlusions and improving detection and tracking accuracy through enhanced methods, and validating

these approaches with extensive experimentation, showcasing improvements over current techniques.

In conclusion, our research significantly enhances the understanding of human behavior and movement by integrating 3D positional data with temporal movement analysis into a 4D representation. This thesis represents a substantial contribution, pushing the boundaries of 4D human perception and opening new avenues for human-machine communication.

Zusammenfassung

Die rasante Entwicklung der Technologien im Bereich der Computer Vision hat das tägliche Leben erheblich beeinflusst und ermöglicht Funktionen wie automatisierte Sicherheitssysteme. Trotz dieser Fortschritte bleiben Herausforderungen beim Verständnis der 3D-Struktur von Umgebungen, menschlicher Bewegung und Interaktionen bestehen. Wir gehen diese Herausforderungen an, indem wir die zeitliche Dimension einbeziehen und die Notwendigkeit eines umfassenden Verständnisses menschlicher Aktivitäten im Zeitverlauf betonen.

Die Gangerkennung, der Prozess der Identifizierung von Individuen basierend auf ihrem Gangmuster, bietet einzigartige Vorteile in der Überwachung, da sie eine Identifizierung aus der Ferne ohne Mitwirkung der betreffenden Person ermöglicht. Traditionelle Methoden basieren auf Silhouetten, die durch Hintergrundsubtraktion extrahiert werden und oft irrelevante visuelle Details enthalten und zeitliche Informationen ungenau erfassen. Wir schlagen einen neuartigen Ansatz vor, der auf skelettbasierten Darstellungen des menschlichen Gangs beruht. Durch die Nutzung von Graph Neural Networks (GNNs) zielt unsere Methode darauf ab, Bewegungsmuster effektiver zu erfassen und zu analysieren, während sie weniger anfällig für Variationen in Kleidung und Schuhen ist. Unsere wichtigsten Beiträge in diesem Bereich umfassen die Einführung einer skelettbasierten Gangerkennung mit GNNs zur Extraktion bedeutungsvoller Gangmerkmale, die Entwicklung eines räumlich-zeitlichen Graph Convolutional Network (GCN) zur Konstruktion und Analyse von menschlichen Skelettgraphen und umfangreiche Experimente an verschiedenen Datensätzen zur Validierung dieser Methoden, die erhebliche Verbesserungen gegenüber dem Stand der Technik zeigen.

Multi-Object Tracking (MOT) ist entscheidend für die Assoziierung von Detektionen über mehrere Frames hinweg, um Bewegungen im Zeitverlauf zu verstehen. MOT ist auch notwendig für die Gangerkennung, da sie Objekte über Zeit assoziieren muss, um die zeitlichen Aspekte des Ganges zu verstehen. Eine wesentliche Herausforderung im MOT ist die Verdeckung, bei der Objekte aus dem Blickfeld der Kamera verschwinden. Die Nutzung mehrerer Kameras kann dieses Problem durch verschiedene Perspektiven verringern, führt jedoch zu der Komplexität, Detektionen über Kameras und Zeit hinweg zu assoziieren. Wir integrieren moderne Lifting-Methoden

mit frühen Fusionsansätzen, um Kameradaten im 3D-Raum zu aggregieren, Verdeckungen zu adressieren und die Genauigkeit und Effizienz der Verfolgung zu verbessern. Unsere wichtigsten Beiträge in diesem Bereich umfassen den Vorschlag eines neuartigen Ansatzes zur Erkennung und Verfolgung mit mehreren Ansichten, der Daten aus mehreren Kameras im 3D-Raum fusioniert, Verdeckungen adressiert und die Erkennungs- und Verfolgungsgenauigkeit durch verbesserte Methoden verbessert sowie diese Ansätze durch umfangreiche Experimente validiert und dabei Verbesserungen gegenüber aktuellen Techniken zeigt.

Zusammenfassend verbessert unsere Forschung das Verständnis von menschlichem Verhalten und Bewegung erheblich, indem sie 3D-Positionsdaten mit der Analyse zeitlicher Bewegungen zu einer 4D-Darstellung integriert. Diese Dissertation erweitert die Personenerkennung auf die vierte Dimension und eröffnet neue Möglichkeiten in der Mensch-Maschine-Kommunikation.

Contents

List of Acronyms	xi
List of Acronyms	xi
List of Figures	xiv
List of Tables	xv
Notation	xvii
1. Introduction	1
1.1. Motivation & Research Objectives	2
1.2. Thesis Outline	4
2. Background	7
2.1. Deep Neural Networks	8
2.1.1. Multi-Layer Perceptron	8
2.1.2. Learning as an Optimization Problem	9
2.1.3. Regularization	12
2.2. Deep Metric Learning	14
2.2.1. Loss Functions	14
2.3. Convolutional Neural Networks	16
2.3.1. Deep and Residual Neural Network	17
2.4. Graph Neural Networks	19
2.4.1. Graph Neural Network Layers	19
2.5. Vision Transformer Architectures	21
2.5.1. Swin Transformer	23
2.5.2. Deformable Attention	24
2.6. Projective Geometry	25
2.6.1. Camera Projection	25
2.7. Epipolar Geometry	28
3. Multi-View 3D Tracking & Detection	31
3.1. Related Work	33
3.1.1. Multi-View Object Detection	34

3.1.2.	3D Perception Systems	35
3.1.3.	Multi Object Tracking	36
3.1.4.	Single-Stage Multi-Target Tracking	37
3.1.5.	Multi-Target Multi-Camera Tracking	38
3.1.6.	3D Gaussian Splatting	39
3.2.	Tracking by Detection in the Bird's Eye View	40
3.2.1.	Encoder	41
3.2.2.	Temporal Aggregation	41
3.2.3.	Detection & Tracking Heads	42
3.3.	Lifting to the Bird's Eye View	43
3.3.1.	Perspective Transformation	44
3.3.2.	Bilinear Sampling	45
3.3.3.	Depth Splatting	46
3.3.4.	Deformable Attention	47
3.4.	Lifting with Gaussian Splatting	49
3.5.	Experiments	50
3.5.1.	Datasets	51
3.5.2.	Evaluation Metrics	54
3.5.3.	Implementation Details	54
3.5.4.	Comparison to State-of-the-Art	55
3.5.5.	Ablation Studies	57
3.5.6.	Qualitative Results	59
3.5.7.	Computational Efficiency	60
3.6.	Discussion	62
3.7.	Conclusion	63
4.	Skeleton-based Gait Recognition	65
4.1.	Related Work	68
4.1.1.	Gait Representations	68
4.1.2.	Silhouette-based Gait Recognition	71
4.1.3.	Model-based Gait Recognition	72
4.1.4.	Multi-Modal Gait Recognition	74
4.1.5.	Novel Research Directions in Gait Recognition	74
4.1.6.	Skeleton-based Action Recognition	75
4.2.	Overview of GaitGraph	76
4.3.	Skeleton-based Gait Representation	77
4.3.1.	Human Pose Estimation	77
4.3.2.	2D Human Pose Estimation	78
4.3.3.	3D Human Pose Estimation	80
4.4.	Gait Recognition with Graph Convolutional Networks	80
4.4.1.	Skeleton Graph Construction	80
4.4.2.	Spatial Graph Convolutional Neural Network	81
4.4.3.	Implementing ST-GCN	84

4.5. ResGCN Architecture	85
4.5.1. Multi-Branch Input	86
4.5.2. ResGCN Block	87
4.6. Experiments	88
4.6.1. Datasets	88
4.6.2. Evaluation Metrics	91
4.6.3. Implementation Details	91
4.6.4. Main Results	92
4.6.5. Ablation Studies	97
4.6.6. Cross-Domain Generalization	99
4.6.7. Computational Efficiency	100
4.6.8. Gait Recognition Visualization	102
4.7. Discussion	104
4.8. Future Work	105
4.9. Conclusion	107
5. Conclusion	109
5.1. Contributions	109
5.2. Conclusion	110
5.3. Outlook	111
A. Tracking Metrics	115
References	121
Publications	138
Supervised Student Works	141

List of Acronyms

BEV	Bird's Eye View.
CAM	Class Activation Map.
CNN	Convolutional Neural Network.
DNN	Deep Neural Network.
FLOPs	Floating Point Operations.
FOV	Field of View.
GCN	Graph Convolutional Network.
GEI	Gait Energy Image.
GNN	Graph Neural Network.
LiDAR	Light Detection and Ranging.
LSTM	Long Short-Term Memory.
mAP	mean Average Precision.
mINP	mean Inverse Negative Penalty.
MLP	Multi-Layer Perceptron.
MODA	Multiple Object Detection Accuracy.
MODP	Multiple Object Detection Precision.
MOT	Multiple Object Tracking.
MOTA	Multiple Object Tracking Accuracy.
MSA	Multi-Head Self-Attention.
MTMC	Multi-Target Multi-Camera.
NAS	Neural Architecture Search.
NMS	Non-Maximum Suppression.
POM	Probabilistic Occupancy Map.
PTSN	Pose-Based Temporal-Spatial Network.
re-ID	Re-Identification.
ResGCN	Residual Graph Convolutional Network.
ResNet	Residual Neural Networks.
RNN	Recurrent Neural Network.
SGD	Stochastic Gradient Descent.
SH	Spherical Harmonics.
SMPL	Skinned Multi-Person Linear.
ST-GCN	Spatial-Temporal Graph Convolutional Network.
TbD	Tracking-by-Detection.
TTA	Test Time Augmentation.
ViT	Vision Transformer.

List of Figures

1 Introduction	
1.1. Gait Silhouette	3
Images from [176]	
1.2. Occlusion in Multi-View Tracking	4
Image from [110]	
1.3. Publications Overview	4
2 Background	
2.1. 2D Convolution Operation	16
2.2. ResNet-101 Architecture	18
Image created with: [75]	
2.3. Graph Neural Network Layer Types	20
Image Source: [15]	
2.4. Vision Transformer Model	21
Image Source: [34]	
2.5. Swin Transformer	23
Image Source: [102]	
2.6. Deformable Attention	24
Image Source: [182]	
2.7. Pinhole Camera Geometry	26
2.8. Epipolar Geometry	29
Adapted from: [149]	
3 Multi-View 3D Tracking & Detection	
3.1. Example of a Multi-View Scene	32
Images from: [22]	
3.2. Projection Shadows	34
Images adapted from [69]	
3.3. Fusion Strategies for Multi-Target Multi-Camera (MTMC)	38
3.4. Overview of Our Approach	40
Image Source: [11 [†]]	
3.5. Visualization of the Feature Encoder	41
Image Source: [10 [†]]	
3.6. Perspective Transformation	44
3.7. Bilinear Sampling	45
Image Source: [11 [†]]	

3.8. Depth Splatting	46
Image Source: [11 [†]]	
3.9. BEVFormer Lifting	47
Image Source: [11 [†]]	
3.10. 3D Gaussian Splatting	49
3.11. 3D Gaussian	50
3.12. Synthicle Ambiances	53
Image Source: [3 [†]]	
3.13. Qualitative Results Pedestrian	59
Image Source: [10 [†]]	
3.14. Qualitative Results Vehicles	60
Image Source: [11 [†]]	
3.15. Computational Efficiency of Lifting Methods	61
4 Skeleton-based Gait Recognition	
4.1. Gait Cycle	66
Image Source: [119]	
4.2. Gait Representations	67
Image Source: [8 [†] , 169]	
4.3. History of Gait Representations	69
Image Source: [65, 9 [†] , 8 [†] , 176]	
4.4. Overview of our Approach	77
Image Source: [9 [†]]	
4.5. Human Pose Estimation with Heatmaps	78
4.6. Images as Graphs	82
Image Source: [129]	
4.7. Multi-Branch Input	85
Image Source: [141]	
4.8. Extended ResGCN Architecture	86
Image Source: [8 [†]]	
4.9. ResGCN Block	87
Image Source: [8 [†]]	
4.10. Example Image from Gait Datasets	90
Image Source: [169, 146, 183, 176]	
4.11. Analysis of Computational Efficiency	101
4.12. Gait Activation Analysis	103
Image Source: [8 [†]]	
A Tracking Metrics	
A.1. Bijective Matching	116
Adapted from: [9]	
A.2. Mismatched Tracks	118
Adapted from: [9]	

List of Tables

3 Multi-View 3D Tracking & Detection

3.1. Multi-View Tracking Datasets	51
3.2. Pedestrian Detection	55
3.3. Pedestrian Tracking	56
3.4. Synthehicle	57
3.5. Encoder Ablation	58
3.6. Temporal Ablation	58

4 Skeleton-based Gait Recognition

4.1. Overview of Gait Approches	73
4.2. Gait Datasets Comparison	88
4.3. CASIA-B Results	93
4.4. OU-MVLP Results	94
4.5. GREW & Gait3D Results	95
4.6. Model Ablation	97
4.7. Temporal Ablation	98
4.8. Keypoint Dimensions	98
4.9. Cross- and Single-Domain Performance	99

Adapted from [48]

Notation

In this section, we will introduce the notation that is used in this work.

Numbers, Arrays, and Points

a	A scalar (integer or real number)
A	A constant
x	A vector
X	A matrix or tensor
$\mathbf{p}$	A point in 2D
$\mathbf{P}$	A point in 3D

Sets and Graphs

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^{n \times m}$	The set of $n \times m$ real matrices
$\{0, 1, \dots, n\}$	The set of integers from 0 to n
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$

Indexing

a_i	Element i of a vector $\mathbf{a}$, with indexing starting at 1
$A_{i,j}$ or $A(i, j)$	Element i, j of a matrix A

Calculus

$\frac{dy}{dx}$	Derivative of y with respect to x
$\nabla_x y$	Gradient of y with respect to x
$\otimes$	Tensor product

Functions

$f : \mathbb{A} \rightarrow \mathbb{B}$	Function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(x; \theta)$	Function f of x parameterized by θ
$\ x\ $	L^2 norm of x
$\mathcal{L}$	Loss function (per sample)
J	Total cost/loss function (e.g. sum of losses)

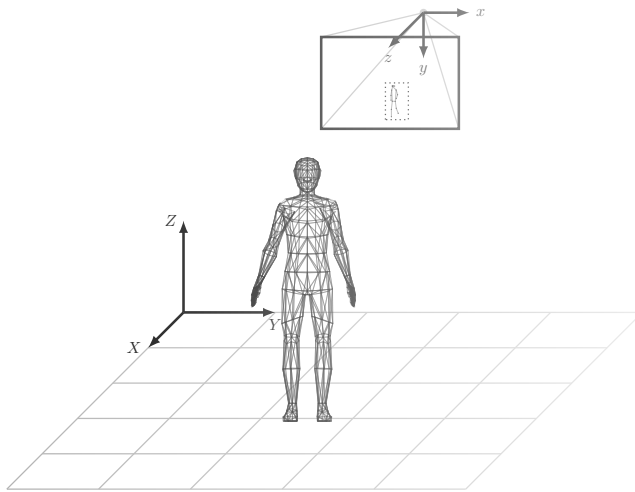
Datasets

$\mathcal{D}$	Dataset
$x^{(i)}$	i -th sample of dataset
$y^{(i)}$ or $\mathbf{y}^{(i)}$	Label of i -th sample
$\hat{y}^{(i)}$ or $\hat{\mathbf{y}}^{(i)}$	Predicted label of i -th sample

Citations

[9 [†]]	Self-citations are highlighted by a [†]
[2 ⁺]	Citation to a work by a supervised student are highlighted by a ⁺
[5]	Citation to an external work

Multi-references are separated by a comma in alphanumerical order, not in the order of appearance in the text, for example: The works of Elsner [3] and Böttcher [3] were both completed in 2017 in cooperation with Autoliv [2,3]. To avoid clutter from frequently recurring references, each reference is usually repeated *only once per paragraph* unless an attribution is ambiguous.



1

Introduction

The field of computer vision has undergone rapid advancement in recent years, becoming increasingly integrated into our daily lives. Contemporary technologies enable facial recognition for device unlocking, traffic sign detection in vehicles, and automated person identification in security camera footage. These capabilities are facilitated by modern computer vision algorithms that leverage vast amounts of data. The advent of deep learning methodologies has revolutionized the field, expanding the boundaries of automated visual perception.

This work focuses on the challenges associated with human perception via camera systems. The proliferation of cameras in both private and public spaces has created an urgent need for automated, real-time analysis of the collected visual data. To extract meaningful information from these video streams, it is imperative to comprehend the human world in its full complexity. This encompasses understanding the three-dimensional structure of the environment, the kinematics of people and objects, and the intricate interactions between them. While many of these camera systems are deployed for surveillance purposes, their potential applications extend far beyond, encompassing smart home systems, automated retail, and numerous other domains. All these applications necessitate the ability to detect individuals, track their motion, and perform person recognition. While these tasks may appear trivial to human observers, the temporal aspects pose significant challenges for computer vision algorithms. This temporal dimension serves as a central focus of our research.

In the domain of recognition, we concentrate on gait recognition - the process of identifying or verifying an individual's identity based on their ambulatory patterns. Analogous to fingerprints or facial features, gait is a biometric characteristic unique to each individual. While there are static aspects of gait such as height, weight, and build, the core of gait recognition lies in capturing and analyzing motion patterns. Gait recognition offers several advantages over other biometric methods, including the ability to identify individuals at a distance and without their cooperation, rendering it particularly valuable in surveillance and security applications.

The second part of the thesis is dedicated to multi-view tracking. Tracking involves following the motion of a person or object across multiple frames, again emphasizing the crucial temporal aspect of our work. Tracking is not only a challenging problem in its own right but also a prerequisite for effective gait recognition. By associating detections across frames, we can analyze the motion of the same person over time, enabling more robust and accurate gait analysis. In this work, we extend our understanding of tracking to three-dimensional space and multiple cameras, addressing the complexities that arise when integrating information from various viewpoints and dealing with occlusions.

By combining three-dimensional positional data with temporal movement analysis, we arrive at a four-dimensional representation of the world (3D space + time). This 4D perspective allows for a more comprehensive understanding of human behavior and movement, opening up new possibilities in areas such as behavioral analysis, anomaly detection, and predictive modeling of human actions.

In summary, this thesis aims to make significant contributions to the field of computer vision, providing novel insights into 4D human understanding. By advancing the state-of-the-art in gait recognition and multi-view tracking, we endeavor to enable more intelligent and responsive systems that can better understand and interact with the human world. Our work has potential applications across various domains, from security and surveillance to healthcare and human-computer interaction, paving the way for more sophisticated and nuanced interactions between humans and intelligent systems.

1.1. Motivation & Research Objectives

Our work brings several contributions to the field of gait recognition and multi-view tracking. We will now discuss the motivation for our work and the research objectives we pursue.

Gait Recognition Gait recognition presents a challenging task due to the complex nature of human locomotion. It is influenced not only by an individual's anthropometric characteristics such as height, weight, and build, but also by their unique walking style, which can be affected by factors such as clothing, footwear, and the walking surface. Gait recognition is typically performed not on raw RGB images, but on

representations that abstract gait information while obscuring visual details. To date, most works have utilized background subtraction techniques to extract the person's silhouette. Figure 1.1 illustrates an example of a silhouette in a supermarket environment, where the subject is carrying a basket. The elevated position of the camera and the background subtraction method render this a challenging silhouette to extract accurately. However, this silhouette also contains extraneous appearance information, such as clothing and hair, which is not directly relevant to gait recognition and can make algorithms susceptible to changes in these factors. Furthermore, extracting temporal information from a sequence of silhouettes is non-trivial.



Figure 1.1.: **Gait Silhouette.** Overhead view of a person's silhouette captured while walking with a shopping basket, demonstrating the challenges of silhouette extraction in real-world scenarios.

Our work aims to address these challenges by investigating a novel gait representation based on a skeletal model of the human body. We propose the utilization of a graph neural network to extract gait features from this skeletal representation. We seek to evaluate whether this approach provides a more suitable representation for gait recognition and if it allows for the extraction of genuine temporal information. Additionally, we aim to assess whether this representation is less susceptible to changes in clothing and footwear.

Multi-View Detection & Tracking Multiple Object Tracking (MOT) involves the task of tracking multiple objects in a video sequence. The primary challenge in tracking is occlusion, where the object of interest may be obscured by other objects or fall outside the camera's field of view. A straightforward approach to mitigate this problem is the use of multiple cameras, which can observe the same scene from different perspectives. This configuration reduces the likelihood of complete occlusion, as an object obscured in one view may be visible from another. An example is illustrated in Figure 1.2, where a target object (denoted by a red rectangle) is occluded in one camera but visible in others. However, the introduction of multiple cameras presents a new challenge: the association of detections across cameras and across time. Conventionally, these two association problems are addressed separately, leading to the accumulation of errors over time.

Our work aims to overcome these challenges by proposing a novel approach to multi-view tracking that combines modern lifting methods with an early fusion approach.

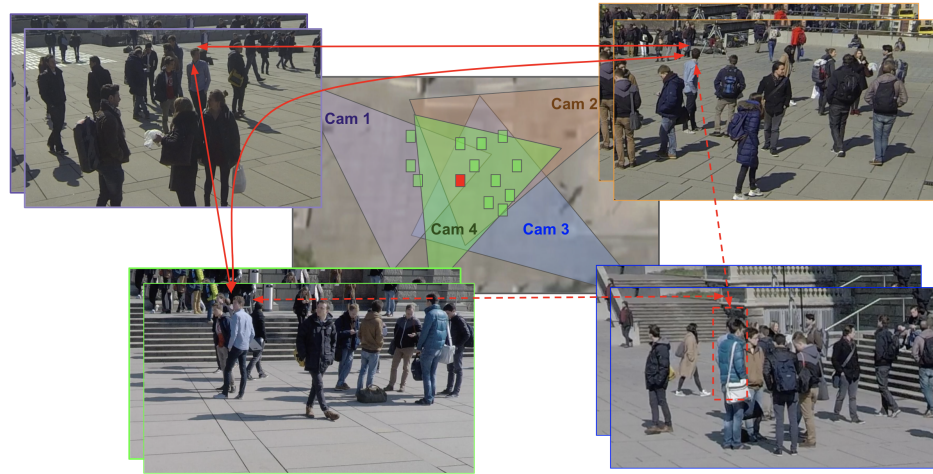


Figure 1.2.: **Occlusion in Multi-View Tracking.** A target object (red rectangle) is occluded in Camera 3 but is still observed the other cameras.

By aggregating data from all cameras in 3D space, we can address the occlusion problem and associate detections across cameras. We seek to evaluate whether this approach can improve tracking accuracy and enable real-time multi-view tracking systems. Additionally, we aim to assess which association method, appearance-based or motion-based, is most suitable for this task.

1.2. Thesis Outline

We present the structure of a detection, tracking, and gait recognition system, along with our respective publications, in Figure 1.3. The main body of the thesis is divided into two chapters that we present in the order they appear in the pipeline. We start with our work on multi-view tracking and detection and follow with gait recognition, the last step of the pipeline. The order of the two main chapters is interchangeable as to date gait recognition does not consider the first part of the recognition pipeline and assumes perfect tracklets. Finally, we conclude the thesis with a summary of our contributions and a discussion of the results.

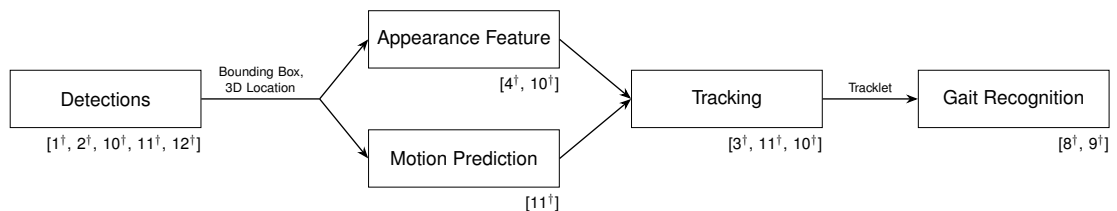


Figure 1.3.: **Publications Overview.** The figure shows the main research areas of the publications. Starting with detections, either 2D or 3D, appearance or motion features are extracted. These features are then used in the association yielding tracklets. Finally, gait recognition is performed on the tracklets.

Chapter 2 introduces the notation and establishes the mathematical foundation essential for comprehending the subsequent chapters. We present the fundamentals of Convolutional Neural Network (CNN), with a focus on convolutional neural networks and metric learning. Furthermore, we discuss more specialized neural architectures, including Transformers and Graph Neural Network (GNN). Finally, we revisit the pinhole camera model and perspective projection.

Chapter 3 presents our work on multi-view tracking. We begin with a brief introduction to multi-view detection and the evolution of multi-view tracking. We then introduce our novel multi-view tracking approach that combines modern lifting methods with an early fusion approach. We provide comprehensive experimental results and comparisons to previous and current state-of-the-art methods. Finally, we discuss the limitations and potential future work in multi-view tracking.

Chapter 4 presents our work on gait recognition. The chapter begins with a brief introduction to gait recognition and the historical development of gait representation. We then introduce our novel gait representation based on a graph neural network. We provide extensive experimental results and comparisons to previous and current state-of-the-art methods. Finally, we discuss future directions and potential advancements in gait recognition.

Chapter 5 concludes the thesis with a summary of our contributions and a discussion of the results. We also revisit the research questions stated in Section 1.1 and analyze our progress in 4D human understanding, as well as outlining future research directions.

Chapter A provides an overview of the tracking metrics employed in this thesis.

2

Background

Deep neural networks have revolutionized the fields of machine learning and artificial intelligence, demonstrating remarkable capabilities in solving complex problems across diverse domains. Their success can be attributed to several factors, including the availability of large-scale datasets, advancements in hardware technology, and the development of sophisticated optimization techniques. These networks have significantly transformed tasks such as pattern recognition, feature extraction, and decision-making, leading to groundbreaking advancements in image classification, object detection, speech recognition, and natural language understanding.

This thesis builds upon these advancements, focusing on three key architectural paradigms: Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), and Transformer models. CNNs excel in processing grid-like data structures, such as images and video sequences, while GNNs extend deep learning capabilities to non-Euclidean data structures. Transformer models, initially developed for natural language processing tasks, have demonstrated remarkable potential in computer vision applications [34, 102].

Furthermore, this work explores the principles of epipolar geometry, which are crucial for understanding multi-view perception and three-dimensional reconstruction. By

leveraging these advanced deep learning architectures and incorporating fundamental concepts from epipolar geometry, this thesis advances temporal 3D perception.

2.1. Deep Neural Networks

This section introduces the fundamental building blocks of general Deep Neural Networks (DNNs) and discusses effective training methodologies. The content presented here focuses on the most critical concepts and definitions essential for comprehending the subsequent chapters of this work. For a more comprehensive overview, readers are directed to [52], which serves as the primary reference for this section.

2.1.1. Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) represents the most fundamental type of feed-forward neural network, comprising multiple layers of neurons. Each layer performs a linear transformation followed by a non-linear activation function. A feedforward network defines a mapping $y = f(x; \Theta)$ and learns the optimal values of the parameters Θ that result in the best function approximation, *i.e.* mapping the input x to the category y .

The term *perceptron*, coined by Rosenblatt [128], originally referred to a single neuron. In modern usage, it can also describe a single-layer neural network — the simplest form of a Multi-Layer Perceptron (MLP). For an input vector $x \in \mathbb{R}^D$ and layer ℓ , the output of a perceptron is given by:

$$y^\ell = f^\ell(x; \Theta^\ell) = \sigma(W^\ell x + b^\ell), \quad (2.1)$$

where $W^\ell \in \mathbb{R}^{d \times D}$ and $b^\ell \in \mathbb{R}^d$ denote the weight matrix and bias vector, respectively, and $\sigma(\cdot)$ represents a non-linear activation function. The weight matrix W^ℓ and bias vector b^ℓ constitute the learnable parameters Θ^ℓ of the perceptron.

An L -layer Multi-Layer Perceptron (MLP) is constructed by stacking layers of perceptrons such that the output of the $(\ell - 1)$ -th layer serves as the input to the ℓ -th layer. This architecture can be described as a directed acyclic graph. For instance, consider three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ chained to form $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x))) = f^{(3)} \circ f^{(2)} \circ f^{(1)}(x)$. Here, $f^{(1)}$ represents the input layer, $f^{(2)}$ the first hidden layer, and $f^{(3)}$ the output layer. The overall length of the chain $L = 3$ determines the depth, hence the term *deep* neural network. The dimensionality of the hidden layers defines the width of the model. These networks are termed *neural* due to their loose inspiration from the structure of the human brain.

Activation Functions The activation function $\sigma(\cdot)$ in Equation (2.1) is applied element-wise to the input. Historically, the sigmoid function $\text{sigmoid}(x) = 1 / (1 + e^{-x})$ was employed as the activation function. However, it is now rarely used due to numerical instabilities and issues with gradient propagation. The sigmoid function is also not *zero-centered*, *i.e.* the mean of its output is non-zero, which can lead to complications in the optimization process, such as exploding or vanishing gradients.

In contemporary practice, the Rectified Linear Unit (ReLU) function $\text{ReLU}(x) = \max(0, x)$ is widely adopted. The ReLU function is computationally efficient and has demonstrated superior performance in various applications. However, it is non-differentiable at $x = 0$, which can potentially impede the optimization process. To address this limitation, variants such as the *Leaky ReLU* function $\text{LeakyReLU}(x) = \max(0.1x, x)$ have been proposed. While numerous other activation functions exist, the ReLU family remains particularly popular due to its simplicity and effectiveness.

2.1.2. Learning as an Optimization Problem

This section elucidates the process of optimizing the parameters of a neural network. Training a neural network can be formulated as an optimization problem, where the objective is to determine the optimal parameters Θ that minimize the loss function $J(\Theta)$. It is important to note that optimization in the context of machine learning differs from the optimization of mathematical problems. In machine learning, the primary goal is not to find the global minimum of the loss function, but rather to use the loss function as an indirect means of optimizing a performance metric P on unseen data. For instance, we might aim to optimize the classification accuracy P on the test set, but as P is not differentiable, we employ the cross-entropy cost function J to indirectly optimize P .

Typically, the cost function can be expressed as:

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^m \mathcal{L}(f(\mathbf{x}^{(i)}; \Theta), y^{(i)}) \quad (2.2)$$

where $\mathcal{L}$ represents the per-example loss function, $f(\cdot)$ denotes the neural network model, and m is the number of training samples. To optimize the cost function, we perform gradient descent with respect to the parameters $\nabla_{\Theta} J(\Theta)$. Computing the gradient requires propagating the error from the output layer to the input layer, a process accomplished by the *backpropagation* algorithm. In most practical scenarios, computing the gradient for the entire dataset is computationally infeasible due to memory constraints. Consequently, we compute the gradient for a subset of the data, referred to as a *minibatch*. Methods that utilize minibatches are termed *stochastic* methods, with Stochastic Gradient Descent (SGD) being the most prevalent. Optimization with SGD encounters challenges similar to those in mathematical optimization, including poor conditioning of the Hessian matrix, non-convexity, and the presence of local minima and saddle points. These factors contribute to the

complexity of the optimization problem, and the choice of optimization algorithm can significantly impact the model's performance.

Backpropagation Backpropagation is an efficient algorithm for calculating the gradient of the loss with respect to every parameter in the network. The fundamental concept of neural networks involves decomposing computation into a series of layers, which can be represented as a computational graph. As the name suggests, backpropagation propagates the error backwards through this computational graph.

The key innovation of backpropagation lies in its reuse of computation. To elucidate this concept, consider a simple computational graph represented by a chain of functions $f^\ell \circ f^{\ell-1} \circ \dots \circ f^{(1)}$, where each function f^ℓ has parameters θ^ℓ and output x^ℓ . The objective is to optimize the parameters θ^ℓ with respect to a loss function $\mathcal{L}$, which outputs a scalar value J .

To update the parameters θ^ℓ , it is necessary to compute the gradients $\frac{\partial J}{\partial \theta^1}, \frac{\partial J}{\partial \theta^2}, \dots, \frac{\partial J}{\partial \theta^\ell}$. Each of these gradients can be computed using the chain rule of calculus:

$$\frac{\partial J}{\partial \theta^{(1)}} = \boxed{\frac{\partial J}{\partial x^\ell} \frac{\partial x^\ell}{\partial x^{\ell-1}} \dots \frac{\partial x^{(3)}}{\partial x^{(2)}}} \frac{\partial x^{(2)}}{x^{(1)}} \frac{\partial x^{(1)}}{\partial \theta^{(1)}} \quad (2.3)$$

$$\frac{\partial J}{\partial \theta^{(2)}} = \boxed{\frac{\partial J}{\partial x^\ell} \frac{\partial x^\ell}{\partial x^{\ell-1}} \dots \frac{\partial x^{(3)}}{\partial x^{(2)}}} \frac{\partial x^{(2)}}{\partial \theta^{(2)}} \quad (2.4)$$

The boxed terms in Equation (2.3) represent the computations that are reused in the calculation of the gradients. This reuse of computations from the forward pass to compute gradients in the backward pass constitutes the core principle of backpropagation.

Stochastic Gradient Descent Stochastic Gradient Descent (SGD) [127] and its variants are the predominant optimization algorithms for training DNNs. The fundamental principle of Stochastic Gradient Descent (SGD) is to update the parameters in the direction opposite to the gradient of the loss function. This is achieved by computing the gradient of the loss function with respect to the parameters and subtracting it from the parameters. Algorithm 1 illustrates the process of following the gradient downhill.

Algorithm 1 Stochastic Gradient Descent (SGD) update at training iteration k

Require: Learning rate ϵ_k

Require: Initial parameters Θ_0

while stopping criterion is not met **do**

 Sample minibatch $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$

 Compute gradient $g^k \leftarrow +\frac{1}{m} \nabla_{\Theta} \sum_{i=1}^m \mathcal{L}(f(x^{(i)}; \Theta), y^{(i)})$

 Update parameters $\Theta^{k+1} = \Theta^k - \epsilon_k g^k$

end while

A critical parameter in SGD is the learning rate ϵ_k , which determines the step size in the parameter space at each iteration k . An excessively high learning rate may cause the parameters to overshoot the minimum, while an excessively low learning rate may result in slow convergence. In practice, the learning rate is typically chosen manually, and a learning rate scheduler is employed to adjust the learning rate during training.

Momentum Momentum [122] is a technique that enhances SGD by addressing the issues of poor conditioning of the Hessian matrix and variance in the stochastic gradient. The term momentum, inspired by physics, describes the inertia of an object. In the context of SGD, the momentum term facilitates continued movement in the direction of the previous update and mitigates oscillations. This is accomplished by adding a fraction α of the update vector from the previous time step to the current update vector:

$$\mathbf{v}^{k+1} \leftarrow \alpha \mathbf{v}^k + \mathbf{g}^k \quad (2.5)$$

$$\Theta^{k+1} \leftarrow \Theta^k - \epsilon^k \mathbf{v}^{k+1} \quad (2.6)$$

The magnitude of the momentum term α determines the extent to which the previous update vector $\mathbf{v}_k$ influences the current update vector $\mathbf{v}_{k+1}$. The SGD algorithm presented in Algorithm 1 can be augmented with this momentum term.

Adam Adaptive Moment Estimation (Adam) [82] is an optimization algorithm that builds upon the principles of SGD and momentum while addressing some of their limitations. Similar to SGD, Adam updates parameters based on gradient information, and like momentum, it accumulates past gradients to smooth out the updates. However, Adam extends these concepts by adapting the learning rate for each parameter individually.

Adam estimates both the first-order moment (analogous to momentum) and the second-order moment of the gradients. The first-order moment aids in maintaining the overall direction of updates, similar to the momentum method discussed previously. The second-order moment enables the algorithm to adapt the learning rate for each parameter separately, which is particularly beneficial when dealing with poorly conditioned problems, as mentioned in the SGD discussion.

Furthermore, Adam incorporates a bias correction mechanism to counteract the bias towards zero in the early stages of training. This feature, combined with its adaptive learning rates and momentum-like behavior, often results in faster convergence and improved performance compared to standard SGD or SGD with momentum, especially in scenarios with noisy or sparse gradients.

Batch Normalization Batch Normalization [74] is a meta-technique employed to enhance the training of deep neural networks by normalizing the inputs of each layer. While empirical evidence strongly supports the efficacy of this technique, its theoretical underpinnings are not yet fully elucidated.

The core concept of Batch Normalization is to normalize the inputs of each layer for each mini-batch. This is achieved by subtracting the batch mean and dividing by the batch standard deviation. Subsequently, the normalized values are scaled and shifted by learnable parameters.

For a mini-batch B of size m , the mean μ_B and variance σ_B^2 are computed as follows:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x^{(i)} \quad \sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_B)^2 \quad (2.7)$$

Each dimension i for a d -dimensional input $\mathbf{x} = (x^{(1)}, \dots, x^{(d)})$ is first normalized (*i.e.* re-centered and re-scaled) and then subjected to a learnable affine transformation:

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu_B}{\sqrt{\sigma_B^2 + \delta}}, \quad (2.8)$$

$$y^{(i)} = \gamma \hat{x}^{(i)} + \beta. \quad (2.9)$$

where δ is a small constant added for numerical stability, and γ and β are learnable parameters that allow the network to undo the normalization if necessary.

During inference, the moving averages of mean and variance computed during training are utilized instead of mini-batch statistics, ensuring consistent output regardless of batch size.

2.1.3. Regularization

A fundamental challenge in machine learning is to develop algorithms that generalize effectively to unseen data. This field of study is collectively referred to as *regularization*. In the following subsections, we will elucidate several common regularization techniques employed in DNN.

Weight Decay Weight decay, also known as L2 regularization, is a technique that penalizes large weights within the network. This approach encourages the network to learn weights closer to the origin, thereby mitigating overfitting. Mathematically, weight decay augments the loss function with a regularization term:

$$\mathcal{L}_{total} = \mathcal{L}_{original} + \frac{\lambda}{2} \|\Theta\|_2^2 \quad (2.10)$$

where $\mathcal{L}_{total}$ represents the total loss, $\mathcal{L}_{original}$ denotes the original loss function, λ is the weight decay coefficient, and Θ encompasses the learnable parameters of the network.

Data Augmentation The most effective method to enhance model performance is to increase the volume of training data. However, as data availability is often limited, performance can be improved by applying transformations to the existing input data, effectively generating new (x, y) pairs from the original dataset. This technique is particularly prevalent in image processing, where transformations such as rotation, translation, scaling, and cropping can be applied to generate additional training samples. It is important to note that data augmentation can be generalized to various types of data beyond images.

Multi-Task Learning Multi-Task Learning [19] is a paradigm where a single neural network is trained to perform multiple related tasks simultaneously. This approach exerts additional pressure on the model to learn more robust features, as it must generalize across multiple tasks. For instance, in a supervised learning context, we predict multiple outputs $y^{(i)}$ from the same input x :

$$y^{(i)} = f(x, \Theta). \quad (2.11)$$

The learnable parameters Θ can be partitioned into shared parameters Θ_{shared} and task-specific parameters $\Theta_{\text{task}}^{(i)}$.

Early Stopping When training large models with substantial representational capacity, there is a risk of overfitting to the training data, manifested by an increase in validation loss. Early stopping is a regularization technique that addresses this issue by storing the parameters Θ that yield the lowest validation loss and returning these parameters as the final model.

Dropout Dropout [142] is a regularization technique that introduces stochastic noise into the network during training. This prevents neurons from co-adapting and encourages the network to learn more robust features. During the training process, dropout randomly sets a fraction of input units to 0 at each update, thereby mitigating overfitting.

Mathematically, for a layer ℓ , the output of the dropout layer can be described as:

$$y^\ell = \begin{cases} 0 & \text{with probability } p \\ \frac{1}{1-p} \cdot f^\ell(y^{(\ell-1)}) & \text{otherwise} \end{cases} \quad (2.12)$$

where y^ℓ denotes the output of the ℓ -th layer $f^\ell(\cdot)$, and p represents the dropout probability. The factor $\frac{1}{1-p}$ is employed to scale the remaining neurons' outputs, maintaining the expected sum of the outputs.

During inference, dropout is typically deactivated, and the full network is utilized. Dropout can be interpreted as inducing an ensemble-like effect, as the network has learned to be robust to the removal of arbitrary neurons.

2.2. Deep Metric Learning

Deep metric learning, an application of DNNs, aims to learn a mapping into a latent space where the distance between data points serves as an effective proxy for their similarity in the original space. This approach has proven valuable in various applications, including Face Recognition [6[†], 5[†]] and Re-Identification (re-ID) [4[†]], where the objective is to learn a robust similarity measure between images.

For each data point $\mathbf{x}^{(i)} \in \mathcal{D}$, the goal is to learn a feature embedding $f(\mathbf{x}^{(i)}, \Theta)$ that maps the data points to a latent space of d dimensions:

$$\mathbf{f}^{(i)} = f(\mathbf{x}^{(i)}, \Theta) \in \mathbb{R}^d, \quad (2.13)$$

where $\mathbf{f}^{(i)} \in \mathbb{R}^d$ represents the feature embedding of the i -th data point. Typically, standard architectures such as MLP, Convolutional Neural Network (CNN), or Graph Neural Network (GNN) are employed to learn these feature embeddings. The overarching objective is to ensure that the feature embeddings $\mathbf{f}^{(i)}$ are highly discriminative, such that the distance between feature embeddings of the same identity is smaller than the distance between feature embeddings of different identities:

$$D(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}) \leq D(\mathbf{f}^{(i)}, \mathbf{f}^{(k)}) \quad \forall i, j \in \mathcal{D}_{id}; k \notin \mathcal{D}_{id} \quad (2.14)$$

where D denotes a distance function and $\mathcal{D}_{id}$ represents the set of data points for a single identity. While the Euclidean distance is commonly used as the distance metric D , alternative metrics such as the Mahalanobis distance can also be employed.

Many feature extractors impose a constraint on the final feature representation to have a unit norm. This constraint forces the embedding manifold to be an n -dimensional unit hypersphere. In such cases, angular separation serves as the metric to measure similarity and dissimilarity between feature representations. Given two feature representations $\mathbf{f}_i$ and $\mathbf{f}_j$, the cosine similarity between them can be computed as:

$$\text{sim}(\mathbf{f}^{(i)}, \mathbf{f}^{(j)}) = \cos(\theta_{ij}) = \frac{\mathbf{f}^{(i)T} \mathbf{f}^{(j)}}{\|\mathbf{f}^{(i)}\| \cdot \|\mathbf{f}^{(j)}\|} \quad (2.15)$$

where θ_{ij} represents the angle between the two feature representations. Given that the magnitude of the feature representation is 1, the dot product of the two feature representations directly yields the cosine similarity, which is equivalent to the angular separation between the two feature vectors.

2.2.1. Loss Functions

The evolution of research on loss functions for metric learning has led to the development of various approaches, each with distinct properties and applications. Early methods [130] utilized the softmax cross-entropy loss to classify each identity into

a single class. In this approach, the softmax function is first applied to the network output $\tilde{\mathbf{y}}$ to obtain a probability distribution over the classes:

$$\hat{\mathbf{y}}^{(i)} = \frac{e^{\tilde{\mathbf{y}}^{(i)}}}{\sum_{j=1}^M e^{\tilde{\mathbf{y}}^{(j)}}} \quad (2.16)$$

The softmax ensures that the network output $\hat{\mathbf{y}}^{(i)}$ represents a probability distribution over the m classes, with the i -th output neuron representing the probability $P(i = y \mid \mathbf{x})$.

Following the principle of maximum likelihood estimation, this probability is maximized by minimizing the negative log-likelihood, resulting in the cross-entropy loss:

$$\mathcal{L}_{CE} = - \sum_{i=1}^M \mathbf{y}^{(i)T} \log(\hat{\mathbf{y}}^{(i)}) \quad (2.17)$$

where $\mathbf{y}^{(i)}$ represents the one-hot encoded ground truth label. Despite its simplicity, this loss function has demonstrated efficacy in practical applications.

The ongoing research in metric learning has led to the development of more sophisticated loss functions, including triplet loss [130], contrastive loss [81], and various triplet-based loss [33, 63] functions.

Triplet Loss Triplet loss [130], a contrastive loss function, aims to minimize the distance between embeddings of positive pairs while maximizing the distance between embeddings of negative pairs. Consider three embeddings $\mathbf{f}_a$, $\mathbf{f}_p$, and $\mathbf{f}_n$ representing an anchor, a positive sample, and a negative sample, respectively. The objective is to minimize the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative sample:

$$\mathcal{L}_{triplet} = \sum_{a,p,n \in \mathcal{D}} \max(0, \alpha + \|\mathbf{f}_a - \mathbf{f}_p\|^2 - \|\mathbf{f}_a - \mathbf{f}_n\|^2), \quad (2.18)$$

where α denotes the margin that ensures a minimal distance between the positive and negative samples.

Triplet loss necessitates careful selection of informative triplets to efficiently create a highly separable embedding space. Informative triplets are those whose pairwise distances violate the margin. Sample mining, the process of identifying these informative samples, can be performed exhaustively offline for the entire dataset, but this approach becomes computationally expensive for large datasets. A more practical strategy involves performing sample mining within mini-batches, computing similarities only for samples in the current batch, thereby reducing computational cost.

2.3. Convolutional Neural Networks

Convolutional Neural Network (CNN) represent a class of deep neural networks particularly effective for processing time series, image, and video data. These networks are designed to learn spatial hierarchies of features from input data, enabling efficient processing and high performance in image-related tasks such as classification, object detection, and segmentation. Fundamentally, CNNs can be conceptualized as MLPs that employ convolution operations in place of matrix multiplication.

Compared to traditional MLPs, CNNs offer two primary advantages:

- **Parameter Sharing:** In a CNN, each filter (or kernel) is applied across the entire input data, sharing the same set of parameters. This means that the same weights and biases are used to detect specific features regardless of their position in the input. This approach significantly reduces the number of parameters compared to fully connected networks, enhances the model's efficiency, and allows the network to detect features invariant to their location in the input.
- **Inductive Bias:** CNNs incorporate a prior belief in the structure of the input data. This is achieved through the use of convolutional layers, which exploit the local structure of the input data. By stacking multiple convolutional layers, CNNs can learn increasingly abstract features, facilitating translation invariance in the input data.

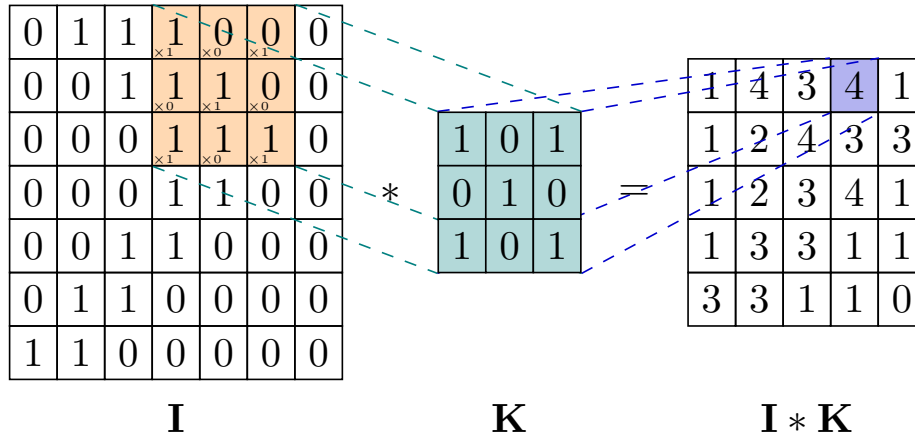


Figure 2.1.: **2D Convolution Operation.** Illustration of the 2D convolution operation. The input is convolved with the kernel to produce the output.

Convolution The convolution operation serves as the fundamental building block of CNNs. As this work focuses on 2D data, we will concentrate on 2D convolutions. In CNN terminology, an input is convolved with a learned kernel, resulting in a 2D output feature map. Formally, given a two-dimensional input I and a kernel K , the convolution is computed as:

$$F_{i,j} = \sum_m \sum_n I_{i+m,j+n} \cdot K_{m,n} \quad (2.19)$$

where i and j are the indices of the output feature map. The convolution operation can be further modulated by adjusting the stride, which determines the kernel's movement across the input. A stride of 1 indicates that the kernel shifts by one pixel at a time, while a larger stride (e.g., 2 or 3) causes the kernel to skip pixels, resulting in a smaller output feature map. Formally, we can modify our convolution equation to incorporate stride:

$$F_{i,j} = \sum_m \sum_n I_{s \cdot i + m, s \cdot j + n} \cdot K_{m,n} \quad (2.20)$$

where s represents the stride. This adjustment allows for more efficient computation and can aid in reducing the spatial dimensions of the feature maps, which is particularly beneficial in deep networks.

It is worth noting that despite being called a *convolutional layer*, this operation does not perform a true convolution but rather a *cross-correlation*. This means that the kernel is not flipped. While this is a minor detail, it is important to recognize that in practice, the terms *convolution* and *cross-correlation* are often used interchangeably.

Pooling Layers A pooling layer is employed to reduce the dimensionality of the output feature by replacing the output at certain locations with a summary statistic of the neighboring outputs. The most commonly used operation is *max-pooling*, which replaces the output at a given location with the maximum value of the neighboring outputs. Other pooling operations include *average pooling* and *L^2 -norm pooling*.

In all cases, pooling renders the representation approximately invariant to small translations of the input. Invariance to translation implies that if we translate the input by a small amount, the values of most of the pooled outputs remain unchanged.

In summary, the convolutional layer can be conceptualized as a sequence of convolutional layers, followed by a non-linear activation function and a pooling layer:

$$H = P(\sigma(C(X))) \quad (2.21)$$

where C represents a convolutional layer, P denotes a pooling layer, and σ is a non-linear activation function.

2.3.1. Deep and Residual Neural Network

Residual networks represent a class of neural network architectures that employ residual blocks to enhance the training of very deep networks. A crucial observation by He *et al.* [61] is that increasing the number of layers in a network does not always lead to improved performance. In fact, performance often deteriorates for very deep networks. This phenomenon occurs because the additional layers become difficult to train, and the network becomes more susceptible to overfitting. However, modifying

the convolutional layers to model a perturbation of the input to the output effectively addresses this issue:

$$\mathbf{H} = P(\mathbf{X} + \sigma(C(\mathbf{X}))) \quad (2.22)$$

This concept aligns with the perspective that a deep learning network represents a discretization of an underlying continuous dynamical system, modeled as an ordinary differential equation. Crucially, learning a dynamical system by modeling its velocity proves to be significantly easier than directly learning its position. In the learning scenario, this approach leads to a more favorable optimization landscape and enables the training of much deeper networks.

Residual connections have become a standard feature in modern architectures, with the Residual Neural Networks (ResNet) serving as the seminal work in this domain.

Residual Neural Networks (ResNet) Residual blocks were initially introduced in convolutional networks for image classification tasks. The resulting architectures, known as residual networks or ResNet [61], incorporate these blocks as fundamental components. In ResNets, each residual block comprises a batch normalization layer, a ReLU activation function, and a convolutional layer, followed by an identity mapping of the input to the output.

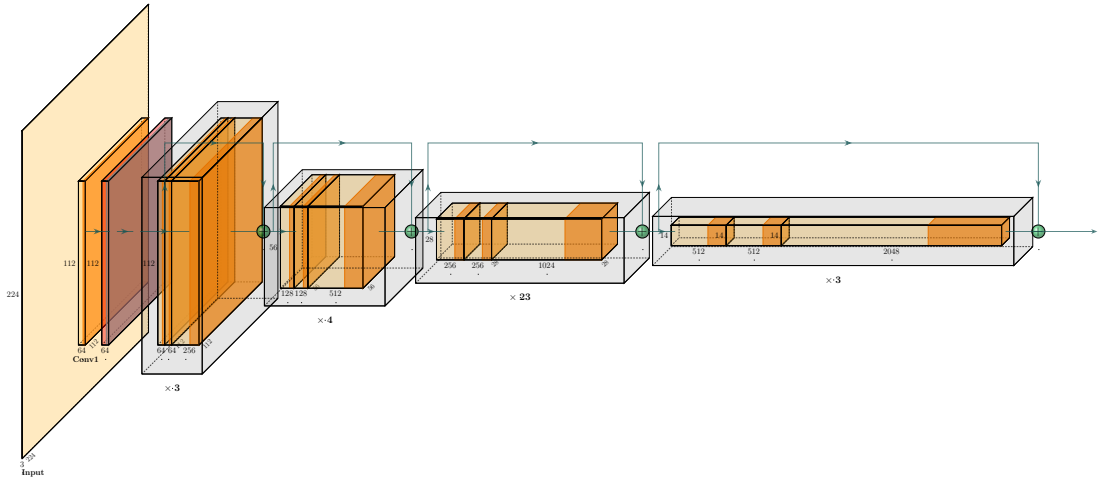


Figure 2.2.: **ResNet-101 Architecture.** Schematic representation of a ResNet-101 architecture.

As network depth increases, the number of parameters becomes prohibitively large, rendering training challenging. To address this issue, *Bottleneck Res Layers* were introduced to optimize parameter utilization through a three-convolution structure. The first convolution, a 1×1 operation, reduces the channel dimensionality. This is followed by a standard 3×3 convolution, and finally, another 1×1 convolution that restores the original channel size. This configuration enables the integration of information over a 3×3 pixel area while minimizing the parameter count.

The ResNet-101 architecture (Figure 2.2) consists of 101 layers. Spatial resolution is downsampled using convolutions with a stride of 2. Channel dimensionality is increased either by zero-padding or through an additional 1×1 convolution. The network initiates with a 7×7 convolution employing a stride of 2 to reduce spatial resolution.

2.4. Graph Neural Networks

Graph Neural Networks (GNNs) represent a class of deep learning architectures designed to process graph-structured data. This section aims to elucidate the concept of GNNs and demonstrate their relationship to other deep learning architectures. The following discussion is based on the work of Bronstein et al. [15].

Graphs A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as a collection of *nodes* $\mathcal{V}$ and *edges* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ connecting pairs of nodes. In the context of deep learning, each node is associated with a d -dimensional *node feature* vector $\mathbf{x}_u \in \mathbb{R}^d$ for all $u \in \mathcal{V}$. These node features can be aggregated into an $n \times d$ matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$. The edges are represented by an $n \times n$ adjacency matrix $\mathbf{A}$, defined as:

$$\mathbf{A}_{uv} = \begin{cases} 1 & \text{if } (u, v) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (2.23)$$

It is important to note that the adjacency matrix $\mathbf{A}$ and feature matrix $\mathbf{X}$ are "synchronized", meaning that $\mathbf{A}_{uv}$ corresponds to the feature vector $\mathbf{x}_v$ of node v connected to node u .

The *neighborhood* of node u , also referred to as the 1-hop neighborhood, is defined as:

$$\mathcal{N}_u = \{v \in \mathcal{V} \mid (u, v) \in \mathcal{E}\} \quad (2.24)$$

Consequently, the neighborhood feature matrix of node u is expressed as:

$$\mathbf{X}_{\mathcal{N}_u} = \{\mathbf{x}_v \mid v \in \mathcal{N}_u\}. \quad (2.25)$$

2.4.1. Graph Neural Network Layers

GNN Layers are functions $F(\mathbf{X}, \mathbf{A})$ constructed by applying shared functions $\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u}, \mathbf{A})$ over local neighborhoods of nodes. This local function ϕ is often referred to as the *message function* or *propagation*.

The literature presents a multitude of GNN layer types. To provide a comprehensive overview, we can categorize them into different classes (Figure 2.3). All types extend the local neighborhood function ϕ by allowing for greater flexibility in node feature processing. Furthermore, all types aggregate information from $\mathcal{N}_u$ (potentially transformed by some function ψ) and combine it using an aggregation function

$\oplus$, subsequently updating the features of node u through a function ϕ . Typically, $\oplus$ is a non-parametric sum, mean, or max aggregation function, while ψ and ϕ are learnable functions. The most common formulation is $\psi(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$ and $\phi(\mathbf{x}, \mathbf{z}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{U}\mathbf{z} + \mathbf{b})$, where $\mathbf{W}, \mathbf{U}, \mathbf{b}$ are learnable parameters and σ is a non-linear activation function. The additional input of $\mathbf{x}_u$ to ϕ represents an optional *residual* connection (cf. Section 2.3.1).

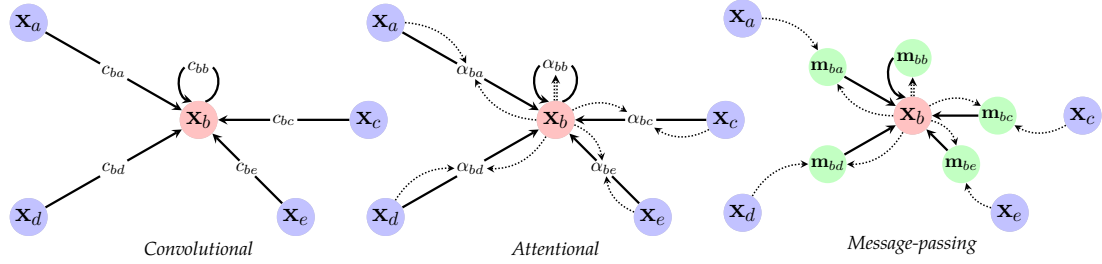


Figure 2.3.: **Graph Neural Network Layer Types.** Schematic representation of different graph neural network layer types. **Convolutional**: sender node features are multiplied with a constant, c_{uv} . **Attention**: Attention layer. **Pooling**: Pooling layer.

For **convolutional** types [83, 165], the features of the neighborhood nodes are directly aggregated with fixed weights:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv} \psi(\mathbf{x}_v) \right), \quad (2.26)$$

where c_{uv} represents the importance of node v to u 's representation. It is a constant and often directly depends on $\mathbf{A}$. If the aggregation function $\bigoplus$ is a simple summation, it can be considered a linear combination of the neighborhood node features.

For **attention** types [151], the importance of the neighborhood nodes is implicitly determined:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right), \quad (2.27)$$

where a is a learnable *self-attention mechanism* that computes the importance coefficients $\alpha_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$ implicitly. When $\bigoplus$ is a simple summation, the aggregation remains a linear combination of the neighborhood node features, but the weights are feature-dependent.

Finally, the **message-passing** types [50] compute arbitrary vectors (messages) across edges:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v) \right), \quad (2.28)$$

where ψ is a learnable message function, computing the vector sent from v to u , and the aggregation can be considered a form of message passing on the graph.

It is crucial to note the representational containment between these approaches: *convolutional* $\subseteq$ *attention* $\subseteq$ *message-passing*. Indeed, attentional GNNs can represent a convolutional GNN by setting $a(\mathbf{x}_u, \mathbf{x}_v) = c_{uv}$. Both are special cases of the more general message-passing GNN where the message function ψ is a simple linear transformation.

Later in this work (*cf.* Section 4.4.1), we will examine the relationship between CNNs and GNNs in the context of a spatial-temporal skeleton-graph. The following section will also provide an intuition regarding the similarity between GNNs and Transformer architectures.

2.5. Vision Transformer Architectures

Transformers, introduced by Vaswani et al. [150] in 2017, revolutionized the field of natural language processing (NLP). This novel architecture, based on the concept of self-attention, enabled efficient processing of long-range dependencies in sequential data, overcoming limitations of previous recurrent and convolutional models. Transformers rapidly became the de facto standard for various language tasks, including machine translation, text summarization, and question answering, achieving state-of-the-art results across the board.

The success of Transformers in NLP inspired researchers to adapt this architecture for computer vision tasks. In 2020, Dosovitskiy et al. [34] introduced the Vision Transformer (ViT), demonstrating that a pure transformer applied directly to sequences of image patches can perform remarkably well on image classification tasks. The ViT architecture treats an image as a sequence of fixed-size patches, linearly embeds each of them, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder.

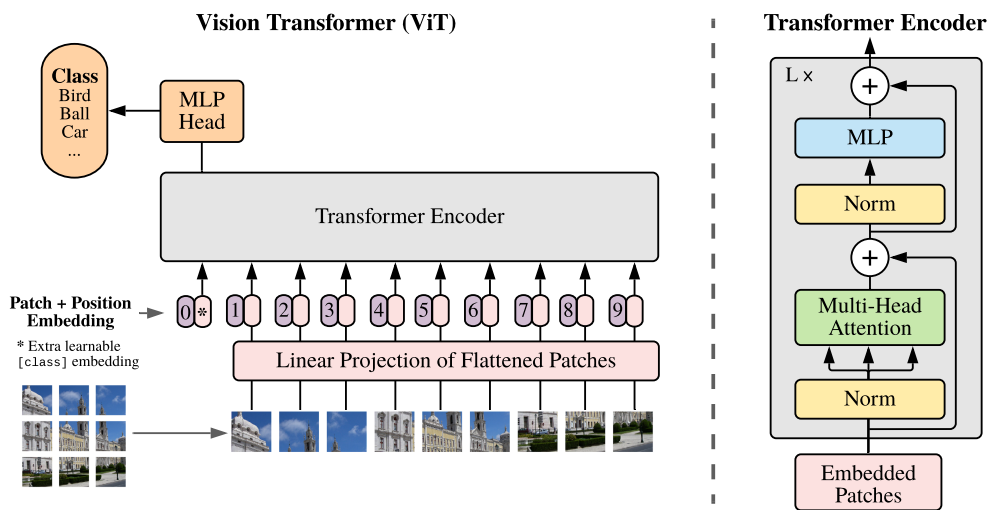


Figure 2.4.: **Vision Transformer Model.** Schematic representation of a Vision Transformer model.

Figure 2.4 illustrates the Vision Transformer (ViT) model scheme. The standard Transformer [150] receives a 1D sequence of token embeddings. For 2D images, the image $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is divided into fixed-size patches $\mathbf{X}_p \in \mathbb{R}^{N \times (P^2 \times C)}$ where (P, P) is the patch size and $N = \frac{HW}{P^2}$ is the number of patches. These patches serve as the input to the Transformer encoder. The Transformer employs a constant latent vector size D for all layers, thus all patches are flattened into an $N \times D$ matrix using a linear projection $\mathbf{E}$ (Equation (2.29)).

Additionally, the patch sequence is prepended by a learnable embedding $\mathbf{z}_0^{(0)} = \mathbf{X}_{class}$ that represents the global image information used for classification. Position embeddings are added to the patch embeddings to preserve the spatial information of the image:

$$\mathbf{z}^{(0)} = [\mathbf{X}_{class}; \mathbf{X}_p^1 \mathbf{E}; \mathbf{X}_p^2 \mathbf{E}; \dots; \mathbf{X}_p^N \mathbf{E}] + \mathbf{E}_{pos} \quad (2.29)$$

The Transformer encoder [150] consists of multiple layers, each containing alternating Multi-Head Self-Attention (MSA) mechanisms and MLP blocks. Layer normalization is applied after each MSA and MLP block.

$$\mathbf{z}^{(l)} = \text{MSA}(\text{LN}(\mathbf{z}^{(l-1)})) + \mathbf{z}^{(l-1)} \quad (2.30)$$

$$\mathbf{z}^{(l)} = \text{MLP}(\text{LN}(\mathbf{z}^{(l)})) + \mathbf{z}^{(l)} \quad (2.31)$$

The output of the final layer is utilized for image classification:

$$\mathbf{y} = \text{LN}(\mathbf{z}_0^{(L)}) \quad (2.32)$$

It is noteworthy that Vision Transformers possess significantly less image-specific inductive bias compared to CNNs. While CNNs incorporate locality, 2D neighborhood structure, and translation equivariance throughout their layers, ViTs only have local and translationally equivariant MLP layers, with global self-attention layers. The 2D structure is minimally utilized, primarily when dividing the image into patches and adjusting position embeddings during fine-tuning. Initially, position embeddings contain no 2D positional information, necessitating the model to learn spatial relationships from scratch.

Multi-Head Self-Attention The standard qkv self-attention mechanism [150] has emerged as a fundamental building block for diverse vision tasks. For each element in the input sequence $\mathbf{z} \in \mathbb{R}^{N \times D}$, a weighted sum over all values $\mathbf{v}$ is computed. The attention weights $\mathbf{A}_{ij}$ are derived from the pairwise similarity between queries $\mathbf{q}_i$ and keys $\mathbf{k}_j$.

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z} \mathbf{U}^{qkv} \quad \mathbf{U}^{qkv} \in \mathbb{R}^{D \times 3D_h} \quad (2.33)$$

$$\mathbf{A} = \text{softmax}(\mathbf{Q} \mathbf{K}^T / \sqrt{D}) \quad \mathbf{A} \in \mathbb{R}^{N \times N} \quad (2.34)$$

$$\text{SA}(\mathbf{z}) = \mathbf{A} \mathbf{v} \quad (2.35)$$

Multi-Head Self-Attention (MSA) is an extension of the self-attention mechanism that employs k self-attention "heads" in parallel, and projects their concatenated output through a linear layer. To maintain consistent computational complexity and parameter count across different values of k , the hidden dimension D_h is set to D/k .

$$\text{MSA}(z) = [\text{SA}^{(1)}(z); \text{SA}^{(2)}(z); \dots; \text{SA}^{(k)}(z)]U^{msa} \quad U^{msa} \in \mathbb{R}^{kD_h \times D} \quad (2.36)$$

2.5.1. Swin Transformer

The Swin Transformer [102] is a hierarchical vision Transformer that addresses certain limitations of the original ViT. It introduces a novel attention mechanism, termed *shifted window-based self-attention*, which computes self-attention within local windows and shifts these windows between consecutive layers (*cf.* Figure 2.5b). This approach enables cross-window connections while maintaining computational efficiency.

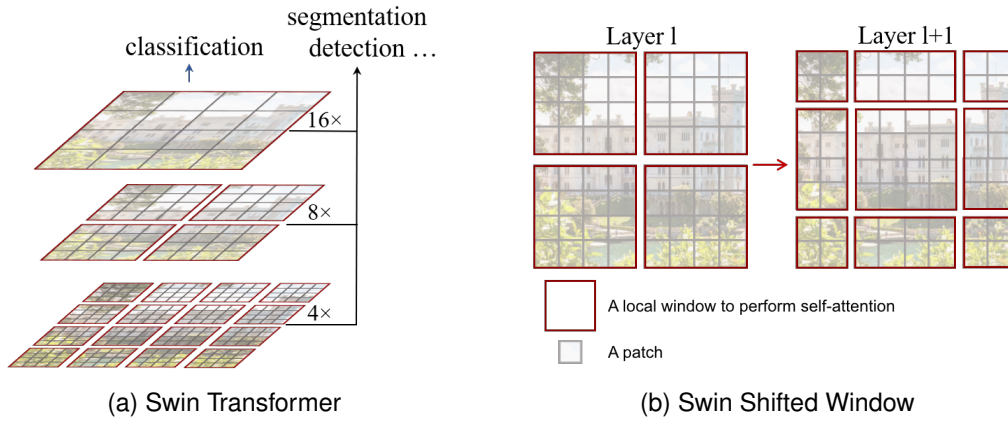


Figure 2.5.: **Swin Transformer.** Visualization of a Swin Transformer architecture. (a) The Swin Transformer architecture utilizes a hierarchical feature map (b) The attention mechanism is applied within local windows that are shifted between layers.

The Swin Transformer architecture is designed to be more efficient than the ViT architecture. It achieves this by employing a hierarchical feature map, where the feature map is progressively merged in deeper layers (*cf.* Figure 2.5a). This allows the model to capture both fine and coarse-grained features. Consequently, it is more suitable for tasks that require dense predictions, such as object detection and semantic segmentation. Additionally, its similarity to most CNN-based architectures allows it to be used as a drop-in replacement, even with a feature pyramid network [98].

The Swin Transformer utilizes the same Transformer encoder as the ViT architecture. However, it replaces the global self-attention with a local window-based self-attention. The feature map is divided into non-overlapping windows, each containing $M \times M$

patches. To address the lack of attention between these windows, the Swin Transformer shifts the windows between consecutive layers. For instance, as illustrated in Figure 2.5b, the windows in layer l start at the top left, and the 8×8 features are evenly divided into four windows of 2×2 with a size of 4×4 ($M = 4$). Subsequently, layer $l + 1$ displaces the window by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels from the regularly partitioned windows. This shift is repeated in layer $l + 2$ and so on. It has been demonstrated that this shift introduces connections between non-overlapping windows of the previous layer.

The shifted window-based attention mechanism exhibits linear scaling with image size $\mathcal{O}(4HWC^2 + 2M^2HWC)$, in contrast to the quadratic complexity of ViT's global attention $\mathcal{O}(4HWC^2 + 2(HW)^2C)$.

In summary, the Swin Transformer architecture presents a more efficient and flexible alternative to the ViT architecture, demonstrating superior performance.

2.5.2. Deformable Attention

Deformable Attention [182] is an attention mechanism designed to enhance efficiency compared to standard attention mechanisms. The primary challenge in applying attention mechanisms to image feature maps is the necessity to consider all possible spatial locations in the feature map. The deformable attention mechanism addresses this by attending to only a small set of key sampling points around a reference point, irrespective of the feature map size. By sampling a fixed number of keys for each query, the deformable attention mechanism achieves greater efficiency than standard attention mechanisms, particularly for large feature maps. Figure 2.6 illustrates the deformable attention module.

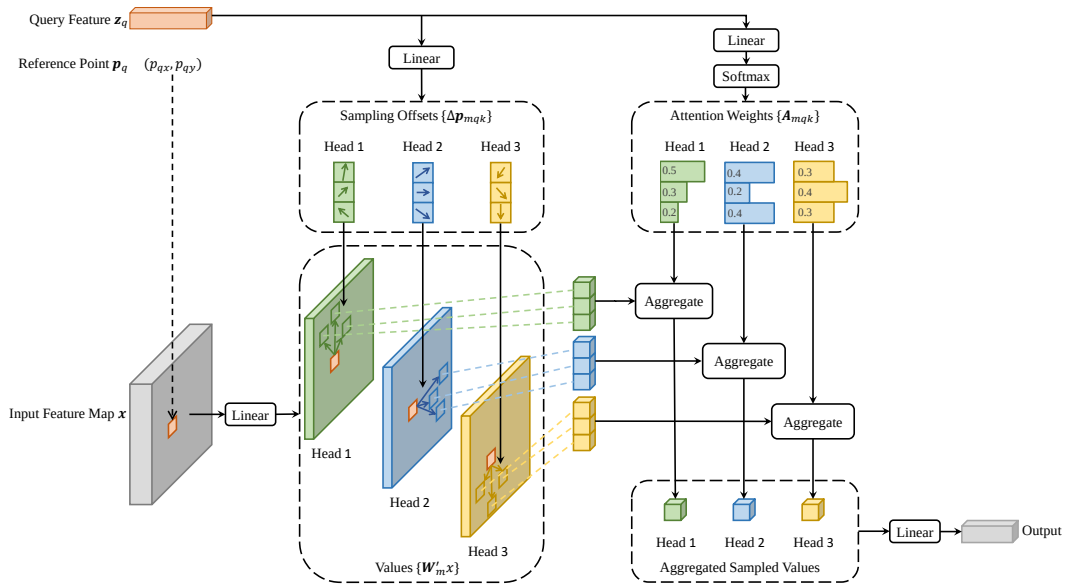


Figure 2.6.: **Deformable Attention.** Visualization of a Deformable Attention mechanism.

For an input feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, the feature z_p at a specific spatial location $p = (x, y)$ and a 2D reference point $\mathbf{p}_q$, the deformable attention feature is computed as:

$$\text{DeformAttn}(z_q, \mathbf{p}_q, \mathbf{X}) = \sum_{m=1}^M \mathbf{W}_m \left[\sum_{k=1}^K \mathbf{A}_{mqk} \cdot \mathbf{W}'_m \mathbf{X}(\mathbf{p}_q + \Delta \mathbf{p}_{mqk}) \right], \quad (2.37)$$

where m denotes the index of the attention head, K represents the number of sampling points ($K \ll H \times W$), $\mathbf{A}_{mqk}$ is the attention weight, $\Delta \mathbf{p}_{mqk}$ is the offset of the k -th sampling point relative to the reference point, and $\mathbf{W}_m$ and $\mathbf{W}'_m$ are the weight matrices for the attention mechanism. The attention weight $\mathbf{A}_{mqk}$ is constrained to the range $[0, 1]$ and normalized such that $\sum_{k=1}^K \mathbf{A}_{mqk} = 1$. $\Delta \mathbf{p}_{mqk} \in \mathbb{R}^2$ is unconstrained in range. Both $\mathbf{A}_{mqk}$ and $\Delta \mathbf{p}_{mqk}$ are obtained through an MLP conditioned on the query feature z_q .

The deformable attention mechanism is applied to the feature map as key elements. Let N_q be the number of query points in the feature map, with $N_q \ll HW$. When $M \cdot K$ are relatively small, the complexity of the deformable attention mechanism is $\mathcal{O}(2N_q C^2 + N_q M C^2)$, compared to $\mathcal{O}(4HWC^2 + 2(HW)^2 C)$ for the standard attention mechanism.

2.6. Projective Geometry

Projective geometry is a fundamental concept in computer vision and image processing, providing a mathematical framework for understanding the projection of the 3D world onto a 2D image plane. Central to this framework is the camera model, which describes the mapping of 3D points in the world to 2D points on an image.

The pinhole camera model (*cf.* Figure 2.7) is the most widely used camera model in computer vision. This model assumes that all light rays pass through a single point (the pinhole) before intersecting the image plane. Despite its simplicity, this model effectively approximates the behavior of most real-world cameras.

In projective geometry, homogeneous coordinates are employed to represent points and lines. Homogeneous coordinates augment Euclidean coordinates with an additional dimension, enabling the representation of points at infinity and simplifying numerous geometric operations. A 3D point (X, Y, Z) is represented as $[X, Y, Z, 1]$ in homogeneous coordinates, while a 2D point (x, y) becomes $[x, y, 1]$. This representation simplifies projective transformation equations, transforming them into linear operations that can be expressed as matrix multiplications.

2.6.1. Camera Projection

Cameras translate the three-dimensional world into two-dimensional pixels, but modeling the relationship between a 3D point in the world and its corresponding 2D

projection on the image plane is a non-trivial task. The world is measured in metric units, while the image is measured in pixels. The pinhole camera model provides a mathematical description of how a camera projects 3D points onto a 2D image plane. It is a simple yet widely used camera model in computer vision and image processing.

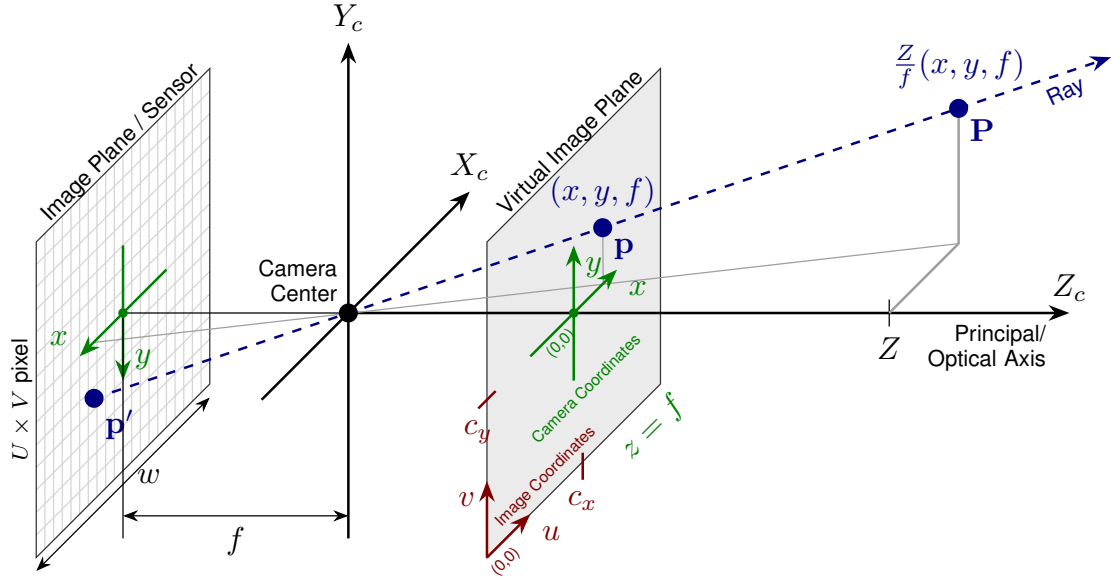


Figure 2.7.: **Pinhole Camera Geometry.** A 3D point P projects onto the location p in the projection plane, situated at a distance f from the pinhole or camera center. The virtual camera plane is a radially symmetric projection of the camera plane. The virtual camera plane also illustrates the relationship between the camera (x, y) and the image coordinate system (u, v) . The sensor has a physical width of w and $U \times V$ pixels.

Figure 2.7 illustrates the pinhole camera model and the projection of a 3D point P in the world to a 2D point p on the image plane. We initially consider the camera center as the origin of a Euclidean coordinate system, with the image plane $z = f$ being the plane where the image is formed. The focal length f represents the distance between the camera center and the image plane. Through similarity of triangles, we can derive that a point $(X, Y, Z)^T$ is mapped to point $(fX/Z, fY/Z, f)^T$ on the image plane.

Utilizing homogeneous coordinates, we can represent the projection of a 3D point P in the world to a 2D point p on the image plane as a matrix multiplication:

$$KP = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.38)$$

Camera Intrinsic Parameters The camera intrinsics establish the mapping from metric space to pixels. Figure 2.7 illustrates the image formation process and image

sampling by the sensor. Thus, the projection must account for the perspective projection of the point $\mathbf{p}_c$, described in camera coordinates, to pixel coordinates on the sensor plane. Consequently, we replace the focal length from Equation (2.38) with α :

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.39)$$

where the constants α and β are related to physical parameters of the camera:

$$\alpha = f \frac{U}{w} \quad \text{and} \quad \beta = f \frac{V}{h} \quad (2.40)$$

where f is the focal length (in meters), w and h are the width and height of the sensor (in meters), and U and V are the image width and height (in pixels). If the pixel aspect ratio is 1, then $\alpha = \beta$.

Additionally, the image center (u_0, v_0) is conventionally set to either the top left or bottom left of the image. Thus, we translate the origin of the camera coordinate system to the image center. This results in the following camera matrix:

$$\mathbf{K} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.41)$$

For a calibrated camera, the camera intrinsics can also be used to project a ray for a point on the image plane. The ray connecting the camera center and the point $\mathbf{p}$ is represented by $\lambda(x, y, f)$, where λ is a positive real number. Any 3D point along this line will project to the same point $\mathbf{p}$ on the image plane. Figure 2.7 illustrates this ray with the scaling factor $\lambda = Z/f$:

$$\frac{Z}{f} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.42)$$

It is important to note that when utilizing pixel coordinates, f must be substituted with α .

Camera Extrinsic Parameters While the intrinsic parameters describe the mapping between a point $\mathbf{P}$ in the 3D camera reference system and a point $\mathbf{p}$ in the 2D image plane, it is also necessary to account for the camera's position and orientation within the global reference system. This is accomplished through the extrinsic parameters.

The extrinsic transformation is defined by a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, where $SO(3)$ denotes the special orthogonal group of 3D rotations.

Consequently, for a point $\mathbf{P}_W$ in world coordinates, its corresponding coordinates in the camera reference system $\mathbf{P}_C$ are given by:

$$\mathbf{P}_C = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \mathbf{P}_W \quad (2.43)$$

The parameters $\mathbf{R}$ and $\mathbf{t}$ constitute the extrinsic parameters of the camera.

Homography Homography is a fundamental concept in computer vision and image processing, representing a transformation that maps points from one plane to another while preserving collinearity. This concept finds applications in various tasks, including image stitching and the creation of Bird's Eye Views (BEVs) when observing a planar surface from a non-coplanar perspective.

A homography, also known as a perspective projection, is a geometric transformation that preserves straight lines. The homography function h maps points to points, $\mathbf{p}' = h(\mathbf{p})$, with the property that any three collinear points $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ in the image plane remain collinear after the transformation. In homogeneous coordinates, this function $\mathbf{p}' = h(\mathbf{p})$ can be expressed as a matrix multiplication:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (2.44)$$

or in concise form as:

$$\mathbf{p}' = \mathbf{H}\mathbf{p}, \quad (2.45)$$

where $\mathbf{H}$ is a 3×3 matrix and $\mathbf{p}$ and $\mathbf{p}'$ are 2D points in homogeneous coordinates.

2.7. Epipolar Geometry

While the previous sections focused on the projection of points from 3D to 2D, computer vision often involves working with image pairs, such as when observing a scene from different perspectives. In these scenarios, it becomes necessary to consider the relationship between points in two images. This is the domain of epipolar geometry, which enables the reconstruction of 3D scene structure from two or more images.

For two calibrated cameras (*i.e.* with known intrinsic and extrinsic parameters), their relationship is defined by a generic translation $\mathbf{t}$ and a rotation matrix $\mathbf{R}$, such that their optical/principal axes are parallel. A point observed in camera 1, denoted as $\mathbf{p}_1$, corresponds to a 3D point lying on a ray connecting the camera 1 origin $\mathbf{O}_1$ and $\mathbf{p}_1$. Consequently, $\mathbf{p}_2$ is constrained to lie on the line resulting from the projection of this ray into camera 2. This line is referred to as the **epipolar line**. To project the ray into camera 2, the camera calibration is utilized to transform the camera 1 origin $\mathbf{O}_1$ and the point $\mathbf{p}_1$ into camera 2 coordinates. These projected points define the equation of the epipolar line in camera 2.

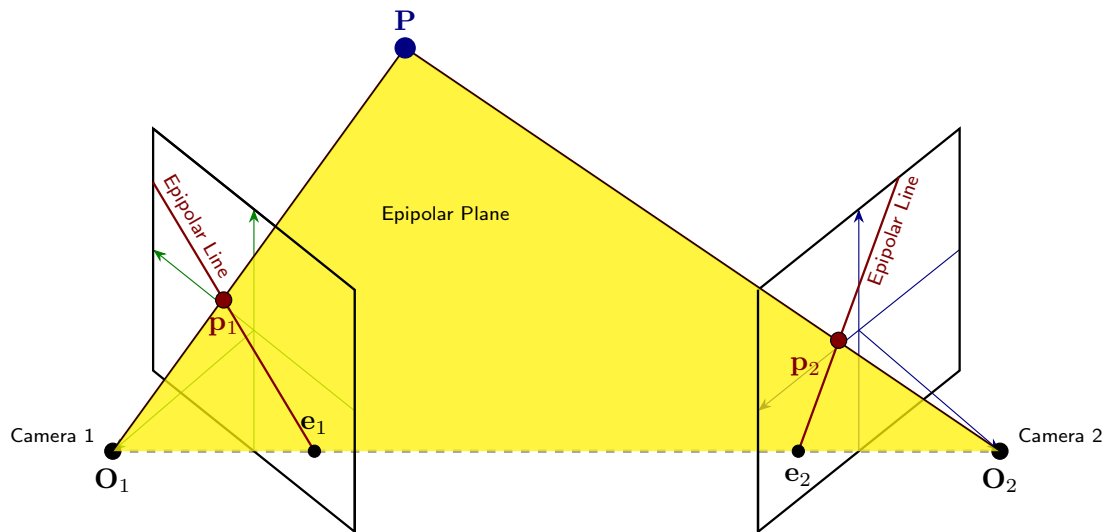


Figure 2.8.: **Epipolar Geometry.** Illustration of the geometry of a stereo pair and associated terminology. A point P in space is observed by both camera 1 and camera 2, with origins O_1 and O_2 , respectively. The line connecting the camera centers is termed the baseline. As the point P moves in space, the epipoles (e_1 and e_2) remain constant.

Figure 2.8 illustrates the geometry of a stereo pair. Given a 3D point P in space, the following terms are defined:

- The **epipolar plane** is the plane containing the two camera origins and the point P .
- The **epipolar lines** are the intersections of the epipolar plane with the image planes of the two cameras.
- The **epipoles** are the intersection points of the line connecting the camera origins with the image planes.

The Essential Matrix is defined to relate the two image planes and is crucial for the classical reconstruction of points in space from two images. In this work, however, we will focus on solving the correspondence of points in 3D space by intersecting and associating rays from multiple cameras.

3

Multi-View

3D Tracking & Detection

Multi-Target Multi-Camera (MTMC) tracking has remained a specialized domain within the tracking research community, often overshadowed by the more prevalent Multiple Object Tracking (MOT) task. The utilization of multiple cameras to mitigate challenges associated with occlusion and missed detections was initially introduced in one of the seminal modern tracking datasets, PETS2009 [46]. In recent years, however, the MTMC task has garnered increased attention, coinciding with the proliferation of multi-camera deployments in natural environments and the emergence of more comprehensive MTMC datasets [29, 147, 70, 3[†], 51]. Contemporary methodologies have been developed for tracking both pedestrians [22, 70, 147] and vehicles [3[†]]. This paper aims to bridge these two domains by presenting a unified approach that demonstrates superior performance across three public datasets: the well-established pedestrian datasets Wildtrack [22] and MultiviewX [70], and the vehicle dataset Synthehicle [3[†]]. The commonly utilized pedestrian datasets, Wildtrack and MultiviewX, exhibit certain limitations for advanced computer vision research, primarily due to their single-scene composition and their propensity for overfitting, given their training-testing frame split (90% training, 10% testing). Consequently, we assess a novel, more complex dataset. The Synthehicle dataset [3[†]] is a synthetic roadside dataset encompassing a diversity of intersection scenarios for vehicle tracking, necessitating methodological adaptability to various unobserved scenes.

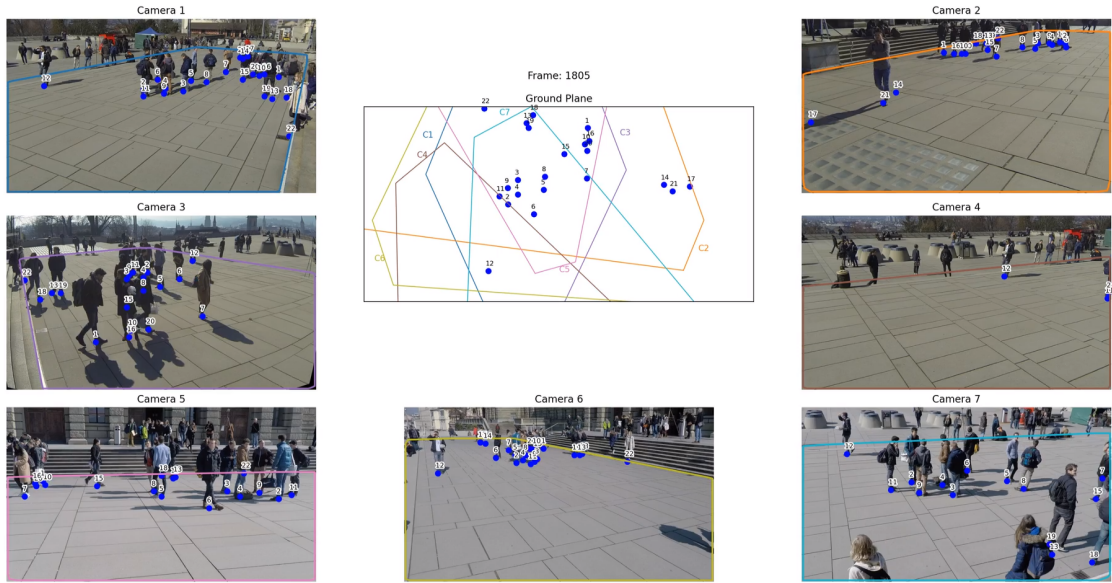


Figure 3.1.: **Example of a Multi-View Scene.** A sample scene from the Wildtrack dataset [22] that is captured by seven cameras. In each camera, the pedestrian IDs are annotated on the ground plane. In the center the BEV projection of the scene is shown with the pedestrian IDs and each camera’s field of view.

In comparison to the single-camera task, the MTMC task presents heightened complexities, requiring bifurcated association steps: (1) inter-camera association and (2) intra-detection association. Conversely, multi-camera methodologies facilitate more robust 3D perception, enabling object position triangulation from multiple camera feeds. Historically, MTMC approaches [67, 3[†], 110] initiate with 2D bounding box detection and subsequently create camera-specific tracklets which are then globally associated. Some methodologies have employed 2D detection to infer 3D location, performing the association in three-dimensional space [27]. More recently, novel approaches have bypassed the 2D detection phase, opting to directly project features into Bird’s Eye View (BEV), thereby conducting both detection and tracking in 3D [10[†]]. Empirical evidence suggests that early data fusion in 3D space yields improved outcomes over late-fusion techniques. Adopting this early-fusion paradigm, we propose an innovative architecture employing advanced projection methods to accurately map features within a 3D framework.

The projection of camera features has been extensively studied in perception models [92, 56, 118] for autonomous vehicles and has become an integral component of multisensor perception systems. While sensors such as Lidar and Radar inherently provide depth information, camera images represent a projection of the 3D world onto a 2D plane, with no trivial inverse transformation. Consequently, lifting methods have been developed to recover 3D information. Projection is also crucial for fusing camera data with other sensor modalities [12[†]].

However, there are significant distinctions between ego-centric vehicle perception and multi-view detection: (1) The cameras in our approach have a larger overlapping field of view, enabling the aggregation of information from multiple perspectives. (2) Our cameras are predominantly static, allowing for the exploitation of scene geometry. (3) The observed area is substantially larger, and the cameras are positioned at greater distances from the targets. Nevertheless, the overlap provides a key advantage: we can triangulate the 3D position of targets using an approximation to epipolar geometry [58]. An exemplar scene illustrating this is presented in Figure 3.1. This scene also highlights the primary challenge: the association of detections across cameras. Each of the seven camera views captures a subset of all pedestrians, and the task is to establish correspondences between detections across the cameras.

While tracking is also a component of autonomous perception, the challenge is more manageable with static cameras, as targets are confined to a small surrounding area and observed for brief periods. Consequently, most methods [155, 167] employ simple distance-based association techniques. In our approach, we confront more expansive scenes with a higher number of targets that require tracking over extended durations. Continuing with the early-fusion paradigm, we propose a novel association method that utilizes historical BEV information to predict the location of each detection in the previous frame. This method combines the advantages of appearance-based and motion-based association, learning to integrate both cues effectively.

In summary, our contributions are as follows:

- We propose an early-fusion approach for multi-view tracking that projects camera features to the BEV space and performs detection and tracking in 3D. We evaluate various lifting methods and demonstrate that the choice of lifting method significantly impacts tracking performance.
- We introduce the first multi-view single-shot tracker and evaluate two distinct association methods incorporating appearance and motion cues.
- We establish a new, robust baseline on more challenging datasets for MTMC with a standardized evaluation protocol to facilitate further and more comparable research in this domain.

This chapter synthesizes the work presented in our publications EarlyBird [10[†]] and TrackTacular [11[†]], and extends it with a novel lifting method, Gaussian Splatting, and additional experiments. It builds upon the student work of Zitao Wang [6⁺] and Yassine Hamila [1⁺].

3.1. Related Work

In this section, we provide a comprehensive review of the foundational literature underpinning our work. Given that the tracking process initiates with detection, we

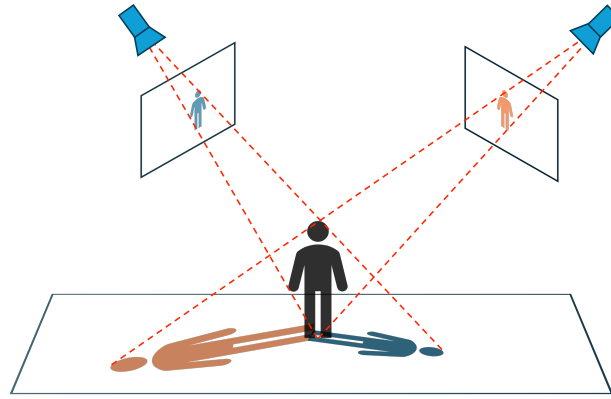


Figure 3.2.: **Projection Shadows.** The shadow effect on the ground plane when using perspective projection.

reassess multi-view detectors, which we have augmented with innovative methodologies in 3D perception systems. Following, we give provide a overview of single-view MOT and single-stage trackers. Lastly, review MTMC tracking approaches.

3.1.1. Multi-View Object Detection

Multi-camera systems are widely used to tackle the challenge of pedestrian detection in highly occluded environments. Such systems comprise synchronized and calibrated cameras that capture a common area from various angles. The multi-view detection system then processes the overlapping views to detect pedestrians.

MVDet [70] introduced a novel approach utilizing convolutions for end-to-end model training, wherein encoded image features from multiple perspectives are projected onto a shared ground plane. This methodology resulted in significant performance enhancements and has since served as a foundation for subsequent research, including our own. The projection technique is not confined to sparse detections from individual viewpoints; rather, the method proposed by [70] encodes the entire input image and projects all features onto the ground plane using perspective transformation. Figure 3.2 illustrates the resultant shadow effect on the ground plane when employing perspective projection.

To address the limitations inherent in perspective transformation, several alternative methodologies have been proposed [69, 87, 140]. One such approach [69] employs projection-aware transformers with deformable attention in the BEV-space to rectify the "shadow" artifacts by repositioning them to their original locations. Another method [87] utilizes regions of interest derived from 2D detections and independently projects these to the estimated foot positions on the ground plane. A third approach [140] implements multiple stacked homographies at various heights to more accurately approximate a complete 3D projection.

Diverging from model-centric improvements, [123] sought to enhance detection performance through data augmentation. This approach introduced additional occlusions by incorporating 3D cylindrical objects, thereby reducing the system’s dependence on multiple cameras and mitigating the risk of overfitting.

While our approach fundamentally aligns with the principles established by MVDet [70], and we incorporate their projection method as our baseline, we extend our investigation to explore alternative projection techniques that circumvent the shadow effect. As elucidated in the subsequent section, these advanced projection methods are implemented with the objective of further improving detection performance.

3.1.2. 3D Perception Systems

Multi-sensor perception systems, primarily developed for autonomous vehicles, are designed to fuse data from diverse sensor modalities. The field of autonomous driving has witnessed rapid advancements, particularly in the domain of camera-based feature lifting. This section focuses on approaches that emphasize camera lifting techniques.

The most rudimentary unprojection methods are homography-based approaches [70, 69, 87, 140]. These methods operate under the assumption of a planar ground surface and utilize a homography matrix to project image features onto this plane. While this technique is parameter-free, it exhibits reduced accuracy for objects elevated above the ground plane and generates shadow-like artifacts for objects at considerable distances from the camera [69]. Simple-BEV [56] introduced an alternative parameter-free projection method, which defines a three-dimensional coordinate volume over the BEV plane, projects these coordinates onto all images, and computes the average of features sampled from the projected locations [57]. In contrast to homography-based projection, this method employs a *pull* mechanism, extracting information from the image to a 3D location, rather than *pushing* information from the image to the world reference system.

Depth-based approaches [118] incorporate a monocular depth estimator to approximate pixel-wise depth, facilitating the projection of pixels into 3D space. This method demonstrates high efficacy, as it does not necessitate explicit depth information and can infer it from the scene context. An alternative technique for image feature lifting is BEVFormer [92]. While conceptually similar to Simple-BEV [56] in its aggregation of information from multiple images to a 3D location, BEVFormer employs a deformable attention mechanism to learn aggregation weights. This approach offers enhanced flexibility compared to Simple-BEV [56], enabling focused attention on specific objects, albeit at the cost of increased computational complexity.

In our research, we conduct a comparative analysis of these diverse lifting strategies and extend them to explicitly enforce feature triangulation.

3.1.3. Multi Object Tracking

The paradigm of Tracking-by-Detection (TbD) has emerged as the predominant approach in MOT in recent years [106, 144, 170, 10, 8, 174, 181]. This methodology initially employs object detection techniques to identify pedestrians, rendering the efficacy of tracking highly dependent on the accuracy of these detections. Subsequently, the detected objects are correlated across frames to construct trajectories pertaining to specific identities, utilizing various cues such as motion, location, appearance, or a combination thereof. This association process can be addressed either in an online, frame-by-frame manner or offline, in a track-wise fashion over the entire sequence. In this section, we provide a concise review of the development of single-camera multi-object trackers, noting that our approach utilizes both motion and appearance cues, in contrast to the majority of contemporary research which typically employs complex graph-based associations.

Graph-Based Approaches A prevalent formalism for executing data association within the TbD paradigm involves conceptualizing each detection as a node in a graph, with edges connecting various nodes in the temporal domain to delineate trajectories. The determination of connected nodes can subsequently be addressed through maximum flow [7] or minimum cost techniques [76, 120, 172], for instance, with the incorporation of motion models [85]. Recent advancements have integrated graph-based tracking models with neural networks [14]. These graph models can also incorporate motion or appearance attributes to resolve the graph association problem. While framing tracking as a graph problem is a logical approach, our findings demonstrate that simple motion or appearance cues are sufficient for an effective tracker.

Motion-Based Association In contrast to graph-based approaches, numerous TbD methods perform frame-by-frame association directly using motion and location cues derived from detections and existing trajectories [10, 11, 180, 116, 113]. For short-term preservation, these trackers exploit the tendency for object displacements to be small between adjacent frames. This allows them to utilize spatial proximity for matching, often employing techniques such as Kalman filters [10]. Extending this concept, approaches following the tracking-by-regression paradigm utilize object detectors [8, 181] to regress bounding box positions. Recent advances have introduced Transformer-based models [106, 144, 170] that perform tracking following the tracking-by-attention paradigm. While these approaches, particularly those employing sophisticated motion models, achieve outstanding performance on several datasets, especially in short-term associations, they often require complex training strategies. A recent method [32] improved long-term tracking performance by predicting trajectories on the ground plane. Although this necessitates complex homography estimation in single-camera settings, it demonstrates robust results by tracking in BEV. In contrast to these approaches, we demonstrate that a simple motion model suffices to model short-term associations. However, in scenarios involving moving

cameras or requiring non-trivial long-term associations, such as those with numerous occlusions, purely motion-based trackers encounter difficulties, necessitating a combination with appearance-based cues.

Appearance-Based Association To achieve superior performance in long-term association scenarios, numerous approaches employ additional appearance-based Re-Identification (re-ID) networks that encode appearance cues to re-identify individuals after occlusions [152, 8, 159]. Further exploring this direction, recent work [114] proposed training a detection network utilizing only embedded information during the training process. In pursuit of real-time MOT, several studies have proposed jointly computing detections and embeddings in a multitask setting [163, 53, 157, 104, 174]. Some of these introduce more balanced training schemes [53, 163] to better leverage the synergies between both signals. Although promising, approaches that use appearance to motion cues typically require rather complex association schemes with multiple steps [174, 10].

Person Re-Identification Appearance-based association is strongly coupled to re-ID. The objective of person re-ID is to retrieve bounding boxes of an individual from a large gallery set that depict the same person as a given query image, based on appearance cues [4[†]]. These query and gallery image sets predominantly contain fully visible bounding boxes, which differs from the MOT scenario where pedestrians are often occluded. Numerous approaches [10, 14, 152] utilize off-the-shelf re-ID networks to extract features from bounding boxes and subsequently employ these features for detection association. Single-step trackers [174, 157] adopt a similar approach but train the re-ID network jointly with the detection network, demonstrating enhanced performance. Consequently, we propose learning the association directly from the BEV features, enabling the network to aggregate appearance cues from all views.

3.1.4. Single-Stage Multi-Target Tracking

A subset of multi-object single view tracking encompasses single-stage tracking. These systems conduct detection and tracking in a single step, thereby reducing computational overhead; however, they generally exhibit inferior performance compared to two-step trackers. The predicted features can be re-ID features [174, 152, 157] or motion cues [180, 8, 44].

Track-RCNN [152] represents the first example of a re-ID-based approach. It augments Mask R-CNN [60] with re-ID feature extraction, generating a bounding box and a corresponding re-ID feature for each proposal. Similarly, JDE [157] is built upon YOLOv3 [124], while FairMOT is based on CenterNet [181]. The key advantage of FairMOT, compared to its counterparts, is its anchor-free design, which means that detections do not rely on a fixed set of anchor bounding boxes but on a singular detection point, enhancing the separation of re-ID features.

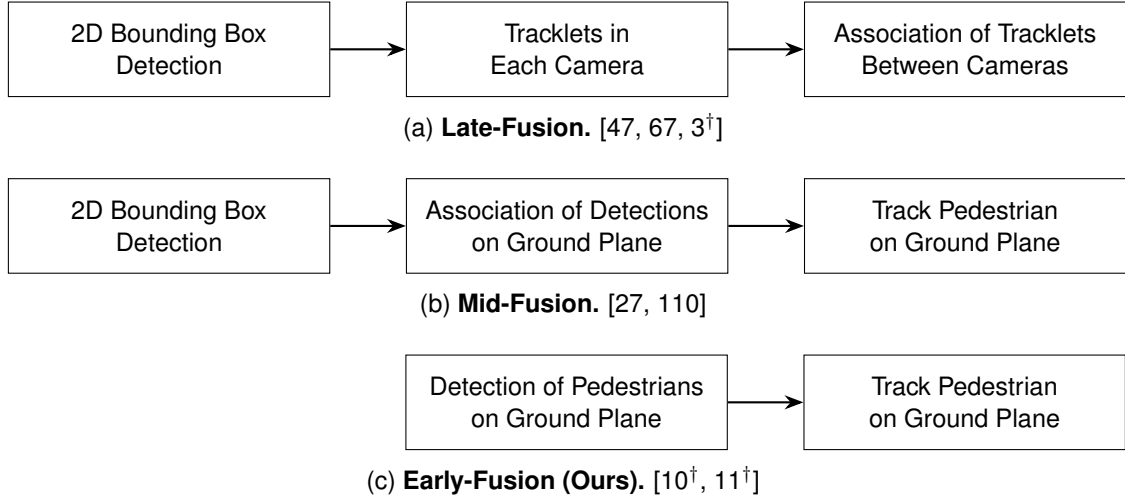


Figure 3.3.: **Fusion Strategies for MTMC.** Comparison of fusion strategies for MTMC tracking approaches. While other approaches start with 2D bounding boxes, our approach fuses camera views early and directly operates in BEV.

Motion-based trackers such as D&T [44] process input from adjacent frames, predicting interframe offsets between bounding boxes. Tracktor [8] exploits the bounding box regression head to associate identities, thus eliminating the need for box association. In contrast, CenterTrack [180] predicts the offset of the object’s center using a triplet input, consisting of the current frame, the previous frame, and the heatmap of the last frame detection. This prior heatmap allows objects to be matched anywhere, even in cases of overlapping boxes. However, these motion-based methods, which only associate objects in adjacent frames without re-initializing lost tracks, can encounter difficulties in managing occlusions.

In our approach, we propose to evaluate both association methods. Initially, we implement an appearance-based method that learns a re-ID feature vector for each detection in the BEV space, analogous to FairMOT [174] or JDE [157]. In the extension work, we learn to associate between detections in the previous and current time steps, as proposed by CenterTrack [180] and D&T [44]. However, we apply this concept at the BEV feature level and incorporate temporal context. Our approach combines the advantages of appearance-based and motion-based association and can learn to integrate both cues. Incorporating a Kalman Filter [78] we can also re-initialize briefly lost tracks.

3.1.5. Multi-Target Multi-Camera Tracking

While the previous section addresses single-camera tracking, the following section focuses on MTMC tracking. The majority of MTMC trackers base their models on the assumption of overlapping Field of View (FOV) among cameras. Fleuret et al. [47] utilize this overlapping FOV to represent targets within a Probabilistic Occupancy Map (POM), integrating color and motion characteristics with occupancy probabilities

during the tracking process. Berclaz et al. [7] enhanced this approach by formulating the tracking within POMs as an integer programming problem, optimally solving it using the k-shortest paths (KSP) algorithm.

MTMC tracking can also be conceptualized as a graph problem. Hypergraphs [67] or multi-commodity network flows [136, 86] are employed to establish correspondences across views, which are then resolved using min-cost [67, 136] or branch-and-price algorithms [86].

Recently, a two-stage approach has gained prominence: it commences with generating local tracklets for all targets within each camera, followed by matching local tracklets corresponding to the same target across all cameras. The task of generating local tracks within a single camera, known as single camera MOT, has been extensively investigated [8, 174, 180, 157, 44, 159, 26]. With advancements in object detection techniques, tracking-by-detection [44, 132, 8, 180, 159] has emerged as the preferred method for multi-target tracking. For the second step, various strategies for cross-view data association have been proposed to match local tracklets across different cameras. Some studies [71, 37] employ principles of epipolar geometry to find correspondences based on ground plane locations. In addition to ground plane locations, Xu et al. [162] incorporate appearance features as cues for association.

State-of-the-art models [110, 27] have inverted the initial two steps: 2D detections are first projected onto the 3D ground plane, and a graph is constructed using re-ID node features. These nodes are spatially and temporally assigned at the outset [27] or both assignments are performed simultaneously [110] using graph neural networks for link prediction. Although all current methods [27, 132, 8, 174, 180] are evaluated based on detection results to account for inaccuracies in detection, LMGP [110] utilizes ground truth bounding boxes and thus cannot be directly compared to recent studies.

Of particular note is the AI City Challenge [109, 154], an annual event focusing on multi-camera tracking. The challenge provides two multi-camera datasets: one for vehicle tracking [147] and one for pedestrian tracking [154]. Held in conjunction with the CVPR conference, the challenge showcases the best-performing methods. While most presented methods are not novel in their approach, the challenge provides a valuable benchmark for real-world multi-camera tracking.

Our approach diverges from previous work and aligns more closely with single-stage trackers, which are discussed in the subsequent section. Similar to the latest methods [110, 27], our approach establishes spatial associations within our detector, then proceeds to associate on the ground plane.

3.1.6. 3D Gaussian Splatting

Gaussian Splatting [80, 21] has recently gained prominence in the field of novel view synthesis from multiple views. 3D Gaussian Splatting represents a scene using a set

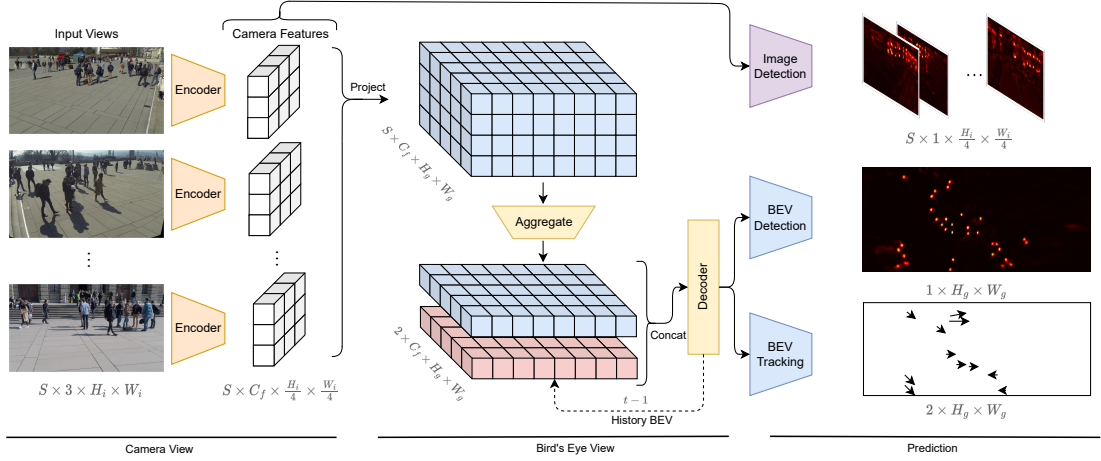


Figure 3.4.: **Overview of Our Approach.** The input views are encoded, and the resulting camera features are projected using one of three lifting methods. After aggregation, the BEV feature is concatenated with the feature of the previous step. With the decoded BEV feature, we predict the locations and offset to the location in the previous step. Additionally, we guide the architecture by predicting the object centers in the image features.

of 3D Gaussian primitives, each with position, size, and color attributes, which are then projected onto 2D images using a renderer. These are successors to NeRFs (Neural Radiance Fields) [108] that employ a neural network to model the scene as a continuous volumetric function, representing color and density at any 3D point. While NeRF requires extensive training and rendering times, 3D Gaussian Splatting offers faster training and real-time rendering, and provides an explicit representation of the scene.

A challenge of 3D Gaussians is that they generally need to be fitted to each scene individually. A recent work [21] addresses this by utilizing 2D pixel-aligned features that predict a 3D Gaussian and generalize over scenes. Each pixel predicts a 3D Gaussian that can then be used to render a photorealistic view of the scene.

In this work, we explore whether 3D Gaussians are also a suitable representation for multi-view detection and tracking. The learnable size of each Gaussian allows for the representation of a wide range of object sizes and different depths.

3.2. Tracking by Detection in the Bird's Eye View

In this section, we present our approach to multi-view tracking. The novelty of our approach is that both detection and tracking start in 3D. In this section, we first introduce the general architecture of our approach and then explain the different components in detail.

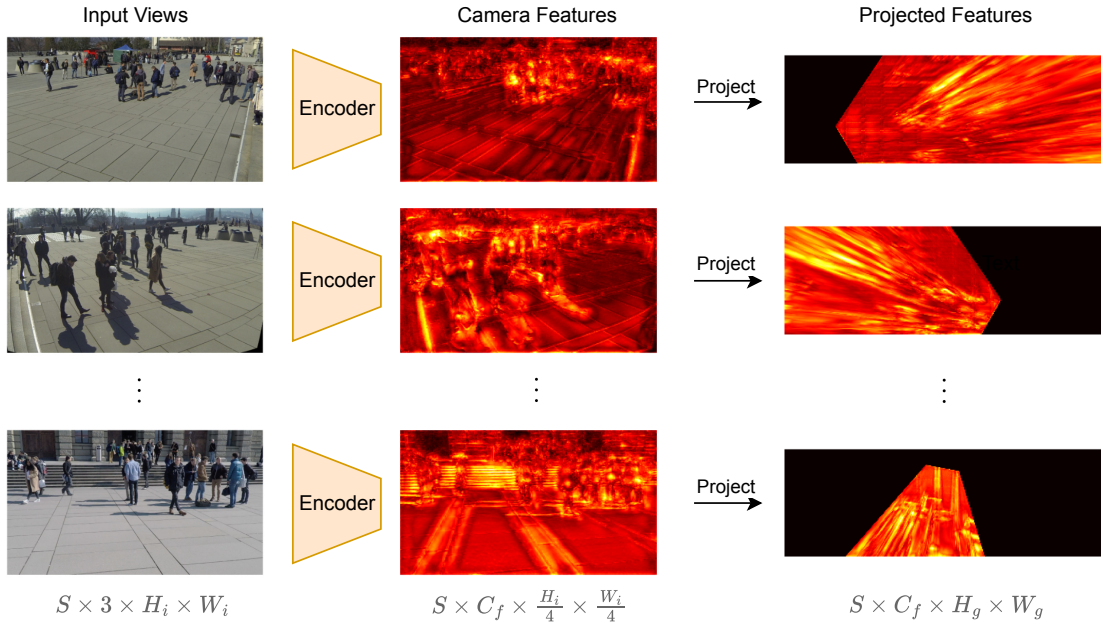


Figure 3.5.: **Visualization of the Feature Encoder.** The input images are encoded with a ResNet or Swin Transformer network. The features are then projected to the BEV space.

3.2.1. Encoder

Our approach processes synchronized RGB input images from S cameras, each with dimensions $3 \times H_i \times W_i$. We employ either ResNet [61] or Swin Transformer [102] networks to encode image features, utilizing three network blocks, each downsampling the input by a factor of 2. This progressive downsampling is crucial for capturing higher-level, more abstract features within the network. However, to project a high-resolution feature map—particularly important for preserving small objects in the image view during projection—we implement a feature pyramid network [98]. This network combines features from different scales through three blocks, each upsampling the feature by a factor of 2 and concatenating it with features from the previous block. The resulting output of the feature pyramid network has dimensions $C_f \times H_f \times W_f$, where $H_f = H_i/4$, $W_f = W_i/4$, and $C_f = 128$. This output is subsequently projected into the BEV space using one of five lifting methods: homography, bilinear sampling, depth splatting, deformable attention, or Gaussian splatting. Figure 3.5 illustrates the encoder architecture and a feature projected using perspective transform.

3.2.2. Temporal Aggregation

Temporal information aggregation forms the core of tracking, and in our approach, this process occurs in the BEV space. The BEV space offers greater stability compared to the image space, where drastic changes can occur between frames (e.g., a person moving very close to the camera). In contrast, the BEV space normalizes all motion in metric space. Our architecture aims to fuse these features at an early stage

rather than at the detection level, as temporal information can enhance detection by ensuring consistency between time steps. Access to previous features enables the architecture to learn the motion patterns of each detection. While trackers are typically categorized as either appearance-based or motion-based, our feature representation incorporates both types of information, allowing our architecture to fuse these cues at the feature level. We implement temporal aggregation using a *late to early* fusion design [62]: the decoded BEV feature from the previous timestep $t - 1$ is concatenated along the channel dimension with the current undecoded feature (cf. Figure 3.4). This temporal aggregation yields a feature tensor of dimensions $(2 \times C_f) \times H_g \times W_g$ for our decoder.

3.2.3. Detection & Tracking Heads

The architecture of our detection and tracking heads is inspired by CenterNet [181], with a training strategy adapted from CornerNet [84]. The primary detection branch generates a heatmap or POM $\hat{Y} \in [0, 1]^{C \times H_g \times W_g}$ for C classes on the ground plane ($H_g \times W_g$), scaled by a factor R to the metric ground plane. For pedestrian datasets, $C = 1$, while vehicle datasets may differentiate between multiple vehicle types. A prediction $\hat{Y}_{x,y,c} = 1$ indicates a detected object, whereas $\hat{Y}_{x,y,c} = 0$ denotes background.

To train the keypoint detection, we construct a target heatmap $Y \in [0, 1]^{C \times H_g \times W_g}$. For each ground truth location $p \in \mathbb{R}^2$, we compute its low-resolution equivalent $\tilde{p} = \lfloor \frac{p}{R} \rfloor$. Subsequently, we generate a Gaussian kernel for each object:

$$Y_{xyc} = \exp \left(-\frac{(x - \tilde{p}_x)^2 + (y - \tilde{p}_y)^2}{2\sigma_p^2} \right), \quad (3.1)$$

where σ_p represents the size-adaptive standard deviation [84]. The detection head is optimized using pixel-wise logistic regression with Focal Loss [99]:

$$\mathcal{L}_{center} = \frac{-1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^\beta (\hat{Y}_{xyc})^\alpha \log(1 - \hat{Y}_{xyc}) & \text{otherwise} \end{cases}, \quad (3.2)$$

where α and β are hyper-parameters of the focal loss [84], and N denotes the number of objects in the scene. Following CornerNet [84], we employ $\alpha = 2$ and $\beta = 4$.

To enhance precision, the detection head also predicts the local offset $\hat{O} \in \mathbb{R}^{2 \times H_g \times W_g}$ of each object caused by discretization:

$$\mathcal{L}_{off} = \frac{1}{N} \sum_p \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right|. \quad (3.3)$$

Additionally, we incorporate a detection head for the image features to predict head and foot locations, guiding the features prior to voxel space projection. This image detection head is trained using the focal loss L_{img} , analogous to the BEV detection head.

For tracking, we develop two distinct methods for detection association: one based on appearance and another based on motion, as detailed below.

Appearance-based Association (EarlyBird) In EarlyBird [10[†]], we predict a re-ID feature for each detection in the BEV space. The re-ID head generates a feature embedding of dimension 128 for each detection, which is subsequently utilized for association between detections in the current and previous frames. We optimize the re-ID head using a contrastive loss $\mathcal{L}_{contras}$ for the feature embeddings and a cross-entropy loss $\mathcal{L}_{ce}$ over the person IDs. The composite tracking loss is formulated as:

$$\mathcal{L}_{tracking} = \mathcal{L}_{contras} + \mathcal{L}_{ce}. \quad (3.4)$$

Motion-based Association (TrackTacular) For TrackTacular [11[†]], we predict the motion offset of each detection in the BEV space. Analogous to the center offset, we estimate the offset $\hat{O}_{t-1} \in \mathbb{R}^{2 \times H_g \times W_g}$ to the location p_{t-1} in the previous frame. The decoder, having access to both current and historical frames, can establish associations between them. To accommodate varying magnitudes of these offsets, we employ the Smooth L1 Loss:

$$L_{tracking} = \begin{cases} \frac{1}{2}((p - p_{t-1}) - \hat{O}_{t-1})^2 & \text{if } |((p - p_{t-1}) - \hat{O}_{t-1})| < \delta \\ \delta((p - p_{t-1}) - \hat{O}_{t-1}) - \frac{1}{2}\delta & \text{otherwise} \end{cases}, \quad (3.5)$$

where $\delta = 1$.

The comprehensive loss objective for our approach is thus formulated as:

$$\mathcal{L} = \mathcal{L}_{center} + \mathcal{L}_{tracking} + \mathcal{L}_{off} + \mathcal{L}_{img}. \quad (3.6)$$

3.3. Lifting to the Bird's Eye View

We present a comprehensive overview of our architecture in Figure 3.4. The process initiates with S RGB input images ($S \times 3 \times H_i \times W_i$) that undergo augmentation before being processed by the encoder network, yielding downsampled image features ($S \times C_f \times \frac{H_i}{4} \times \frac{W_i}{4}$). Subsequently, these features are projected into a common BEV space ($S \times C_f \times H_g \times W_g$) utilizing various projection methodologies. The BEV space is then reduced along the vertical dimension ($S \times H_g \times W_g$). In the ensuing step, the feature from the previous time step is concatenated with the current BEV feature ($2 \times S \times C_f \times H_g \times W_g$). Finally, the BEV features are processed through a decoder network.

3.3.1. Perspective Transformation

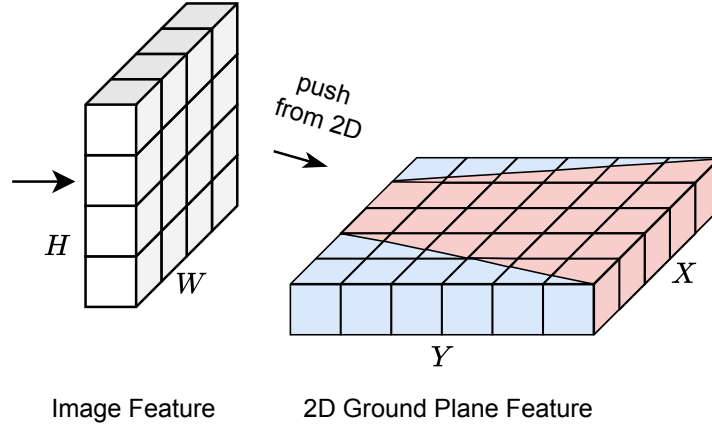


Figure 3.6.: **Perspective Transformation.** The perspective transformation *pushes* the 2D image features to a 2D ground plane feature.

The perspective transformation represents the most elementary lifting method, as it does not incorporate height information. Following the approach of [70], we employ perspective projection to map the image features onto the ground plane. The pinhole camera model [58] utilizes a 3×4 transformation matrix $\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}]$ that maps 3D locations (x, y, z) to 2D image pixel coordinates (u, v) . By projecting all pixels to the ground plane at $z = 0$, we simplify the projection to:

$$d \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{P}_0 \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (3.7)$$

where d denotes a real-valued scaling factor and $\mathbf{P}_0$ represents the 3×3 perspective transformation matrix, derived from $\mathbf{P}$ by omitting the third column. Features from all S cameras are projected onto the ground plane using this equation, with each camera employing its unique projection matrix $\mathbf{P}_0^{(s)}$.

This perspective transformation effectively projects the 2D image features onto a 2D ground plane feature representation, as depicted in Figure 3.6. While this method exhibits computational efficiency and requires no additional parameters, it is subject to certain limitations. Primarily, it operates under the assumption that all objects of interest are situated on a planar surface, which may not accurately reflect real-world scenarios. Furthermore, larger objects tend to cast a "shadow" on the ground plane, wherein the head is projected at a greater distance from the camera compared to the feet. This phenomenon can potentially compromise the localization accuracy of the objects [69].

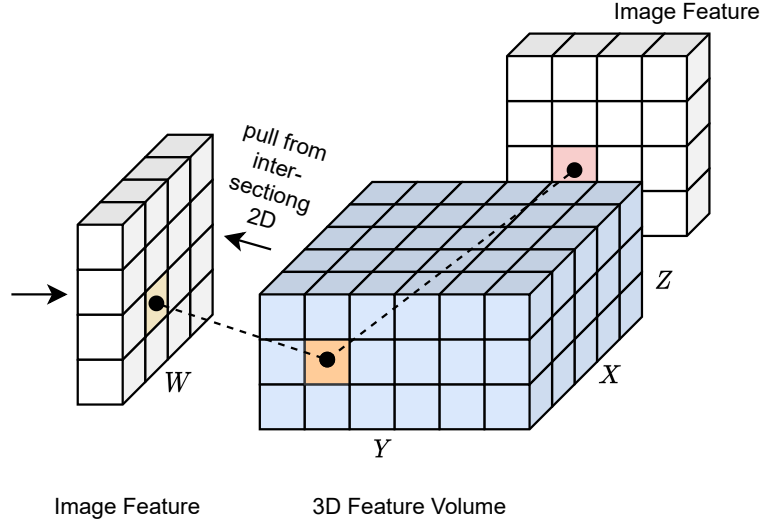


Figure 3.7.: **Bilinear Sampling.** The bilinear sampling method introduced in Simple-BEV [56] each 3D Voxel *pulls* feature from the 2D map by projection and sampling.

3.3.2. Bilinear Sampling

The methodology proposed in Simple-BEV [56] simplifies depth projection without explicit depth prediction. In contrast to traditional approaches where each ray propagates information to intersecting voxels, this method inverts the process: 3D voxels *extract* information from the 2D image. This extraction is accomplished through a series of steps: projecting the 3D voxel onto the image plane, determining the projection's validity within the image bounds, and subsequently sampling the image features with sub-pixel accuracy for the voxel. For a 3D point $(x, y, z)^T$ in the BEV grid, its projection onto the image plane (u, v) is given by:

$$K [R|t] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = P^{(s)} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = d \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad (3.8)$$

where K denotes the camera intrinsic matrix, and $[R|t]$ represents the extrinsic parameters (rotation and translation). The eight vertices $(x_c, y_c, z_c)^T$ of the voxel are projected onto the image plane, and the sampling area is defined by $[\min(u_c), \min(v_c), \max(u_c), \max(v_c)]$. For each voxel, features from all visible images are sampled using bilinear interpolation and aggregated across all S cameras.

This approach offers several advantages. Primarily, it ensures feature assignment to every voxel, whereas splatting methods may leave voxels unfilled, particularly those distant from the cameras. This property enhances the method's robustness in long-range perception scenarios. Furthermore, the computational efficiency is notable, as it circumvents the need for explicit depth estimation. However, in regions lacking overlap, the method may be susceptible to depth ambiguity. Although the bilinear

sampling method [56, 57] was not originally conceived for multi-camera perception, it exhibits a unique characteristic in overlapping regions: it approximates a triangulation of image points at voxel granularity. This triangulation effect is illustrated in Figure 3.7. It is worth noting that for datasets with complete overlap, i.e., where every voxel is observed by at least two cameras, the bilinear sampling method becomes equivalent to the triangulation method.

3.3.3. Depth Splatting

The "Lift, Splat, Shoot" architecture [118] introduces a novel approach to generate BEV features from monocular images. Its lifting method, often referred to as "depth-based feature unprojection", simulates a point cloud from a camera image. For each pixel (u, v) in the image, the network predicts a depth distribution over D discrete depth bins. We set near and far planes d_{near} and d_{far} . Between these, we discretize depth into Z bins, represented by a vector $\mathbf{b} \in \mathbb{R}^Z$, where its z -th element b_z is defined in disparity space as:

$$b_z = \left(1 - \frac{z}{Z}\right) \left(\frac{1}{d_{\text{near}}} - \frac{1}{d_{\text{far}}}\right) + \frac{1}{d_{\text{far}}}. \quad (3.9)$$

We then define a discrete probability distribution $p_\phi(z)$ over an index variable z , parameterized by a vector of discrete probabilities ϕ , where its z -th element ϕ_z is the probability that the feature of the pixel exists in depth bucket b_z . The probability distribution is estimated as part of the image features from the encoder network.

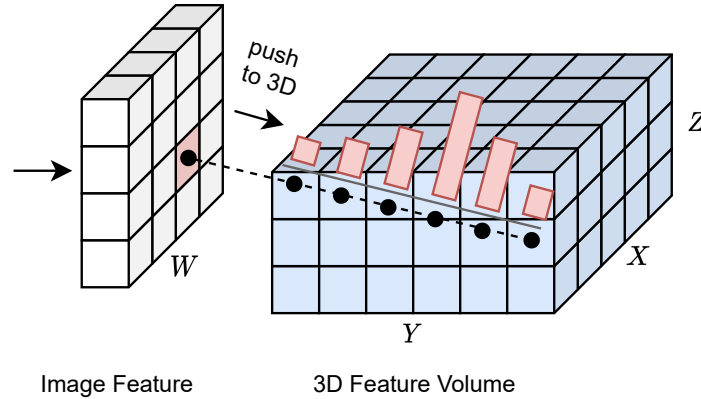


Figure 3.8.: **Depth Splatting.** In the depth splatting approach [118] the 2D feature are *pushed* to 3D with depth prediction, filling voxels that intersect with its ray.

The unprojection of image feature channels, utilizing Equation (3.8) in conjunction with the predicted depth, generates a point cloud of dimensions $(D \times C_f \times \frac{H_i}{4} \times \frac{W_i}{4})$ within each camera frustum. For every pixel (u, v) , the corresponding 3D world coordinates (x, y, z) are computed using the extrinsic calibration matrix $[R|t]$. Subsequently, these point clouds are consolidated into a unified voxel space, with the channel

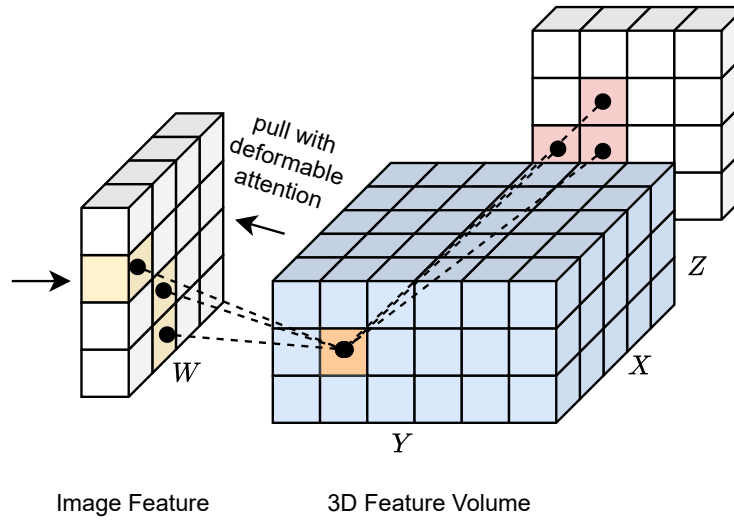


Figure 3.9.: **BEVFormer Lifting.** The lifting method introduced by BEVFormer [92] uses deformable attention to aggregate multiple image features with a learnable offset.

information weighted according to the probability distribution of the discrete depth (*cf.* Figure 3.8).

This methodology is characterized as a 2D-to-3D *push* operation. Its primary advantage lies in its capacity for explicit depth reasoning and 3D geometry learning. The utilization of a depth distribution enables the model to effectively capture and represent the uncertainty inherent in depth predictions. However, this approach is computationally intensive, necessitating depth predictions for each pixel that contributes to the camera frustum. Furthermore, the discretization of depth information may introduce quantization errors, potentially impacting the overall accuracy of the 3D representation.

3.3.4. Deformable Attention

BEVFormer [92] represents a paradigm shift in BEV feature generation by leveraging the power of transformer architectures. The core idea is to use a transformer to implicitly reason about 3D space in BEV, while aggregating information from multiple 2D images. It treats BEV features as learnable queries that interact with multi-view image features through a cross-attention mechanism.

The process begins by projecting each BEV query voxel onto the image planes of all available cameras with Equation (3.8). For each projected point, the method samples a set of 2D reference points and their corresponding image features. These sampled features are then aggregated using a deformable attention mechanism (*cf.* Section 2.5.2), which allows the model to focus on the most relevant image regions for each BEV query.

In the context of BEVFormer, the query, keys, and values are defined as follows:

- **query** q : Represents the BEV features at location (x, y, z) that we want to update or refine.
- **keys** k : The potential 2D image locations (u, v) that backproject to the BEV query location (*cf.* Equation (3.8)).
- **values** v : The encoded image features represent the information that will be aggregated to update the query.

This formulation allows the model to determine how much each sampled image feature (represented by keys and values) contributes to updating each BEV query point. The attention mechanism enables the model to focus on the most relevant parts of the input images for each BEV location.

One of the key innovations in BEVFormer is the use of learnable offsets for the sampling points. These offsets allow the model to adaptively adjust the sampling locations based on the content of the image and the query, enabling more flexible and accurate feature aggregation. The offsets are predicted by a small neural network and are added to the initial reference points before sampling the image features.

The multi-head attention mechanism allows the model to capture different aspects of the relationship between BEV queries and image features. Each attention head can focus on different parts of the input, potentially capturing different scales or types of information.

BEVFormer also introduces a temporal dimension to the BEV representation, allowing it to leverage motion and temporal consistency. This is achieved through a temporal self-attention mechanism that connects BEV features across consecutive frames. However, in our implementation, we replace this temporal self-attention with a more resource-efficient temporal aggregation method, which will be introduced in the next section. This temporal aggregation is applied consistently across all lifting methods to ensure fair comparison.

The approach of BEVFormer is similar to bilinear sampling in that it projects 3D points onto 2D image planes. However, it goes beyond simple interpolation by using attention to aggregate features from a neighborhood around each projected point. This allows the model to consider a wider context and potentially handle occlusions and ambiguities more effectively.

By leveraging the power of transformers and especially attention, BEVFormer achieves state-of-the-art performance on various 3D perception tasks, demonstrating the effectiveness of this approach in fusing multi-view information into a coherent BEV representation. However, the method requires a large number of learnable parameters and is computationally expensive.

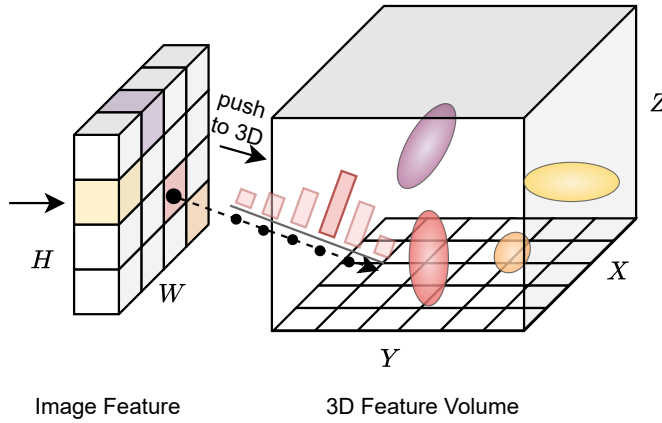


Figure 3.10.: **3D Gaussian Splatting.** The 3D Gaussian Splatting method.

3.4. Lifting with Gaussian Splatting

Additionally to the presented lifting methods, we propose a novel lifting method based on 3D Gaussian Splatting. The motivation behind this new contribution is that the Depth Splatting method (*cf.* Section 3.3.3) each (feature) pixel can only splat a single line. With the Gaussian Splatting, we can model a 3D ellipsoid with different sizes in different directions, representing objects of various size. The size and orientation of the ellipsoid can be perceived as a measure of uncertainty in the depth estimation. This lifting method can be seen as a point cloud lifting method, where each point is represented by a Gaussian. Combined with the rasterization step, this method can represent objects very sparse and thus computationally efficient.

This method also transforms the discrete image features into a continuous BEV space, allowing for more accurate representation of objects and surfaces.

Parameter Estimation For each feature pixel, we predict a 3D Gaussian with parameters:

- mean μ , interpretable as a 3D location x, y, z
- opacity α
- covariance Σ
- parameters: RGB value, Spherical Harmonics (SH), features, etc.

The covariance matrix Σ is learned for each pixel and represents the local geometry of the Gaussian. We learn this as a scale and rotation matrix for each pixel. As parameters, we use the feature of the pixel.

The mean μ is obtained by predicting a depth for each pixel and use the camera intrinsics to transform it to 3D. Similar to the Depth Splatting method, we predict a depth distribution over D discrete depth bins. As we now lift to a continuous space, also learn a per bucket offset that adjusts the depth of the pixel-aligned Gaussian

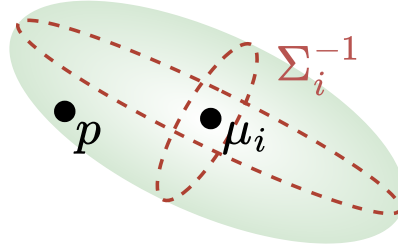


Figure 3.11.: **3D Gaussian.** The influence of a 3D Gaussian i on a point p in 3D.

between bucket boundaries. The location of the Gaussian is then sampled from the depth distribution $z \sim p_\phi(z)$.

To train our model, we need to backpropagate gradients through the non-differentiable sampling operation $z \sim p_\phi(z)$. Inspired by variational autoencoders, we employ a reparameterization trick. We set the Gaussian opacity α equal to the probability of the sampled depth bucket: $\alpha = \phi_z$. In the backward pass, assign gradients of the loss $\mathcal{L}$ w.r.t. opacities α to the depth probability buckets ϕ , *i.e.* $\nabla_\phi \mathcal{L} = \nabla_\alpha \mathcal{L}$.

This approach effectively increases opacity for correct depth predictions, concentrating probability mass in the right bucket. And decreases opacity for incorrect depths, reducing the likelihood of future misclassifications. Thus, we can train the model to learn accurate depth distributions without direct differentiation through the sampling process.

Rasterization As the Gaussians are learned in a continuous space, we need to rasterize them to a discrete grid. Rasterization for 3D Gaussians involves projecting each Gaussian onto the 2D image plane and determining its contribution to each pixel. The process is based on the probability density function of a multivariate Gaussian distribution:

$$f_i(p) = \alpha_i \exp \left(-\frac{1}{2} (p - \mu_i)^\top \Sigma_i^{-1} (p - \mu_i) \right) \quad (3.10)$$

where $f_i(p)$ is the contribution of the i -th Gaussian at point p , α_i is the opacity, μ_i is the mean (center) of the Gaussian, and Σ_i is its covariance matrix. A highly optimized algorithm for this rasterization process is proposed by [80], which efficiently computes these contributions for all relevant pixels, taking into account occlusions and blending between multiple Gaussians.

3.5. Experiments

In this section, we present the datasets utilized for evaluation and the metrics employed to assess performance. Subsequently, we report the results of our approach

	Target	Domain	Scenes	#Cams/Scene	Area	Annotations	Density	Duration
Wildtrack (2018) [22]	Pedestrian	Real	1	7	12×36 m	BEV Grid	23.8	3min
MultiviewX (2020) [70]	Pedestrian	Synthetic	1	8	16×25 m	BEV Grid	40	3min
Synthehicle (2023) [3 [†]]	Vehicle	Synthetic	8	3–8	75×75 m	3D Boxes	7.5	17h

Table 3.1.: **Multi-View Tracking Datasets.** Overview of the tracking datasets used in the evaluation.

on these datasets and compare them with state-of-the-art methods. Furthermore, we conduct additional ablation studies to evaluate the individual components of our approach.

3.5.1. Datasets

Multi-view tracking datasets present significant challenges in terms of recording and labeling compared to their single-view counterparts. The absence of depth information from cameras constrains these datasets to the 2D ground plane. Consequently, most real-world datasets [46, 47, 162, 22] are limited in both scale and quality. To address these limitations, synthetic datasets [70, 3[†]] have been developed. We evaluate our approach on three datasets: Wildtrack [22], MultiviewX [70], and Synthehicle [3[†]]. A summary of these datasets is presented in Table 4.2. As Synthehicle [3[†]] resulted from one of our publications, we provide a more detailed exposition of this dataset.

Wildtrack [22] is a prominent dataset for multi-view pedestrian tracking. The dataset was acquired in a high-traffic outdoor area, specifically a plaza in front of Zürich’s main train station (*cf.* Figure 3.1). This location was selected for its high pedestrian density and diverse movement patterns, providing a challenging environment for multi-view pedestrian detection and tracking algorithms. Seven cameras were strategically positioned around the area to ensure maximum coverage and capture pedestrians from various angles and distances. One of the primary challenges presented by the Wildtrack dataset is the handling of occlusions. Due to the crowded nature of the scene, pedestrians frequently occlude one another, testing the robustness of detection and tracking algorithms. The low frame rate of 2 fps was chosen to simulate realistic surveillance scenarios where high frame rates are often infeasible due to storage constraints. This low frame rate presents an additional challenge for tracking algorithms, as they must handle larger displacements between consecutive frames. The seven cameras capture an overlapping field-of-view of a 12×36 meter public area where pedestrian movement is unscripted. Ground plane annotations are provided on a 480×1440 grid, equating to 2.5 cm grid cells. On average, each frame contains 20 pedestrians covered by 3.74 cameras. The video is recorded at a resolution of 1080×1920 pixels at a frame rate of 2 fps.

Wildtrack has become a benchmark dataset in the field of multi-view pedestrian detection and tracking, enabling researchers to evaluate and compare their algorithms

in a realistic and challenging setting. Its real-world nature, combined with precise annotations and calibration information, makes it an invaluable resource for advancing the state-of-the-art in multi-view computer vision tasks.

MultiviewX [70] is a synthetic dataset designed to mirror the specifications of the Wildtrack dataset, utilizing a game engine for its creation. This dataset offers views from 6 virtual cameras, strategically positioned to provide an overlapping field-of-view covering a slightly more compact area of 16×25 meters.

The annotations in MultiviewX are provided on a ground plane grid of 640×1000 , maintaining the same 2.5 cm square resolution as Wildtrack. On average, each frame contains 40 pedestrians, which is twice the number found in Wildtrack, presenting a more challenging scenario for detection and tracking algorithms. The increased pedestrian density tests the robustness of multi-object tracking systems in crowded scenes. The camera resolution is maintained at 1080×1920 pixels, and the frame rate remains at 2 fps, matching the Wildtrack dataset. This consistency in technical specifications facilitates direct comparisons between algorithms tested on both datasets. The dataset spans 400 frames, equivalent to Wildtrack's duration. However, the synthetic nature of MultiviewX offers several advantages as it has perfect camera calibration and synchronization, eliminating errors that can occur in real-world setups. Additionally, the annotations are free from any human labeling errors.

Despite being synthetic, MultiviewX strives to maintain realism in pedestrian movements, appearances, and interactions. This balance between control and realism makes MultiviewX an invaluable resource for developing and benchmarking multi-view pedestrian detection and tracking algorithms, complementing real-world datasets like Wildtrack.

Synthehicle [3[†]] represents a significant advancement in synthetic datasets for vehicle detection and tracking algorithms in computer vision. In response to the limitations of existing datasets [147] in terms of scale, diversity, and annotation quality, we introduced Synthehicle, offering an extensive and realistic dataset comprising over 1.2 million vehicle instances across 60,000 images.

The dataset's creation leveraged the CARLA 3D driving simulator [35], utilizing its state-of-the-art rendering engine to generate a diverse array of vehicle models, poses, and environmental conditions. This methodology enables the incorporation of complex scenarios, including occlusions, high-density scenes, and varied lighting and weather conditions, which are typically underrepresented in real-world datasets. Synthehicle provides comprehensive annotations, including both bounding boxes and segmentation masks for each vehicle instance.

The dataset's architecture encompasses 8 virtual town maps, each featuring both overlapping and non-overlapping camera configurations. Our focus in this work is on the overlapping scenes, specifically intersections captured by 3-8 cameras. Synthehicle models a range of environmental conditions, including diurnal variations (day, dawn, night) and weather phenomena (rain). Each scene is accompanied by

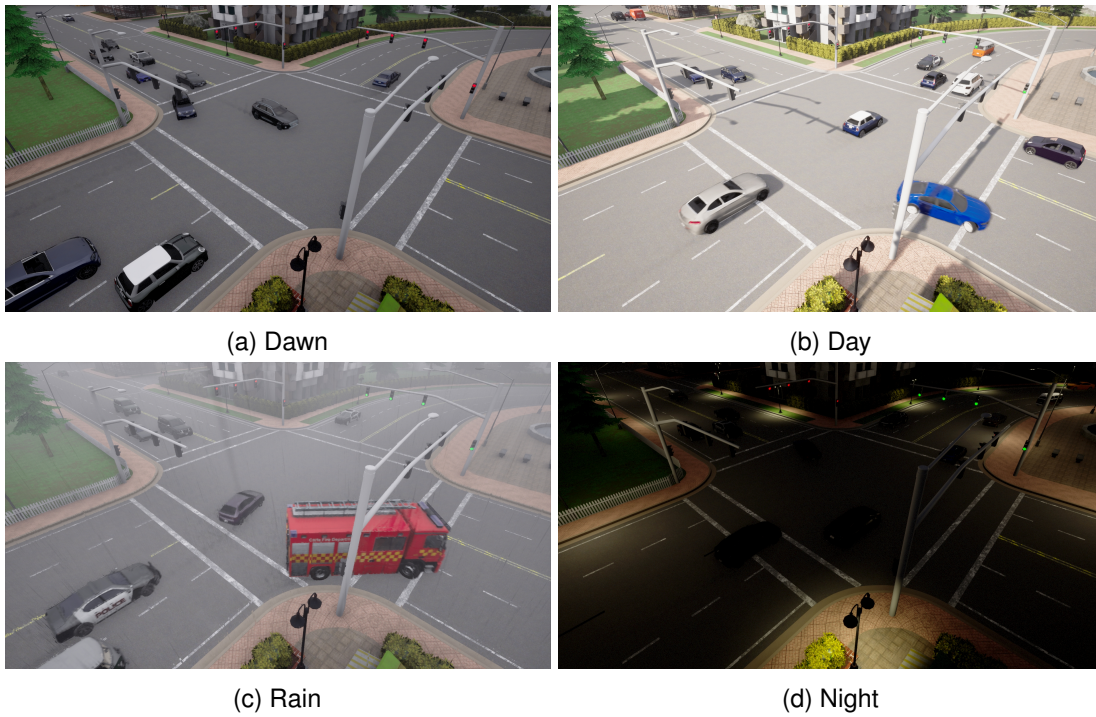


Figure 3.12.: **Synthicle Ambiances.** Synthehicle provides challenging scenes with different weather and lighting conditions that are underrepresented in real-world datasets.

precise camera calibration information for all cameras. The dataset categorizes vehicles into three classes: cars, trucks, and motorbikes.

A key innovation of Synthehicle is its capacity for testing generalization. The dataset is partitioned into distinct towns for training and testing sets, enabling researchers to evaluate their algorithms on previously unseen intersections. This feature is instrumental in assessing the robustness and adaptability of detection and tracking methodologies in novel environments.

The dataset is captured at a high temporal resolution of 30 fps with a spatial resolution of 1920x1080 pixels, providing high-fidelity representations of vehicle movements. This enhanced temporal resolution facilitates more precise tracking and motion prediction, particularly in complex traffic scenarios.

In conclusion, Synthehicle represents a paradigm shift in vehicle detection and tracking datasets. Its unprecedented scale, diversity, and quality introduce a new dimension to the multi-view tracking community, especially in terms of generalization to unseen environments. The comprehensive nature of the dataset renders it an invaluable resource for researchers and practitioners engaged in various aspects of computer vision in traffic scenarios, spanning from single-frame detection to sophisticated multi-object tracking across multiple viewpoints.

3.5.2. Evaluation Metrics

The evaluation of 3D detection encompasses various methodologies, with three predominant protocols: 3D bounding boxes [158, 49], 2D BEV bounding boxes [49], and 2D BEV center points [16]. In this study, we adopt the 2D BEV center point protocol, which is widely employed in MTMC tracking and offers greater tolerance for minor discrepancies in 3D detection.

Detection We quantify false positives (FP), false negatives (FN), and true positives (TP) by employing Hungarian matching to associate detections with ground truth. A pedestrian is classified as a true positive if its detected position falls within a radius $r = 0.5 \text{ m}$ of the ground truth, approximating the typical human body radius. Consistent with previous research [22, 70], we utilize Multiple Object Detection Accuracy (MODA) [79] as the primary performance indicator, as it accounts for normalized missed detections and false positives:

$$\text{MODA} = 1 - \frac{\text{FN} + \text{FP}}{\text{GT}}, \quad (3.11)$$

where GT represents the total number of ground truth objects. The Multiple Object Detection Precision (MODP) metric quantifies the spatial discrepancy between predicted and ground truth locations:

$$\text{MODP} = \frac{1}{\text{GT}} \sum_{i=1}^{\text{GT}} \|GT_i - D_i\|_2, \quad (3.12)$$

with D_i denoting the predicted location matched to the ground truth location GT_i .

Additionally, we report Precision and Recall, calculated as $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$, respectively.

Tracking In alignment with our detection methodology, we evaluate all tracking metrics using 2D BEV center points [16, 22]. We report standard MOT metrics [9, 158] and identity-aware metrics [126]. The threshold for a positive assignment is established at $r = 1 \text{ m}$. The primary metrics of focus are Multiple Object Tracking Accuracy (MOTA) and IDF1. MOTA incorporates missed detections, false detections, and identity switches, while IDF1 measures missed detections, false positives, and identity changes. A comprehensive overview of the multi-object tracking metrics is provided in Chapter A.

3.5.3. Implementation Details

Following the methodology of [56, 69], we apply random resizing and cropping to the RGB input within a scale range of $[0.8, 1.2]$, adjusting the intrinsic camera matrix K accordingly. Furthermore, we introduce noise to the extrinsic camera translation vector t to mitigate overfitting in the decoder. The detector is trained using an Adam

	Lifting Method	Wildtrack				MultiviewX			
		MODA	MODP	Precision	Recall	MODA	MODP	Precision	Recall
RCNN & Cluster [162]	Clustering	11.3	18.4	68	43	18.7	46.4	63.5	43.9
POM-CNN [47]	Clustering	23.2	30.5	75	55	-	-	-	-
DeepMCD [23]	Clustering	67.8	64.2	85	82	70.0	73.0	85.7	83.3
Deep-Occlusion [6]	Clustering	74.1	53.8	95	80	75.2	54.7	97.8	80.2
MVDet [70]	Persp. Proj.	88.2	75.7	94.7	93.6	83.9	79.6	96.8	86.7
SHOT [140]	Persp. Proj.	90.2	76.5	96.1	94.0	88.3	82.0	96.6	91.5
3DROM [123]	Persp. Proj.	91.2	76.9	95.9	95.3	90.0	83.7	97.5	92.4
MVDeTr [69]	Persp. Proj.	91.5	82.1	97.4	94.0	93.7	91.3	99.5	94.2
MVTT [87]	Persp.+RoI	94.1	81.3	97.6	96.5	95.0	92.8	99.4	95.6
EarlyBird	Persp. Proj.	91.2	81.8	94.9	96.3	94.2	90.1	98.6	95.7
TrackTacular	Persp. Proj.	91.8	79.8	96.2	95.6	95.9	89.2	99.5	96.4
	Bilin. Sampl.	92.1	76.2	97.0	95.1	96.5	75.0	99.4	97.1
	Depth Splat.	93.2	77.5	97.3	95.8	96.1	90.4	99.0	97.1
	Deform. Attn.	78.4	73.1	93.8	84.0	94.4	73.1	98.6	95.8
	Gauss. Splat.	89.1	80.2	95.9	93.1	90.2	87.8	97.1	93.0

Table 3.2.: **Pedestrian Detection.** Comparison of the BEV detection performance with the state-of-the-art methods on the Wildtrack and MultiviewX datasets.

optimizer with a one-cycle learning rate scheduler, employing a maximum learning rate of 10^{-3} . Depending on the encoder size, we utilize a batch size of 1 – 2. To enhance training stability, we accumulate gradients across multiple batches before weight updates, effectively achieving an adequate batch size of 16. Both the encoder and decoder networks are initialized with weights pre-trained on *ImageNet-1K*. All experiments are conducted on an RTX 3090 GPU.

Temporal Caching Our method incorporates the BEV feature of the preceding frame. A significant challenge lies in caching the previous feature while maintaining high variation in batch samples, as gradient updates occur sequentially. While testing allows for a batch size of one to ensure the previous feature is always computed, this approach would be detrimental to training performance due to the small batch size or limited sample variation. To address this, we have developed a custom sampler that constructs batches in a semi-sequential order, taking into account the accumulated batch size.

3.5.4. Comparison to State-of-the-Art

Pedestrian We present the detection results for both pedestrian datasets in Table 3.2. Our approach demonstrates significant improvements over previous state-of-the-art methods. Only the two-stage approach MVTT [87] exhibits superior MODA on Wildtrack. Our method outperforms all others across the remaining metrics. The high overall values suggest an approaching saturation point in results. Notably, the improvements for MultiviewX are more substantial compared to Wildtrack. This discrepancy may indicate that the accuracy of Wildtrack’s labeling, which utilizes

		Wildtrack									
		IDF1 $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$					
	KSP-DO [22]	73.2	69.6	61.5	28.7	25.1					
	KSP-DO-ptack [22]	78.4	72.2	60.3	42.1	14.6					
	GLMB-YOLOv3 [112]	74.3	69.7	73.2	79.5	21.6					
	GLMB-DO [112]	72.5	70.1	63.1	93.6	22.8					
	DMCT [168]	77.8	72.8	79.1	61.0	4.9					
	DMCT Stack [168]	81.9	74.6	78.9	65.9	4.9					
	ReST [27]	86.7	84.9	84.1	87.8	4.9					
	MVFlow [36]	93.5	91.3	—	—	—					
							MultiviewX				
		IDF1 $\uparrow$	MOTA $\uparrow$	MOTP $\uparrow$	MT $\uparrow$	ML $\downarrow$					
EarlyBird	Persp. Proj.	92.3	89.5	86.6	78.0	4.9	82.4	88.4	86.2	82.9	1.3
TrackTacular	Persp. Proj.	94.2	89.6	81.7	87.8	4.9	84.2	91.4	86.7	85.5	2.6
TrackTacular	Bilin. Sampl.	95.3	91.8	85.4	87.8	4.9	85.6	92.4	80.1	92.1	2.6
TrackTacular	Depth Splat.	93.6	90.2	84.2	82.9	4.9	83.4	91.8	84.7	90.8	2.6
TrackTacular	Deform. Att.	88.0	82.2	78.9	75.6	4.9	84.8	91.4	80.6	89.5	2.6
TrackTacular	Gauss. Splat.	86.4	85.2	79.5	78.0	7.3	76.7	83.9	83.0	81.6	6.6

Table 3.3.: **Pedestrian Tracking.** Evaluation of tracking results on Wildtrack and MultiviewX.

perspective transform annotation [22], is potentially limiting further advancements and underscores the necessity for more challenging datasets.

Recent related work predominantly employed perspective projection as the lifting method. Our approach surpasses these methods due to the incorporation of additional temporal information. Generally, the parameter-free lifting methods, particularly bilinear sampling, demonstrate competitive results compared to parameterized depth splatting. This observation aligns with the expectation that the depth-splatting method can provide dense features throughout the observed space for this confined area. The deformable attention method, despite its robustness in autonomous driving tasks [56], underperformed in our task. We observed significant overfitting during training, suggesting its unsuitability for smaller datasets such as those employed in this study.

The tracking results are presented in Table 3.3. Our approach advances the state-of-the-art on both datasets, demonstrating enhanced tracking quality compared to the three most recent methodologies. These results indicate that tracking in BEV is currently the most effective approach, as evidenced by, MVFlow [36], ReST [27], and our work [10[†], 11[†]]. Our first work, EarlyBird [10[†]], that utilizes appearance-based association and single-frame information outperforms the other approaches by a significant margin. Our later approach, TrackTacular [11[†]], shows the significant improvement in the IDF1 score, highlighting the advantages of our early-fusion tracker, which successfully combines appearance and motion cues. Consistent with the detection task, we observe similar trends of diminishing improvements on Wildtrack.

Vehicle Results Table 3.4 presents our results on Synthechicle [3[†]], utilizing the two best-performing lifting methods from the pedestrian datasets. Synthechicle’s scenes, being substantially larger than the pedestrian scenes, enable evaluation

Synthehicle	Lifting Method	Scene Specific			Cross Scene		
		IDF1↑	MOTA↑	MOTP↑	IDF1↑	MOTA↑	MOTP↑
TrackTacular	Bilinear Sampling	48.0	10.6	32.8	18.3	-22.0	29.3
	Depth Splatting	57.2	33.1	41.9	24.2	-1.5	32.4

Table 3.4.: **Synthehicle**. Evaluation on the scene specific validation set and the cross scene test set. The validation set consists of temporally separated scenes from the train set and the test set contains only unseen scenes.

of performance over extended distances. The results demonstrate the superiority of the depth splatting lifting method over bilinear sampling, indicating the latter’s inability to provide dense features for larger scenes. These results are notably less saturated compared to the pedestrian datasets, offering the research community a new benchmark for future investigations.

An additional advantage of evaluating on Synthehicle is the ability to assess performance on both familiar scenes from training (scene-specific) and novel scenes (cross-scene). In practical applications, training a model for each specific scene is undesirable, necessitating methods capable of cross-scene generalization. In this context, the depth splatting lifting method again outperforms bilinear sampling. However, the results for cross-scene evaluation remain suboptimal, emphasizing persistent challenges in scene generalization. The low MOTA score, a metric heavily weighted towards detection, suggests that missed detections constitute a significant issue in this scenario.

3.5.5. Ablation Studies

Lifting Methods While the ablation studies have been discussed for each target task, this section presents general observations. The lifting methods can be categorized into two groups: parameter-free and parameterized methods. Parameter-free methods exhibit ease of training and demonstrate robust performance on pedestrian datasets. Conversely, parameterized methods offer greater flexibility and superior generalization to larger scenes, albeit with increased training complexity and less stable results. Three methods have emerged as particularly effective: perspective projection, bilinear sampling, and depth splatting. Notably, the simplest method, perspective projection, yields strong results. However, its extension, bilinear sampling, outperforms it by modeling the scene in 3D rather than 2D.

The deformable attention method, despite its robustness in autonomous driving tasks [56], underperformed in our context. We observed significant overfitting during training, suggesting its unsuitability for smaller datasets such as those employed in this study. The method’s requirement for substantial training data to converge, despite its efficiency as an attention mechanism, poses challenges in our experimental setup.

EarlyBird (Persp. Trans.)	Detection		Tracking		
	MODA	MODP	IDF1	MOTA	MOTP
ResNet-18	91.2	81.8	92.3	89.5	86.6
ResNet-50	89.6	82.3	92.6	88.8	86.5
Swin-T	89.5	81.3	92.0	87.3	87.9

Table 3.5.: **Encoder Ablation.** Ablation of different encoders on detection and tracking results of the Wildtrack dataset.

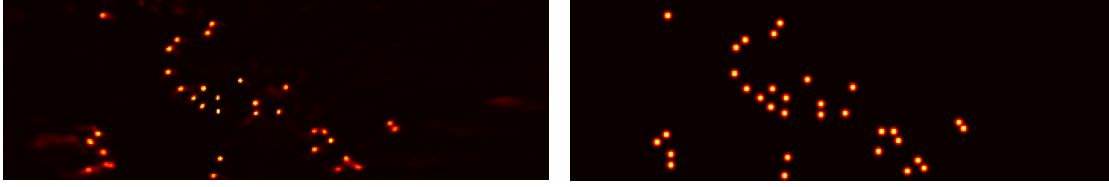
	Detection		Tracking	
	MODA	MODP	IDF1	MOTA
Baseline (Bilin. Sampl.)	95.4	89.8	81.5	90.0
+ History Fusion	96.5	75.0	83.8	87.9
+ Motion Prediction	"	"	85.6	92.4

Table 3.6.: **Temporal Ablation.** Evaluation of temporal aggregation components introduced by our approach compared to the baseline on the MultiviewX dataset.

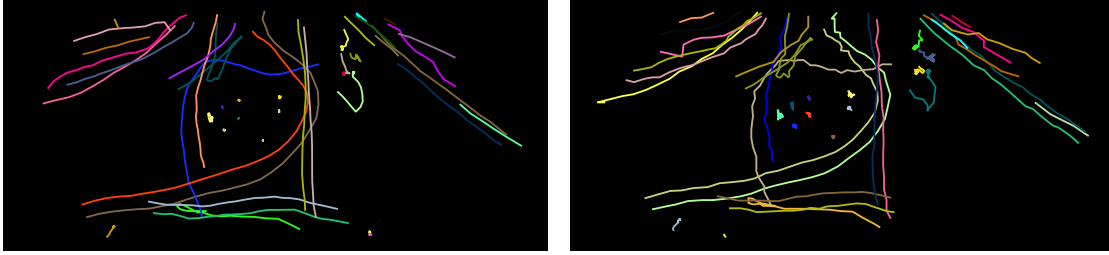
Our proposed method, Gaussian splatting, offers notable computational efficiency (*cf.* Section 3.5.7). While the results approach state-of-the-art performance, there remains room for improvement. In contrast to dense voxelized methods, our approach utilizes sparse Gaussians in continuous 3D space. This sparse representation appears to be less expressive than dense voxelized representations.

Motion vs. Appearance Association The primary distinction between our two approaches lies in the association method. EarlyBird [10[†]] employs appearance-based association, while TrackTacular [11[†]] utilizes motion-based association. Table 3.3 allows for comparison of both approaches using the same perspective projection lifting method. The IDF1 score, which measures identity preservation, is particularly relevant for this comparison. Our results indicate that motion-based association outperforms appearance-based association. The potential advantage of re-ID features in re-associating lost tracks appears to be less significant in the multi-view setting, possibly due to the robust detection results.

Image Encoder The encoder extracts features from the RGB image that are subsequently projected onto the ground plane. While similar detection-based approaches [70, 69, 87, 123] predominantly utilize a ResNet-18 encoder, these methods do not necessitate encoding of identity features. Consequently, we evaluated our approach with larger encoders and transformer-based encoders, as shown in Table 3.5. The ablation study reveals that ResNet-18 exhibits best performance. Although ResNet-50 may marginally outperform in detection and tracking metrics, the more compact ResNet-18 surpasses it in the primary metrics MODA and MOTA, while maintaining competitive IDF1 scores. Thus, we employ ResNet-18 for reporting all other results.



(a) Comparison of the detection result on Wildtrack as a heat map. Each point in the ground truth represents a pedestrian in the BEV.



(b) Comparisons of the tracking results on Wildtrack. Each colored line represents a path taken by one tracked pedestrian as seen from the BEV.

Figure 3.13.: **Qualitative Results Pedestrian.** Qualitative detection (a) and tracking (b) results of our approach (left) compared to the ground truth (right).

Temporal Aggregation The capacity to model temporal information is crucial for a tracking model. In Table 3.6, we ablate the aggregation components of our model. The baseline represents our proposed method without access to history and motion prediction. We observe that with additional history frames, detection accuracy increases, albeit with a decrease in precision. This observation suggests that the model can detect more pedestrians with history frames, but location precision diminishes due to ambiguity introduced by the historical data. The incorporation of motion prediction primarily affects tracking results, leading to significant improvements in performance.

3.5.6. Qualitative Results

Pedestrian Detection & Tracking In Figure 3.13, we present the output of our model for the Wildtrack test set. Figure 3.13a illustrates a comparison between the POM prediction and the ground truth, represented as a heatmap for a single timestep. Each point in the ground truth corresponds to a pedestrian in the BEV. The central region of the scene, characterized by a high density of pedestrians, exhibits near-perfect prediction accuracy. However, the prediction accuracy diminishes for pedestrians located farther from the center. This degradation in accuracy may be attributed to increased distortion at greater distances from the cameras and reduced overlap of camera views in peripheral regions.

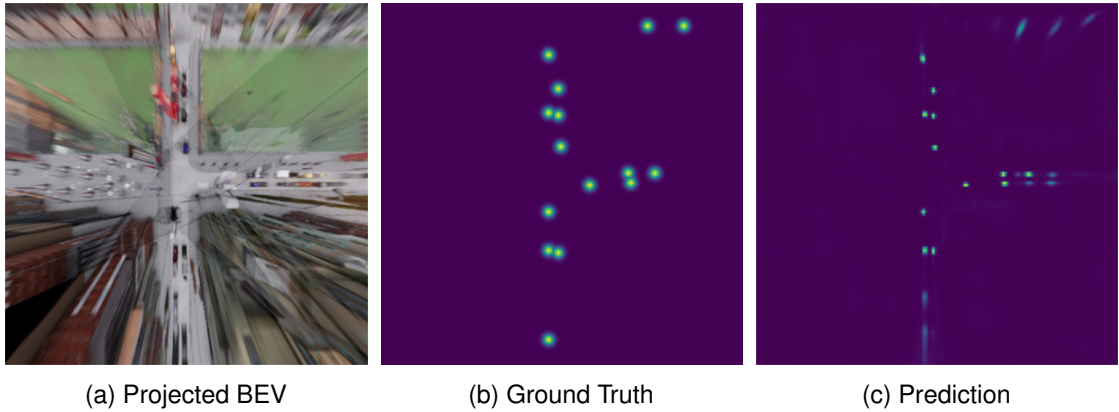


Figure 3.14.: **Qualitative Results Vehicles.** Detection example shown on Synthehicle. (a) shows the input images projected to the BEV space, (b) shows the ground truth heatmap of all vehicles, and (c) our prediction with bilinear sampling.

Figure 3.13b presents analogous results for tracking. We visualize all tracks on the ground plane for the entire Wildtrack test set, with each color and line representing the trajectory of a distinct identity. The tracks in the central area of the scene demonstrate exceptionally high prediction accuracy. In contrast, tracks in the top-left and top-right regions exhibit segmentation or identity switches. This discrepancy may be a consequence of less accurate or absent detections, as well as potentially compromised identity features outside the primary ground plane.

Synthehicle Detection Figure 3.14 presents an example from the Synthehicle validation set. In Figure 3.14a, we employ perspective projection of all camera views onto the ground plane, providing an approximation of the scene’s BEV. This projection also serves as a representation of the image post-encoder projection. Notably, image features or pixels exhibit greater elongation in the peripheral regions of the scene. Figure 3.14b displays all objects in the ground truth. When compared to the prediction (Figure 3.14c) generated by our approach utilizing the bilinear lifting method, we observe promising results in the central area of the scene. However, the prediction efficacy diminishes towards the outer regions. Interestingly, our predictions also indicate the presence of vehicles in areas not highlighted in the ground truth but visible in the projected BEV-View. This discrepancy may be attributed to Synthehicle’s labeling methodology, which is based on 2D views and may consequently omit certain 3D detections.

3.5.7. Computational Efficiency

A significant differentiating factor among the lifting methods employed in our study is their computational efficiency. Figure 3.15 presents a comparative analysis of the Floating Point Operations (FLOPs) and parameters in relation to the MOTA score on MultiviewX. Our findings indicate a positive correlation between the number of FLOPs

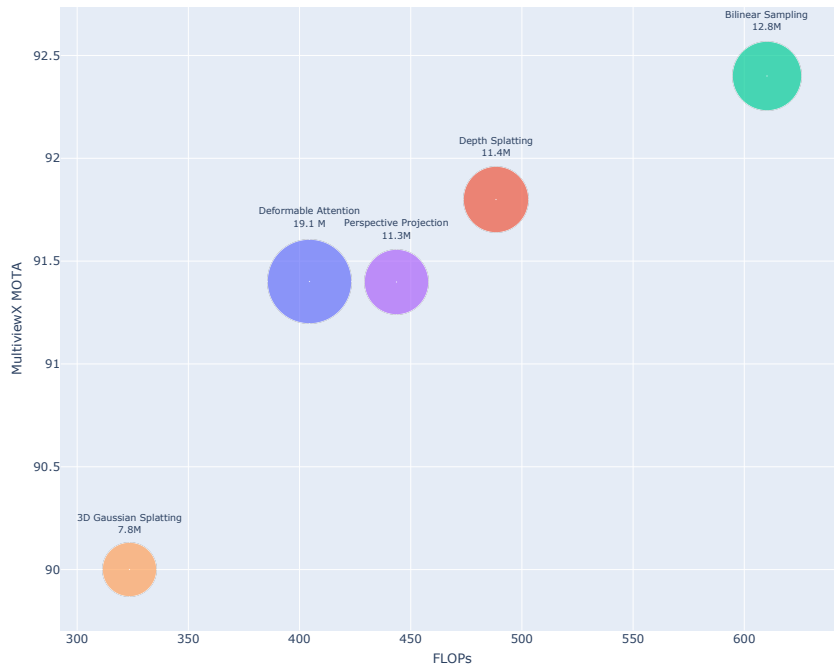


Figure 3.15.: **Computational Efficiency of Lifting Methods.** Comparison of the computational efficiency of different lifting methods.

and the MOTA score, with the method utilizing the most FLOPs achieving the highest MOTA score. When compared to perspective projection, another parameter-free lifting method, our approach exhibits a similar parameter count but a substantially higher FLOPs count. While lifting into a 3D space instead of a 2D BEV space enhances performance, it does so at the expense of increased computational complexity.

Gaussian splatting emerges as the most computationally efficient method, demonstrating significantly lower parameter and FLOPs counts. However, this efficiency comes at the cost of performance, as evidenced by its lowest MOTA score. This trade-off can be attributed to its utilization of a sparse 3D space representation. In contrast, the other parameterized lifting methods, namely depth splatting and deformable attention, exhibit a more balanced performance-efficiency profile. The transformer-based method, as anticipated, presents a higher parameter count but lower FLOPs compared to depth splatting.

In conclusion, our analysis identifies three lifting methods that offer a balanced compromise between performance and efficiency: perspective projection, depth splatting, and deformable attention. For applications where computational efficiency is paramount, we recommend the implementation of the gaussian splatting method.

3.6. Discussion

Limitations Our approach exhibits several notable limitations. Primarily, it necessitates high-fidelity 3D annotations and precise camera calibrations. While this requirement is readily met with synthetic data, it presents a significant cost barrier for real-world datasets. Consequently, we were unable to evaluate certain legacy datasets (e.g., CAMPUS [162], PETS09 [46]), where late-fusion models can operate effectively with only 2D annotations and ground plane homography.

Furthermore, our methodology mandates synchronized camera inputs. Given that we project all camera feeds into a unified 3D space, temporal discrepancies must be minimized to prevent moving objects from being projected to disparate 3D locations. In contrast, late-fusion MTMC tracking approaches can accommodate greater temporal drift. Additionally, our approach imposes higher hardware demands. While late-fusion methods can process each camera feed independently and centralize fusion subsequently, our approach necessitates simultaneous processing of all camera images, thus requiring enhanced memory and computational resources on a single machine.

Ethical Impact Tracking methodologies inherently carry the risk of misuse for illicit surveillance activities. Pedestrian-focused tracking methods, in particular, must confront this ethical scrutiny. The Wildtrack dataset [22] has faced criticism [59] for its lack of explicit consent from recorded individuals. Despite these concerns, a comprehensive comparison with state-of-the-art methods necessitates the use of this dataset, although a synthetic alternative, MultiviewX [70], is now available. In an effort to mitigate ethical implications, we have pioneered the evaluation of tracking on this synthetic dataset.

Vehicle-centric datasets such as Synthehicle [3[†]] present fewer ethical concerns, as they are primarily utilized for traffic surveillance. However, the potential for vehicle tracking to be employed in broader surveillance contexts requires ongoing ethical consideration.

Future Work In the domain of detection, the primary challenge faced by our approach and contemporary methods [69, 87, 140] is the distortion induced by projective transformation. Future research could explore alternative 2D-to-3D lifting techniques for multi-view detection, such as Simple-BEV [56], Lift-Splat-Shoot [118], or BEVFormer [92].

Current approaches predominantly utilize single-frame information for detection. Incorporating multi-frame temporal data could potentially enhance detection performance [92]. Similarly, leveraging extended temporal context could improve tracking quality, with methodologies such as CenterTrack [180] offering promising avenues for motion-based tracking.

The pedestrian datasets employed in this study comprise approximately 400 time-stamps, which is relatively limited by contemporary computer vision standards. As detection and tracking accuracy approaches saturation on these datasets, the need for more extensive and challenging datasets becomes apparent. Datasets addressing similar MTMC challenges in traffic surveillance contexts [147, 3[†]] could potentially bridge this gap.

3.7. Conclusion

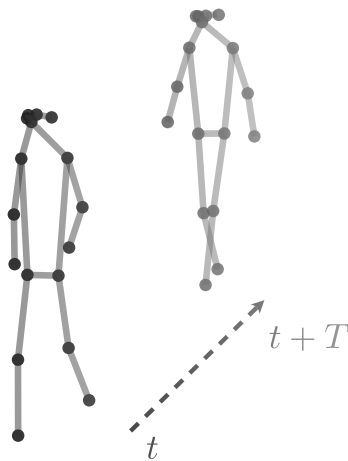
This chapter has presented a comprehensive analysis of diverse lifting strategies for multi-target multi-camera (MTMC) tracking. Our approach, which integrates these lifting strategies with a motion-based tracking method, has demonstrated state-of-the-art performance on two established pedestrian datasets: Wildtrack and MultiviewX. However, our results also indicate a performance plateau on these datasets, suggesting the need for more challenging benchmarks in the field.

To address this limitation, we extended our evaluation to include the roadside perception dataset, Synthehicle, which introduced a new dimension to the MTMC challenge: scene generalization. This expansion enabled us to assess the robustness and adaptability of our approach across diverse environments, a crucial aspect for real-world applications.

Our investigation has revealed a significant gap in the current landscape of MTMC datasets. While existing benchmarks predominantly focus on 2D detections, our findings underscore the necessity for 3D-first datasets to serve as standard benchmarks in the field. This shift towards 3D-centric data would better align with the spatial nature of multi-view tracking and potentially unlock new avenues for performance improvement.

The lifting strategies explored in this work, ranging from perspective projection to Gaussian splatting, offer valuable insights into the trade-offs between computational efficiency and tracking accuracy. Our results demonstrate that more sophisticated lifting methods can yield substantial improvements in tracking performance, particularly in challenging scenarios with occlusions and varying camera perspectives.

In conclusion, our research not only advances the state-of-the-art in MTMC tracking but also highlights critical areas for future development in the field. We anticipate that our approach and findings will catalyze the creation of more sophisticated MTMC datasets and inspire novel one-shot multi-view detection and tracking methodologies. As the field progresses, we expect to see an increased focus on 3D-aware approaches and more diverse, challenging datasets that push the boundaries of MTMC tracking capabilities.



4

Skeleton-Based Gait Recognition

One of the challenges in camera vision is identifying individuals from a significant distance. A commonly used biometric to recognize pedestrians is gait. Gait refers to the manner or pattern of movement of the limbs during locomotion over a solid surface, such as walking or running. It involves coordinated and rhythmic steps of an individual, influenced by factors such as balance, coordination, and muscle strength. Or, in layman's terms, gait is the way someone walks.

In surveillance scenarios, gait can be used to identify criminals, but it can also be used on private properties to distinguish residents from trespassers. Gait recognition models also build the basis for other automated gait analysis tasks, such as psychological traits [28, 13], cognitive fatigue [115], age classification [1]. Gait recognition is considered a soft biometric compared to hard biometrics such as face recognition [7[†], 5[†], 6[†]]. Soft biometric traits are natural and are used by humans to identify each other, but are not unique to a single individual [31]. For example, one might recognize a friend from a distance by their stature and how they walk. This characteristic reveals an advantage of gait recognition: it can be used to identify individuals from a distance and without their consent. This property is a significant advantage in surveillance scenarios, but it also poses risks to privacy and misuse. Although most tracking algorithms rely on appearance-based [4[†]] Re-Identification (re-ID) features that may constantly change, the gait of a person is a more stable feature that allows

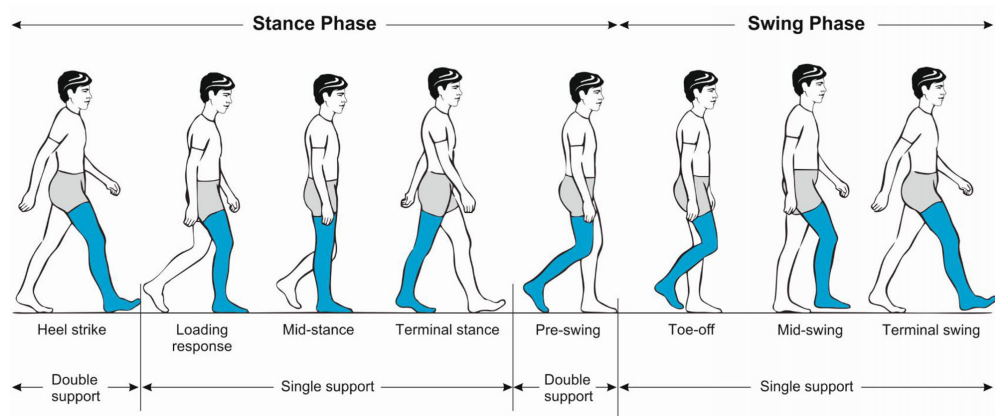


Figure 4.1.: **Gait Cycle.** The gait cycle is the sequence of movements of the limbs during locomotion. The gait cycle is divided into two phases: the stance phase and the swing phase. The stance phase is the period when the foot is in contact with the ground, and the swing phase is the period when the foot is not in contact with the ground.

identification over long periods. Compared to hard biometrics, gait can change due to aging or injuries but can also be temporarily altered by carried items or worn shoes. These external factors are the biggest challenge for gait recognition algorithms, as they must be robust against changes in clothing, items carried, and shoes. The term gait recognition is used here primarily for temporal movement, but, as we will show in this chapter, spatial posture, and statute are also significant features of gait recognition.

Many approaches, including [93, 55, 160, 45, 5, 139, 20, 41], use silhouettes to extract gait characteristics from a video sequence. Silhouettes are usually obtained using background subtraction techniques. However, this method may generate undesired artifacts in the contour, as shown in the two center frames in Figure 4.2, depending on the technique used for background subtraction. Although background subtraction can be reliable in a lab setting, it becomes a complex problem in a real-world scenario with cluttered and rapidly changing environments. Most approaches do not consider how to obtain the silhouettes, except for Song et al. [139], who trained a separate CNN for background subtraction in a real-world scenario. Silhouette representation has a significant drawback due to its sensitivity to clothing and carried items. The person's bag, for example, is visible in the silhouette in Figure 4.2. Moreover, the silhouette reveals many appearance clues, such as hairstyle and clothes, making these approaches more comparable to re-identification tasks.

Recent deep learning-based pose estimation allows the generation of robust keypoints against occlusion, cluttered, and changing backgrounds [143]. These advantages make an approximation of gait features with appearance-based methods obsolete and enable a new generation of model-based gait recognition. Building upon keypoints brings gait recognition back to an early description of gait [77]:

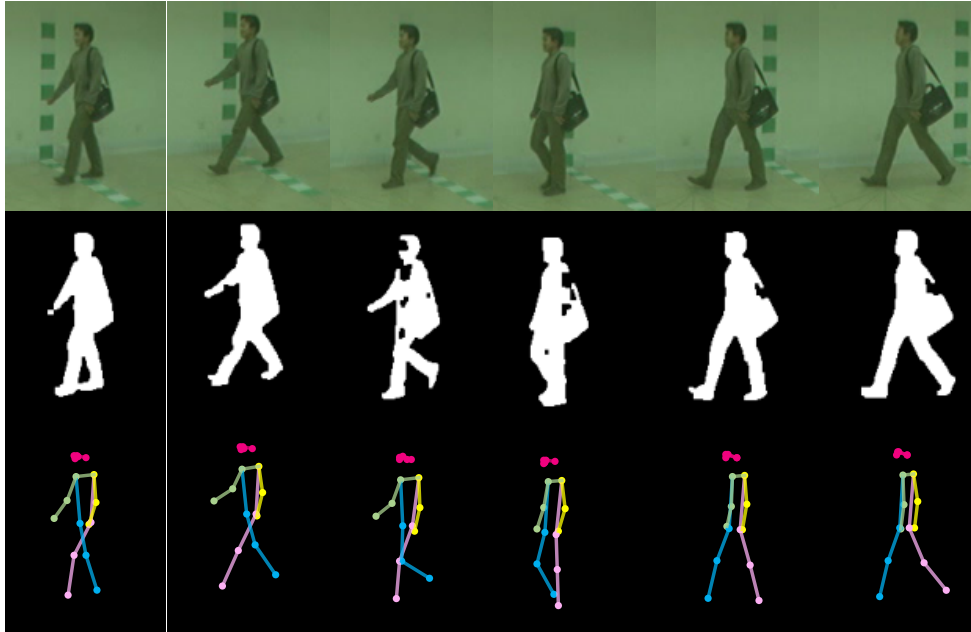


Figure 4.2.: **Gait Representations.** Comparison of the captured RGB image to the two commonly used gait representations: silhouette segmentation and skeleton model (top to bottom)

A few bright spots describing the motions of the main joints [...] evoke a compelling impression of human walking. – *G. Johansson et al. (1973)*

Skeleton-based approaches offer a cleaner gait representation, only capturing the spatial posture and the temporal movement. Thus, we can bring back the *actual* gait with a focus on motion recognition instead of visual recognition.

Other model-based approaches [138, 94, 95] still fall short in their performance compared to appearance-based methods. We introduce Graph Convolutional Networks (GCNs) to process the skeletons described by the keypoints and further bridge the gap to appearance-based methods.

Our contributions in this chapter can be summarized as follows:

1. We use a modern interpretation of model-based gait recognition, exploiting robust human pose estimation and powerful temporal and spatial modeling of GCNs.
2. We propose a multi-branch graph-based network architecture for gait recognition that precomputes velocities, bone lengths, and angles for robust temporal gait features.
3. We conducted extensive experiments on multiple datasets to compare our approach to state-of-the-art methods and ablate our model-based features.
4. We provide a deeper understanding of gait with extensive ablation and visualization of our model-based features.

The code and models are publicly available¹ and integrated on popular gait frameworks².

This chapter summarizes pre-published works in **GaitGraph** [9[†]] and **GaitGraph2** [8[†]] and extends the work with new experiments and insights. It builds on the student work of Ali Khan [2⁺].

4.1. Related Work

In this section, we establish an overview of the history of gait recognition and the different representations used in the field. Afterward, we separately reviewed model-based, model-free, and hybrid approaches in gait recognition. We provide an overview of the different methods in Table 4.1 for quick comparison. Finally, we discuss recent advances in skeleton-based action recognition as they provide the central network architectures of our approach.

4.1.1. Gait Representations

Finding a representation that abstracts the RGB image into a representation that preserves the gait information but does not contain appearance information is the key to gait recognition. Ideally, the representation should be invariant to changes in clothing, items carried, or occlusion.

In recent years, silhouette-based approaches dominated gait recognition. As modeling temporal information explicitly in neural architecture was computationally expensive, the first methods [55, 4, 64, 66, 65] aggregate a gait sequence into a single representation (i.e., Gait Energy Image (GEI)). Model-based approaches only played a minor role, but recently, new model-based approaches have emerged using pose estimation and the human skeleton as the gait feature representation. Silhouette-based methods still set the state-of-the-art for gait recognition, but skeleton-based approaches have started challenging this lead. Our work also shows the powerful temporal modeling capabilities of the model-based approaches. The promising results of our approach caused a new wave of research in model-based gait recognition, with new model-based representations emerging. Combining silhouette- and model-based input advantages also shows a significant performance increase. More recent approaches [133], also released a data set, SUSTech1K, which recorded pedestrians with a synchronized camera and a Light Detection and Ranging (LiDAR) setup to obtain depth information for more structured 3D gait representations. Overall, we can observe a trend towards more structured or model-based representations. In the following paragraphs, we will discuss the history of gait representations.

¹<https://github.com/tteepe/GaitGraph2>

²<https://github.com/BNU-IVC/FastPoseGait> & <https://github.com/ShiqiYu/OpenGait>

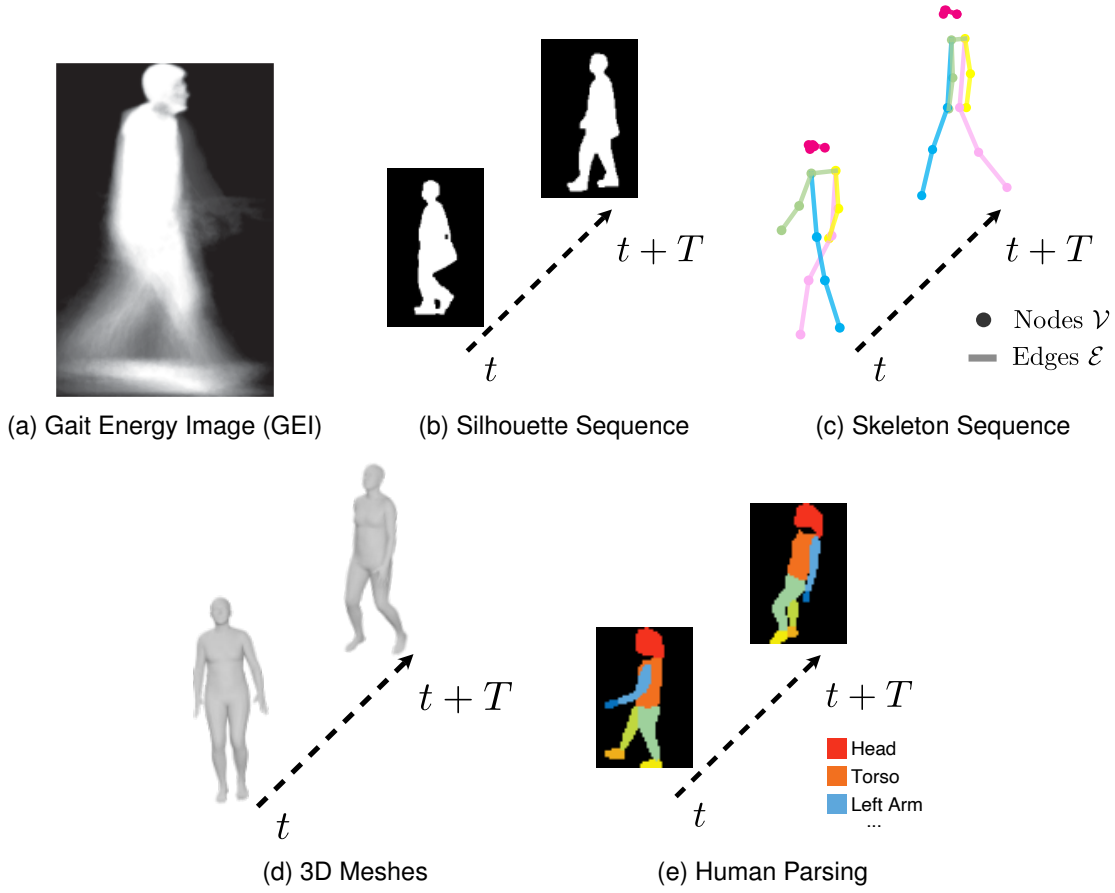


Figure 4.3.: **History of Gait Representations.** The history of gait representations from the early GEI to the recent skeleton model. Starting from GEI that averages the temporal dimension to a single image. The silhouette sequence captures the temporal information in a 3D tensor. The skeleton sequence captures the 2D or 3D keypoints of the human body. The 3D meshes capture the 3D shape of the human body.

Gait Energy Image (GEI) The GEI proposed by Han *et al.* [55] is a widely used representation in gait recognition. It is a temporal average of the silhouettes of a gait cycle. The GEI is a 2D image that captures the spatial and temporal information of a gait cycle, as shown in Figure 4.3a. The representation builds on the normalized binary silhouette $B_t(\mathbf{X})$ at time t extracted from the input RGB image tensor $\mathbf{X}$. The gray-scale gait energy image is calculated by taking the average over the binary silhouettes:

$$GEI = \frac{1}{N} \sum_{t=1}^N B_t(\mathbf{X}_t), \quad (4.1)$$

where N is the number of frames in a gait cycle. The compact representation was an advantage for early learning algorithms. Later, different approaches extended GEI by incorporating more aspects like the Depth Gradient Energy Image [64] (extended with depth information) and the α -part-based Gradient Histogram Energy Image [66] (with part-based edge information).

Silhouette Sequence Even though the silhouette sequence is the base of GEI, it became popular later [160, 139, 20, 41] as deep learning methods moved from 2D images to 3D video inputs. The silhouette sequence is a temporal sequence of binary silhouettes extracted from the RGB images. The silhouette sequence is a 3D tensor with time, height, and width dimensions, as shown in Figure 4.3b. The silhouette sequence is a more detailed representation than the GEI and can capture more temporal information. The silhouette or foreground is usually obtained by subtraction of the background. This method is reliable in lab conditions, but in real-world scenarios with cluttered and rapidly changing environments, this becomes a complex problem. In real-world scenarios, the silhouette is obtained by deep learning methods [175, 183]. This extraction can have undesired artifacts in the contour, as shown in the two center frames in Figure 4.2. New emerging works extend this representation with more structured segmentation information, but to this day it is the gold standard for gait recognition [176, 183].

Skeleton Sequence The skeleton representation is a model-based representation that emerged in gait recognition [94] in recent years. The skeleton sequence is a temporal sequence of 2D or 3D keypoints extracted from the RGB images. The skeleton sequence is a combination of joints, nodes, joint connections, and edges, as shown in Figure 4.3c. This representation is only made possible by the recent advances in human pose estimation [17, 43]. These pose estimators can extract accurate joint points from RGB images. Since these estimators are robust against occlusion, cluttered, and changing backgrounds, the skeleton representation is more stable than the silhouette representation. Figure 4.2 shows a person carrying a bag in RGB, silhouette, and skeleton images. The silhouette image shows the bag, while the skeleton image does not.

We interpret the skeleton as a graph, where the joints are the nodes and the connections between the joints are the edges. This graph representation allows us to use graph-based neural networks, such as GCNs, for gait recognition.

3D Meshes Another model-based gait representation is the 3D mesh representation [175, 183]. Like skeleton sequences, this representation was enabled by advances in 3D human shape and pose estimation [12, 145, 24]. Among them, the Skinned Multi-Person Linear (SMPL) model [103] is a skinned vertex-based model that can accurately represent a wide variety of body shapes in natural human poses, as shown in Figure 4.3d. The SMPL model captures 24 3D body keypoints, a vector of dimension 10 for the 3D body shape, and three parameters for the camera scale and translation. Based on the SMPL model, a series of 3D human mesh recovery approaches are developed to estimate accurate 3D shapes, poses, and viewpoints of human bodies from RGB images [12, 145, 24]. The advantages of 3D meshes for gait recognition are that they capture the pose and the human body shape in 3D space combined with a 3D normalized representation.

Human Parsing A recent promising approach, Gait Parsing Sequence (GPS) [177], uses human segmentations as input. Instead of having a binary silhouette image with foreground and background, a human parser [101] predicts different parts of the body or cloth, such as the head, left arm, torso, or dress, as shown in Figure 4.3e. This representation is a hybrid between the model- and silhouette-based approaches, as it incorporates more model information into the silhouette representation. Since the human parser is robust against occlusion, cluttered, and changing backgrounds, the GPS representation is more stable than the silhouette representation. Another advantage is that the GPS representation can be used with all existing silhouette-based approaches, as the GPS representation, instead of binary classification, uses multiple classes for each part, leading to significant performance increases [177].

4.1.2. Silhouette-based Gait Recognition

Silhouette-based methods, which rely on binary human images extracted from RGB inputs [93], encounter significant challenges in dynamic and evolving environments. While background subtraction techniques can effectively obtain silhouettes in static scenes, they become increasingly complex in dynamic settings. However, recent advancements in deep learning have facilitated silhouette extraction even in these challenging scenarios [139].

Silhouette approaches are characterized by their temporal modeling strategies and can be categorized into single-image, sequence-based, and set-based methodologies. Early approaches condensed a complete gait cycle into a single image, such as the GEI [135, 139, 55, 5, 64, 160, 161]. While these representations sacrifice most temporal information, they allow for more straightforward processing. Notable examples include GEINet [135], which proposed a concise Convolutional Neural Network (CNN) architecture comprising only four layers. This approach directly learns gait representations from GEI and maps them to corresponding identities. The TS-CNN framework [161] employs a two-stream CNN architecture to learn similarities between GEI pairs for gait recognition, outputting the probability of two subjects sharing the same identity. GaitNet [139] was the first to claim an end-to-end approach, proposing a gait recognition pipeline that extracts the foreground from RGB images, compresses the silhouette sequence into a GEI, and utilizes a CNN to learn the gait representation. This approach also introduced the Outdoor-Gait dataset, featuring gait sequences in controlled outdoor environments.

Sequence-based approaches consider each time step as an input. Early methods employed classical techniques to learn temporal information, such as 3D-CNN [160] or Long Short-Term Memorys (LSTMs) [45]. More successful approaches treated the sequence as a set of images and utilized set-based methods to capture temporal information [139, 20, 41, 96]. Specifically, GaitSet [20] extracts gait representations from unordered silhouette sets and employs a maximum function to compress the sequence of frame-level features into a single set-level feature. It also adopted the Triplet Loss [131] for gait recognition. Building upon GaitSet's success, subsequent

works such as GaitPart [41], GLN [68], and MT3D [96] utilized similar architectures but focused on part-level features, an idea also employed in re-ID and our work with LightMBN [4[†]], highlighting the strong connection between re-ID and silhouette-based gait recognition. GaitGL [97] argued that spatially global gait representations neglect local region-based representations and introduced a global and local convolution layer [97] aimed at learning relations among neighboring parts. CSTL [72] concentrated on temporal features at three scales to obtain motion representations based on temporal contextual information. 3DLocal [73] proposed an architecture that learns temporal features at a local part level with 3D local operations at adaptive scales. SPOSGait [54] introduced an approach obtained through Neural Architecture Search (NAS) that optimizes an architecture based on 3D convolution and separation of body part features. Fan *et al.* released OpenGait, a new framework integrating all existing gait datasets and providing a unified evaluation platform for gait recognition. Alongside this framework, they introduced GaitBase [39], a new gait model combining substantial enhancements with a ResNet-like backbone and temporal pooling to achieve strong state-of-the-art performance on larger outdoor datasets.

Recent approaches have expanded beyond binary silhouette images, utilizing human parsing [177] to extend silhouette images to segmentation images. The authors propose an architecture employing two branches: an encoder combined with GaitSet [20] and a parts head that correlates parsed features of human body parts. The authors demonstrate that human parsing can be integrated with existing silhouette-based approaches, as it provides a more detailed version of the silhouette representation, leading to significant performance improvements [177].

4.1.3. Model-based Gait Recognition

Model-based approaches initially dominated gait recognition [164, 3]. These methods defined a structural human body model and recognized gait patterns by measuring parameters such as limb lengths, joint angles, and relative positions of body parts. However, the requirement for controlled environments and multiple sensors led to a decline in their popularity as silhouette extraction techniques became more reliable and accessible.

Recent advancements in extraction methods [17, 43, 12, 145, 24] have reignited interest in model-based approaches. This renaissance began with the estimation of 2D human keypoints, enabling the modeling of skeletal representations. Pose-Based Temporal-Spatial Network (PTSN) [94] pioneered the use of pose estimation and keypoints, proposing a two-path network architecture combining CNN for spatial modeling and LSTM for temporal modeling. The authors demonstrated superior results on CASIA-B [169] compared to silhouette-based approaches in specific walking conditions and view settings.

Subsequently, *PoseGait* [95] introduced 3D pose estimation with handcrafted features. This approach utilized 3D keypoints in Euclidean space to calculate joint angles, bone

Baseline	Year	Input	# Layer	# Embed.	Loss
GEINet [135]	2016	GEI	4	N	Softmax
TS-CNN [161]	2016	GEI	6	1	2-Cls Cross-Entropy
GaitNet [139]	2019	RGB / GEI	14	N	Softmax & 2-Cls CE
GaitSet [20]	2019	Silhouette	10	256	Triplet
GaitPart [41]	2020	Silhouette	8	256	Triplet
GLN [68]	2020	Silhouette	6	256	Triplet
GaitGL [97]	2021	Silhouette	12	256	All-Batch Triplet & CE
CSTL [72]	2021	Silhouette	7	256	All-Batch Triplet & CE
SPOSGait [54]	2022	Silhouette	11	256	All-Batch Triplet & CE
GaitBase [38]	2023	Silhouette	6	256	All-Batch Triplet & CE
PoseGait [95]	2020	2D & 3D Pose	8	512	Softmax & Center
GaitGraph [9 [†]]	2021	2D Pose	9	256	Contrastive
GaitGraph2 [8 [†]]	2022	2D Pose	12	256	Contrastive
Gait-TR [171]	2023	2D Pose	6	128	All-Batch Triplet
GPGait [48]	2023	2D Pose	6	256	All-Batch Triplet & CE
SkeletonGait [40]	2024	Pose Heatmap	6	256	Triplet & CE
BiFusion [117]	2021	Skeleton & Silh.	6+6	128	All-Batch Triplet & CE
SMPLGait [176]	2022	3D Mesh & Silh.	3+6	256	All-Batch Triplet & CE
MMGaitFormer [30]	2023	Skeleton & Silh.	9+12	256	All-Batch Triplet
SkeletonGait++ [40]	2024	Pose Heat. & Silh.	2×2+4	256	All-Batch Triplet & CE

Table 4.1.: **Overview of Gait Approaches.** Overview of the related work section for quick reference: including input modalities, number of network layers, dimensions of embedding feature, and loss.

lengths, and joint motions. A CNN then learned high-level spatio-temporal features from these handcrafted features. PoseGait showed competitive results in cross-view settings on CASIA-B [169] compared to silhouette-based approaches.

Our work, GaitGraph [9[†]], introduced the first approach using GCNs for skeleton-based gait recognition. By adapting a GCN architecture from action recognition and integrating spatial and temporal modeling, GaitGraph achieved a significant performance increase in model-based gait recognition, providing the first comparative results to silhouette-based approaches. We later extended this approach [8[†]] with a multi-branch input architecture that precomputes velocities, bone lengths, and angles, further improving recognition accuracy.

Building upon our work, Gait-TR [171] employed the same architecture but replaced the GCN encoder with a spatial graph transformer, ST-TR [121]. This transformer-based graph encoder outperformed ST-GCN and ResGCN in action recognition, with these performance gains translating to gait recognition as well. The success of transformer architectures in this context suggests promising future directions for model-based gait recognition.

A recent skeleton-based approach, GPGait [48], addresses the challenge of generalizing model-based approaches. The authors proposed normalizing skeleton data and extracting angle, bone, and joint characteristics. A part-aware GCN is then employed to learn robust gait representations that generalize better to unseen datasets. This focus on generalization is particularly important as gait recognition

systems are increasingly deployed in real-world scenarios with varying conditions and environments.

4.1.4. Multi-Modal Gait Recognition

Hybrid approaches combine the advantages of silhouette-based and model-based methods. The first group of approaches [153, 30, 117] combine skeleton and silhouette inputs. Wang *et al.* [153] proposes a simple two-branch GCN and CNN network architecture and achieves even higher recognition rates. This shows that skeleton-based approaches preserve information that is not captured by silhouette-based methods. BiFusion [117] extends our approach, GaitGraph [8[†]], with a multiscale graph network that handles the skeleton at different levels of granularity and combines it with GaitPart [41] in a multimodal architecture. The authors show that the performance of their multiscale GaitGraph outperforms the single-scale GaitGraph. A similar approach is proposed by MMGaitFormer [30]; it combines a GCN pose encoder with a CNN silhouette encoder and focuses on the fusion of the two modalities. The authors propose a spatial and temporal attention mechanism to fuse both modalities.

Another input based on models, 3D human meshes, was first proposed by [90], as it contains not only the 3D skeleton, but also the 3D shape of the human body. SMPLGait [176] used the 3D human mesh in a hybrid architecture that combines the GaitSet branch for silhouette images with a 3D mesh branch that uses fully connected layers. Both features are fused in the set-pooling and horizontal pyramid pooling introduced by GaitSet. Unfortunately, the authors do not report results that are only mesh-based, making comparing skeleton-based approaches difficult.

In SkeletonGait [40], authors propose a changed version of the skeleton data by using pose heatmaps instead of the extracted key points. The keypoint maps are neither a full model-based approach nor a complete silhouette-based approach. Still, a hybrid such as the keypoint heatmap is an image that captures the skeleton probabilistically. The authors used the heatmap with an image-based CNN backbone. In their extension SkeletonGait++ [40], the authors combine both silhouette and heatmap input in a mid-fusion architecture that claims the current state-of-the-art for multimodal gait recognition.

4.1.5. Novel Research Directions in Gait Recognition

Beyond the specific gait representation and recognition models, several innovative trends are emerging in the field of gait recognition. This section explores some of the most promising developments.

While the majority of approaches discussed thus far rely on supervised learning, there is a growing interest in unsupervised methods for learning gait representations without labeled data. Fan *et al.* [38] have made significant contributions in this

area by introducing GaitLU-1M, a large-scale unlabeled gait dataset comprising 1.02M walking sequences. Additionally, they proposed GaitSSB, a baseline model that learns gait representations through self-supervised techniques. Their research demonstrates that these learned representations can be effectively applied to gait recognition tasks, in some cases outperforming supervised approaches on certain datasets.

Traditionally, gait recognition architectures have relied on a pre-processing step to extract gait representations. However, end-to-end models [90, 139] are gaining traction by integrating gait representation learning and recognition within a single unified framework. While these models tend to be more complex and computationally intensive, they have the potential to achieve superior performance compared to models with separate pre-processing stages.

Another notable trend is the exploration of alternative sensors for capturing gait information. Shen et al. [133] have pioneered the use of LiDAR sensors in gait recognition, demonstrating that LiDAR-based approaches can surpass RGB-based methods in certain scenarios. Their work includes the release of the SUSTech1K dataset, which features synchronized camera and LiDAR recordings of pedestrians, providing depth information for more structured 3D gait representations. In a different approach, EV-Gait [156, 148] utilizes event cameras with dynamic vision sensors (DVS) to capture gait information. The authors highlight the robustness of event cameras in capturing gait information compared to RGB cameras, particularly in low-light conditions.

4.1.6. Skeleton-based Action Recognition

Skeleton-based gait recognition is a relatively recent development, but other domains of human understanding have been utilizing skeleton features for several years. This area has evolved through various deep learning trends, from CNNs to Recurrent Neural Networks (RNNs) and GCNs. Notably, the field of skeleton-based action recognition has demonstrated the most promising results and has pioneered many of the current graph-based spatiotemporal GCN architectures, including those employed in this work [165, 141].

Recognizing the limitations of traditional deep learning methods in handling the non-Euclidean nature of skeleton data, researchers began exploring Graph Convolutional Networks (GCNs). GCNs are specifically designed to operate directly on graph-structured data, making them particularly suitable for modeling human skeletons, which naturally form a graph with joints as nodes and bones as edges.

Yan *et al.* [165] pioneered the application of GCNs to action recognition with the Spatial-Temporal Graph Convolutional Network (ST-GCN) architecture. This architecture represents skeleton data as a graph with natural skeleton connections, interleaving spatial graph convolutions and temporal convolutions for spatial-temporal modeling. Subsequent research has proposed numerous GCN architectures, each

improving upon the original ST-GCN design. A notable example is the Two-Stream Adaptive Graph Convolutional Network (2s-AGCN) [134], which introduced an adaptive adjacency matrix and incorporated precomputed bone information as second-order input.

SGN [173] provides a more in-depth analysis of these second-order features and proposes a data preprocessing step that augments the raw keypoint input with precomputed velocity and bone information. ResGCN [141] integrates residual connections into GCNs to address the problem of vanishing gradients, enabling the construction of deeper networks. By incorporating these residual connections, ResGCN enhances feature propagation across layers, improving the network's ability to capture complex spatio-temporal dependencies in skeleton data.

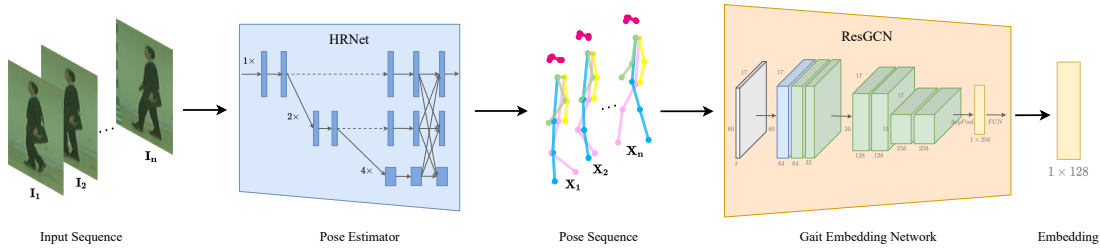
A recent advancement in skeleton-based action recognition is the Spatial Graph Transformer (ST-TR) [121], which employs the self-attention mechanism to model long-range dependencies in both spatial and temporal dimensions. ST-TR applies a transformer architecture to the skeleton data, enabling it to dynamically focus on relevant joints and time steps throughout the sequence, further enhancing the model's ability to capture complex motion patterns.

4.2. Overview of GaitGraph

GCNs have emerged as a powerful tool for skeleton-based action recognition [141, 173, 165], leveraging their ability to model complex relationships within graph-structured data. In this work, we adapt these architectures for gait recognition. We initially employed the ResGCN (Residual Graph Convolutional Network) [141] architecture, originally designed for action recognition. The ResGCN architecture builds upon ST-GCN, a foundational model for action recognition; its general structure is illustrated in Figure 4.8.

To appreciate the advancements of ResGCN, it is crucial to understand the ST-GCN architecture, which pioneered the use of GCNs in skeleton-based action recognition. The strength of ST-GCN lies in its combination of spatial and temporal convolutions within each layer. Spatial convolutions operate on the graph structure, capturing inter-joint relationships in each frame, while temporal convolutions model motion across time by capturing inter-frame dependencies. Following the success of ResNet [61], ST-GCN incorporates residual connections around each block, facilitating the learning of more complex features by improving gradient flow.

ResGCN extends the ST-GCN architecture by introducing additional residual connections around each convolutional component. Furthermore, it incorporates a bottleneck structure, as shown in Figure 4.9, which reduces the dimensionality of features before the residual connections. This bottleneck design serves multiple



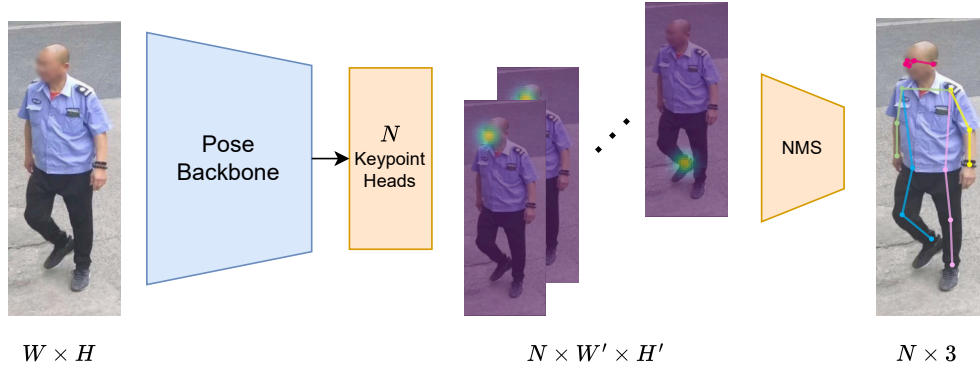


Figure 4.5.: **Human Pose Estimation with Heatmaps.** The architecture processes an input image through a backbone network to extract downsampled feature representations. Subsequently, the network predicts probability heatmaps for each keypoint. The final keypoint locations are determined by identifying the maximum probability coordinates within each respective heatmap.

and algorithms [24, 145]. Although our approach was originally designed with 2D poses, it is compatible with both 2D and 3D pose estimations, allowing for flexibility in the choice of pose estimator. An overview of the datasets and pose estimators used in this work is presented in Table 4.2.

In the following subsections, we first describe the general pose extraction process, followed by a presentation of the specific approaches employed in this work.

4.3.2. 2D Human Pose Estimation

The objective of 2D human pose estimation is to predict N joint locations (e.g., shoulder, hip, or knee) from an RGB image $I \in \mathbb{R}^{W \times H \times C}$. We first examine approaches for single pose estimation, where the input image contains a single subject, before extending to multi-pose estimation scenarios involving multiple subjects simultaneously.

Early approaches to single-pose estimation [18] employed end-to-end deep neural networks to directly regress joint locations from input images. However, contemporary methodologies [143, 17, 42] have demonstrated superior performance using heatmap-based approaches. These methods predict 2D heatmaps generated by applying 2D Gaussian kernels centered at each joint location, as illustrated in Figure 4.5.

Formally, the objective is to estimate N heatmaps $\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_N\}$ of dimensions $W' \times H'$. For each keypoint heatmap, the value $\mathbf{H}_n(x, y)$ represents the probability of keypoint n being located at position (x, y) . During training, the model minimizes the discrepancy between predicted and ground truth heatmaps using metrics such as mean squared error or Focal Loss [99]. This heatmap-based methodology offers

advantages over regression-based approaches by preserving spatial information while facilitating more stable training.

The position of keypoint v_n is determined by locating the maximum value in each heatmap $\mathbf{H}_n$, which subsequently defines the edges $\mathcal{E}$. All 2D models utilized in this work were trained on the COCO dataset [100], which provides annotations for 17 key points per pose. The skeletal structure, defined by bones or edges $\mathcal{E}$, is depicted in Figure 4.3c.

For multi-pose scenarios, two primary methodologies exist. The top-down approach employs a person detector to isolate individual subjects before applying pose estimation sequentially. While this method achieves higher accuracy through increased resolution per subject, the computational complexity scales linearly with the number of detected individuals. Alternatively, bottom-up approaches process the entire scene simultaneously, detecting all keypoints before grouping and associating them with individual subjects in post-processing. This methodology maintains constant computational complexity regardless of the number of subjects, though typically at the cost of reduced accuracy compared to top-down approaches.

Three prominent pose estimation models in this domain are HRNet [143], OpenPose [17], and AlphaPose [42], each offering unique strengths and contributions to the field:

- **HRNet** (High-Resolution Network) maintains high-resolution representations throughout the estimation process, contrasting with traditional CNNs that typically down-sample the image. The architecture incorporates repeated multiresolution fusion and multiscale supervision. While highly accurate, HRNet is designed for single-person pose estimation and requires a person detector for bounding box extraction, limiting its real-time performance in crowded scenes.
- **OpenPose** pioneered multi-person 2D pose estimation, capable of detecting keypoints for multiple individuals in a single image. It employs a two-stage architecture, first predicting confidence heatmaps for each keypoint, then utilizing Part Affinity Fields (PAFs) to model relationships between body parts. This approach enables real-time performance in crowded scenes.
- **AlphaPose** combines elements from both HRNet and OpenPose, focusing on improving accuracy and efficiency in crowded scenes. It utilizes a two-stage architecture with high-resolution representations and incorporates a Pose-Guided Proposals Generator for refinement. AlphaPose also introduces a specialized Non-Maximum Suppression (NMS) technique for handling overlapping poses, enhancing clarity and accuracy in multi-person scenarios.

By employing these three models, we can evaluate the impact of different pose estimators on our gait recognition approach and assess their relative performance impact.

4.3.3. 3D Human Pose Estimation

The field of 3D human pose estimation has witnessed significant advancements in recent years. The objective is to predict the 3D coordinates of human body joints from a single image or video frame. This task presents greater challenges than 2D pose estimation due to the need for depth information interpretation. 3D pose estimation enables the extraction of view-invariant human body skeletons, potentially addressing the challenge of cross-view gait recognition.

In this work, we utilize 3D keypoints extracted by two approaches: Chen *et al.* [24] and ROMP [145].

- The approach by **Chen *et al.*** [24] leverages existing 2D pose estimation techniques to facilitate 3D pose estimation. It employs a two-stage process: first estimating 2D poses using a state-of-the-art estimator, then matching these to the most similar 3D pose from a database. This method capitalizes on extensive 3D pose data to enhance estimation accuracy. While computationally efficient, it relies on database matching rather than direct 3D calculation from the image.
- **ROMP** [145] offers a more direct approach to 3D pose estimation. It proposes a one-stage monocular method for estimating 3D poses of multiple people in a single image, addressing challenges in real-time applications and crowded scenes. ROMP utilizes a common HRNet or ResNet backbone combined with multiple detection heads. The authors introduce a collision-aware loss function, enabling robust prediction under person-to-person occlusion. This model is capable of real-time performance in crowded environments.

By employing these two 3D pose estimators, we can evaluate the potential performance gains of a full 3D pose estimator compared to a pseudo-3D pose lifting approach in the context of gait recognition.

4.4. Gait Recognition with Graph Convolutional Networks

In this section, we explore the application of Graph Convolutional Networks (GCNs) to gait recognition. GCNs are particularly well-suited for processing skeleton data, as they can effectively capture the spatial and temporal relationships between body joints. We will discuss how we construct the skeleton graph, apply spatial-temporal convolutions, and leverage this architecture for robust gait feature extraction.

4.4.1. Skeleton Graph Construction

We approach the representation of a skeleton sequence through the 2D or 3D coordinates of each human joint within individual frames. Unlike previous models that leverage convolution for skeleton-based recognition by concatenating coordinate vectors of all joints into a single feature vector per frame [95], our method introduces

a spatial-temporal graph. This graphical approach yields a nuanced hierarchical representation of skeleton sequences, emphasizing both intrabody and interframe connections.

We define an undirected spatial-temporal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ tailored to a skeleton sequence that includes N joints across T frames. Here, the set of nodes $\mathcal{V} = \{v_{ti} \mid t = 1, \dots, T, i = 1, \dots, N\}$ represents all the joints throughout the sequence. The feature vector for a node $F(v_{ti})$ encapsulates both the coordinate vectors and the i -th joint estimation confidence for frame t , enriching the node's data with critical spatial and confidence information.

Constructing our spatial-temporal graph involves two primary steps:

1. Frame-Level Connectivity: Initially, we establish edges among the joints within the same frame on the basis of the human body's structural connectivity. This self-contained mapping within frames lays the groundwork for delineating intra-body relationships.

2. Temporal Connectivity: Subsequently, we create linkages that extend across frames by connecting each joint to its subsequent incarnation in the next frame.

This strategy fosters natural continuity across the sequence, eliminating the need for manual edge assignment and ensuring adaptability to varying joint counts and configurations. The edge set $\mathcal{E}$ comprises two distinct subsets:

$\mathcal{E}_S$: Represents intraskeleton connections within individual frames, based on naturally occurring body joint connections denoted $\mathcal{E}_S = \{v_{ti}, v_{tj} \mid (i, j) \in H\}$, where H delineates the human body's joint connectivity map.

$\mathcal{E}_F$: Facilitates inter-frame connections by linking identical joints in successive frames, expressed as $\mathcal{E}_F = \{v_{ti}, v_{(t+1)i}\}$. This subset charts the trajectory of each joint through time, leveraging temporal dynamics within the skeletal sequence.

4.4.2. Spatial Graph Convolutional Neural Network

In this section, we will develop the idea of spatial graph convolutions proposed by ST-GCN [165] and start by looking at the graph CNN model within a frame. In this case, in a single frame at time τ , there will be N joint nodes V_t , along with the skeleton edges $E_S(\tau) = \{v_{ti}, v_{tj} \mid t = \tau, (i, j) \in H\}$. Recall the definition of convolution operation on 2D natural images or feature maps, which we will regard as 2D grids. The output feature map of a convolution operation is again a 2D grid. With stride 1 and appropriate padding, the output feature maps can have the same size as the input feature maps. We will assume this condition in the following discussion. Given a convolution operator with a kernel size of $K \times K$ and an input feature map f_{in}

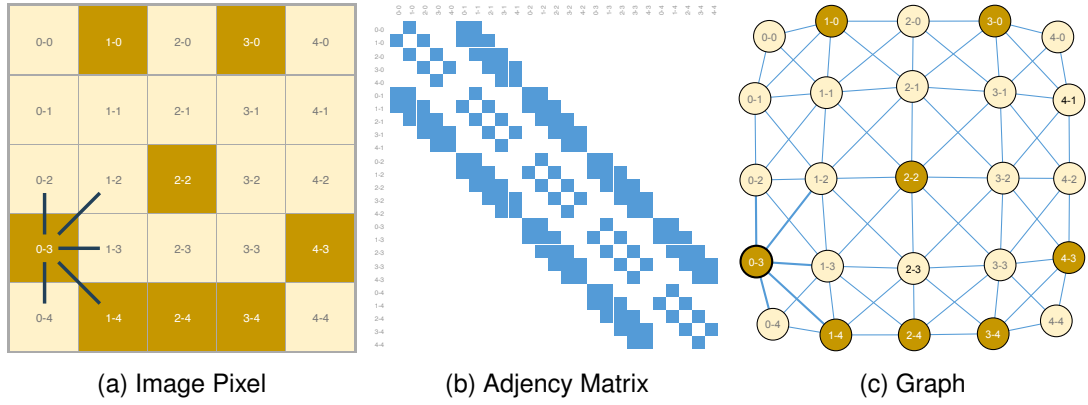


Figure 4.6.: **Images as Graphs.** The image pixel and the connection to their neighbors can be represented as a graph. The pixel is the node and the connections to the neighbors are the edges. These edges can also be represented as an adjacency matrix. Finally, we can plot the graph in a non-grid with the pixel nodes and 1-hop edges.

with the number of channels c . The output value for a single channel at the spatial location $\mathbf{x}$ can be written as:

$$f_{out}(\mathbf{x}) = \sum_{h=1}^K \sum_{w=1}^K f_{in}(\mathbf{p}(\mathbf{x}, h, w)) \cdot \mathbf{w}(h, w), \quad (4.2)$$

where the **sampling function** $\mathbf{p} : \mathbb{Z}^2 \times \mathbb{Z}^2 \rightarrow \mathbb{Z}^2$ enumerates the neighbors of location $\mathbf{x}$. In the case of image convolution, it can also be represented as $\mathbf{p}(\mathbf{x}, h, w) = \mathbf{x} + \mathbf{p}'(h, w)$. The **weight function** $\mathbf{w} : \mathbb{Z}^2 \rightarrow \mathbb{R}^c$ provides a weight vector in c -dimension real space for computing the inner product with the sampled input feature vectors of dimension c . Note that the weight function is irrelevant to the input location $\mathbf{x}$. Thus, the filter weights are shared everywhere in the input image. Standard convolution on the image domain is therefore achieved by encoding a rectangular grid in $\mathbf{p}(\mathbf{x})$.

The convolution operation on graphs is then defined by extending the formulation above to the cases where the input feature map resides on a spatial graph V_t . That is, the feature map $f_{in}^t : V_t \rightarrow \mathbb{R}^c$ has a vector at each node of the graph. The next step of the extension is to redefine the sampling function $\mathbf{p}$ and the weight function $\mathbf{w}$.

Sampling function On images, the sample function $\mathbf{p}(h, w)$ is defined in the neighboring pixels regarding the center location $\mathbf{x}$. In graphs, we can similarly define the sampling function in the set of neighbors $B(v_{ti}) = \{v_{tj} \mid d(v_{tj}, v_{ti}) \leq D\}$ of a node v_{ti} . Here, $d(v_{tj}, v_{ti})$ denotes the minimum length of any path from v_{tj} to v_{ti} . Thus the sampling function $\mathbf{p} : B(v_{ti}) \rightarrow V$ can be written as:

$$\mathbf{p}(v_{ti}, v_{tj}) = v_{tj}. \quad (4.3)$$

In this work, we use $D = 1$ for all cases: the set of 1-hop neighbors of joint nodes.

Weight function Compared to the sampling function, the weight function is more difficult to define. In 2D convolution, a rigid grid naturally exists around the center location. So, pixels within the neighborhood can have a fixed spatial order. The weight function can then be implemented by indexing a tensor of (c, K, K) dimensions according to spatial order. There is no such implicit arrangement for general graphs like the one we just constructed. The solution to this problem is investigated in [111], where a graph labeling process in the neighboring graph around the root node defines the order. We follow this idea to construct our weight function. Instead of giving every neighbor node a unique label, we simplify the process by partitioning the neighbor set $B(v_{ti})$ of a joint node v_{ti} into a fixed number of subsets K , where each subset has a numeric label. Thus, we can have a mapping $l_{ti} : B(v_{ti}) \rightarrow \{0, \dots, K-1\}$ that maps a node in the neighborhood to its subset label. The weight function $\mathbf{w}(v_{ti}, v_{tj}) : B(v_{ti}) \rightarrow R^c$ can be implemented by indexing a tensor of (c, K) dimension or

$$\mathbf{w}(v_{ti}, v_{tj}) = \mathbf{w}'(l_{ti}(v_{tj})). \quad (4.4)$$

Spatial Graph Convolution With the refined sampling function and weight function, we now rewrite Equation (4.2) in terms of a graph convolution as

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(\mathbf{p}(v_{ti}, v_{tj})) \cdot \mathbf{w}(v_{ti}, v_{tj}), \quad (4.5)$$

where the normalizing term $Z_{ti}(v_{tj}) = |\{v_{tk} \mid l_{ti}(v_{tk}) = l_{ti}(v_{tj})\}|$ equals the cardinality of the corresponding subset. This term is added to balance the contributions of different subsets to the output. Substituting Equation (4.3) and Equation (4.4) into Equation (4.5), we arrive at:

$$f_{out}(v_{ti}) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{in}(v_{tj}) \cdot \mathbf{w}(l_{ti}(v_{tj})). \quad (4.6)$$

This formulation can resemble the standard 2D convolution if we treat an image as a regular 2D grid. For example, to resemble a 3×3 convolution operation, we have a neighborhood of 10 pixels in the 3×3 grid centered on a pixel. The neighbor set should be partitioned into 10 subsets, each with one pixel.

Spatial Temporal Modeling Having formulated the CNN spatial graph, we now proceed to model the spatial-temporal dynamics within the skeleton sequence. Recall that in the construction of the graph, the temporal aspect of the graph is constructed by connecting the same joints across consecutive frames. This enables us to define a straightforward strategy to extend the spatial graph CNN to the spatial-temporal domain. That is, we extend the concept of neighborhood to also include temporally connected joints as

$$B(v_{ti}) = \{v_{qj} \mid d(v_{tj}, v_{ti}) \leq K, |q - t| \leq \lfloor \Gamma/2 \rfloor\}. \quad (4.7)$$

The parameter Γ controls the temporal range to be included in the neighbor graph and can thus be called the temporal kernel size. To complete the convolution operation on the spatial-temporal graph, we also need the sampling function, which is the same as in the spatial-only case, and the weight function, mainly the labeling map l_{ST} . Because the temporal axis is well-ordered, we directly modify the label map l_{ST} for a spatial-temporal neighborhood rooted at v_{ti} to be

$$l_{ST}(v_{qj}) = l_{ti}(v_{tj}) + (q - t + \lfloor \Gamma/2 \rfloor) \times K, \quad (4.8)$$

where $l_{ti}(v_{tj})$ is the label map for the single frame case at v_{ti} . In this way, we have a well-defined convolution operation on the constructed spatial-temporal graphs.

Spatial Configuration Partitioning Since the body skeleton is spatially localized, we can utilize this spatial configuration in partitioning. We design a strategy to divide the neighbor set into three subsets: 1) the root node itself; 2) the centripetal group: the neighboring nodes that are closer to the gravity center of the skeleton than the root node; 3) the centrifugal group: the remaining nodes. Here, the average coordinate of all joints in the skeleton at a frame is treated as its gravity center. This strategy is inspired by the fact that the motions of body parts can be broadly categorized as concentric and eccentric motions. Formally, we have

$$l_{ti}(v_{tj}) = \begin{cases} 0 & \text{if } r_j = r_i \\ 1 & \text{if } r_j < r_i \\ 2 & \text{if } r_j > r_i \end{cases} \quad (4.9)$$

where r_i is the average distance from gravity center to joint i over all frames in the training set.

Learnable Edge Importance Weighting Although joints move in groups when people move, one joint could appear in multiple parts of the body. However, these appearances should have different importance in modeling the dynamics of these parts. In this sense, we add a learnable mask $\mathbf{M}$ to every spatial-temporal graph convolution layer. The mask will scale the contribution of the feature of a node to its neighboring nodes based on the weight of learning importance of each edge of the spatial graph in $\mathcal{E}_S$.

4.4.3. Implementing ST-GCN

The implementation of graph-based convolution is more complex than 2D or 3D convolution. Here, we provide details on the implementation of ST-GCN for skeleton-based action recognition.

We adopt a similar implementation of graph convolution as in [83]. The intrabody connections of joints within a single frame are represented by an adjacency matrix $\mathbf{A}$ and an identity matrix $\mathbf{I}$ representing self-connections. In the single frame case,

ST-GCN with the first partitioning strategy can be implemented with the following formula [83]

$$\mathbf{f}_{out} = \Lambda^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\Lambda^{-\frac{1}{2}}\mathbf{f}_{in}\mathbf{W}, \quad (4.10)$$

where $\Lambda^{ii} = \sum_j (A^{ij} + I^{ij})$. Here, the weight vectors of multiple output channels are stacked to form the weight matrix $\mathbf{W}$. In practice, we can represent the input feature map as a tensor of (C, V, T) dimensions in the spatial-temporal cases. The graph convolution is implemented by performing a standard 2D convolution $1 \times \Gamma$ and multiplies the resulting tensor with the normalized adjacency matrix $\Lambda^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\Lambda^{-\frac{1}{2}}$ in the second dimension.

The Equation (4.10) is transformed into

$$\mathbf{f}_{out} = \sum_j \Lambda_j^{-\frac{1}{2}} \mathbf{A}_j \Lambda_j^{-\frac{1}{2}} \mathbf{f}_{in} \mathbf{W}_j, \quad (4.11)$$

where similarly $\Lambda_j^{ii} = \sum_k (A_j^{ik}) + \alpha$. Here we set $\alpha = 0.001$ to avoid empty rows in $\mathbf{A}_j$.

It is straightforward to implement the learnable edge importance weighting. For each adjacency matrix, we accompany it with a learnable weight matrix $\mathbf{M}$. And we substitute the matrix $\mathbf{A} + \mathbf{I}$ in Equation (4.10) and $\mathbf{A}_j$ in Equation (4.11) with $(\mathbf{A} + \mathbf{I}) \otimes \mathbf{M}$ and $\mathbf{A}_j \otimes \mathbf{M}$, respectively. Here, $\otimes$ denotes the element-wise product between two matrices. The mask $\mathbf{M}$ is initialized as an all-in-one matrix.

4.5. ResGCN Architecture

The vanilla ST-GCN architecture, presented in the previous section, has been the foundation for many subsequent graph-based models in action recognition. However, the original ST-GCN architecture has limitations in capturing complex spatiotemporal dependencies due to the shallow network depth and the absence of residual connections. To address these limitations, Song *et al.* [141] proposed the Residual Graph Convolutional Network

(ResGCN), which integrates residual connections into GCNs to enable the construction of deeper networks and improve the propagation of features across layers. In this section, we will dive into the ResGCN architecture and its key components.

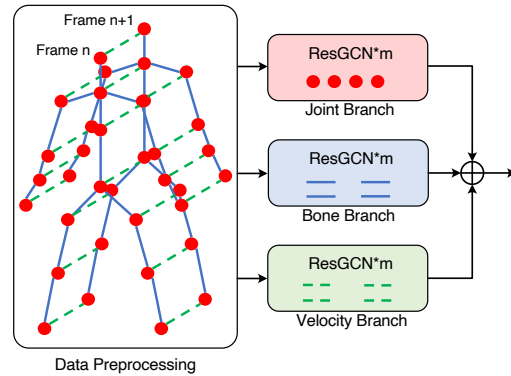


Figure 4.7.: Visualization of the multi-branch input architecture.

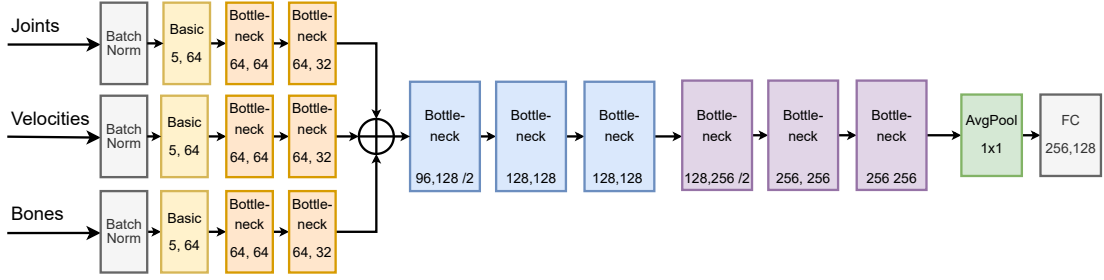


Figure 4.8.: **Extended ResGCN Architecture.** ResGCN architecture with multiple residual connections in the ST-GCN blocks and a bottleneck for feature dimension reduction.

4.5.1. Multi-Branch Input

Recent advances in skeleton-based action recognition have emphasized the importance of using multiple input modalities to capture the complex dynamics of human actions. Two notable works in this area [173, 141] have proposed incorporating skeleton joints, velocity, and bone inputs to improve recognition performance. Figure 4.7 visualizes the precomputed inputs for the multi-branch input architecture.

Skeleton Input The skeleton input consists of the 3D/2D coordinates of key joints in the human body, representing the spatial configuration of the body at each time frame. This input forms the primary data structure for modeling human poses and movements. In addition, we provide confidence for each joint, which is a common practice in pose estimation to indicate the reliability of the estimated joint position. The input of the skeleton is represented as a tensor $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$, where T is the number of frames, N is the number of joints and C is the number of features per joint. The skeleton input is used as the primary input for the network. As the coordinates are given in an absolute system, we also provide the coordinates relative to a central joint, which is the neck joint in our case. This relative representation is used to capture the relative positions of the joints in the body:

$$\mathcal{R} = \{v_{t,n} - v_{t,center} \mid t, n \in \mathbb{N}_0, t < T, n < N\}. \quad (4.12)$$

Bone Input The bone input captures the structural information of the human body by computing the lengths and angles of bones between connected joints. The bone input is calculated from the skeleton input and is represented as two separate tensors: bone lengths $\mathcal{L}$ and bone angles $\mathcal{A}$.

For the bone length, we subtract the coordinate of every joint n with every connected joint n_{adj} :

$$\mathcal{L} = \{v_{t,n} - v_{t,n_{adj}} \mid t, n \in \mathbb{N}_0, t < T, n < N\}. \quad (4.13)$$

Finally, the angle of each bone is:

$$\mathcal{A} = \left\{ \arccos \left(\frac{v_{t,n} - v_{t,n_{adj}}}{\sqrt{\sum v_{t,n}^2}} \right) \mid t, n \in \mathbb{N}_0, t < T, n < N \right\}. \quad (4.14)$$

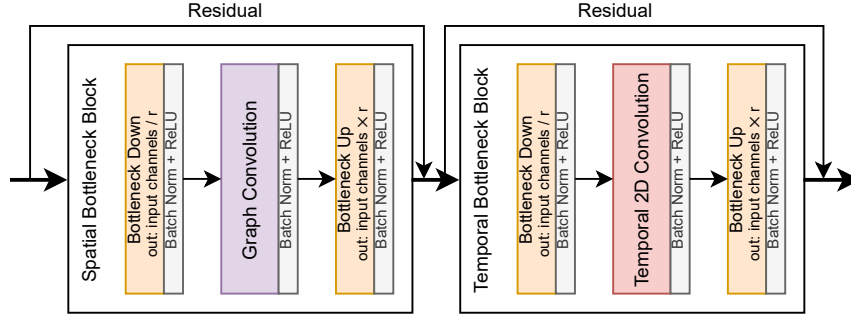


Figure 4.9.: **ResGCN Block**. Each block contains a spatial and temporal graph convolution. The bottleneck structure reduces the feature dimensionality before each of those convolutions. The residual connections are added around each part of the block.

Velocity Input The velocity input captures the temporal dynamics by computing the differences in joint positions between consecutive frames. This input effectively represents the speed and direction of the joint movements. The input of velocity $\mathcal{F}$ is calculated as the difference between the joint positions in the current frame and the previous frame:

$$\mathcal{F} = \{v_{t+i,n} - v_{t,n} \mid t, n \in \mathbb{N}_0, i \in \{1, 2\}, t < T - 2, n < N\}. \quad (4.15)$$

4.5.2. ResGCN Block

The ResGCN block with a bottleneck structure is meticulously designed to enhance not only the representative capacity of the classical ST-GCN architecture but also to ensure computational efficiency, a crucial factor in real-world applications. Like the ST-GCN block, the ResGCN block consists of two graph convolutional layers: a spatial graph convolution and a temporal graph convolution, as introduced in the previous Section 4.4. Although proven effective, this structure can lead to high computational costs and may suffer from gradient vanishing as the network depth increases. ResGCN block introduces residual connections and a bottleneck structure, as shown in Figure 4.9 to improve the propagation of characteristics and reduce the number of parameters.

Residual Connections Inspired by the ResNet architecture in traditional CNNs, ResGCN incorporates skip connections that add the input of a block directly to its output. This design allow to train deeper models more efficiently [61].

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \mathcal{W}) + \mathbf{x}, \quad (4.16)$$

where $\mathbf{x}$ is the input to the block, $\mathcal{F}(\mathbf{x}, \mathcal{W})$ is the function representing the operations within the block (e.g., graph and temporal convolutions), and $\mathbf{y}$ is the output.

	Controlled		In the Wild			
	CASIA-B [169] (2006)	OUMVLP [2] (2020)	GREW [183] (2021)		Gait3D [176] (2022)	
# IDs	124	10,307	26,345		4,000	
# Sequences	13,640	288,596	128,671		25,309	
Split (Train/Test) [%]	60/40	50/50	77/23		75/25	
# Cameras	11	14	882		39	
Conditions	NM, BG, CL	NM	Diverse		Diverse	
Keypoint Dimensions	2D	2D	2D	3D	2D	3D
Keypoint Estimator	HRNet	OpenPose	HRNet	Chen <i>et al.</i>	HRNet	ROMP
Estimator Performance	75.8 mAP	64.2 mAP	75.8 mAP	-	75.8 mAP	-
# Keypoints	17	18	17	14	17	24

Table 4.2.: **Gait Datasets Comparison.** Comparison of Gait Datasets with their Keypoint properties. 2D Keypoint accuracy of the pose estimator is reported on the COCO test-dev dataset as mAP $\uparrow$.

Bottleneck Structure The bottleneck structure is introduced to reduce the computational burden while maintaining high model expressiveness. This is achieved by compressing the feature dimension before the heavy computational operations (spatial and temporal convolutions) and then expanding it back. Typically, a 1x1 convolution is employed for both the compression and expansion stages. The compression stage reduces the dimensionality of the input features, reducing the computational cost. After the primary operations, the expansion stage restores the original or desired feature dimension. This approach significantly reduces the number of parameters and the amount of computation needed without sacrificing the network’s capacity to learn complex patterns.

In general, the ResGCN block with a bottleneck structure represents a powerful extension of the classical ST-GCN framework, combining the strengths of residual learning and efficient network design to allow deeper, more robust, and computationally feasible models.

4.6. Experiments

4.6.1. Datasets

In recent years, gait recognition research has predominantly focused on silhouette-based approaches. Consequently, the majority of available datasets have provided only silhouettes. The widely utilized CASIA-B dataset [169] offers RGB images, enabling the application of pose estimation techniques. Recently, an extension with pose data, OUMVLP-Pose [2], was published for the OU-MVLP dataset [146]. Following the saturation of algorithm performance in controlled laboratory environments, two novel datasets [183, 176] have introduced a new challenge by bringing gait recognition *in situ* on a large scale. Table 4.2 presents a comparative analysis of these four datasets.

CASIA-B [169] is a prominent multiview gait dataset comprising 124 subjects recorded under laboratory conditions. The dataset encompasses three walking conditions captured from 11 viewing angles (0° , 18° , ..., 180°). These conditions include normal walking (NM) (6 sequences per subject), walking with a bag (BG) (2 sequences per subject), and wearing a jacket or coat (CL) (2 sequences per subject). In total, each subject is represented by $11 \times (6 + 2 + 2) = 110$ sequences. While CASIA-B lacks an official training or test set partitioning, several experimental protocols have been proposed [175]. In this study, we employ the widely adopted protocol proposed by [161] for a fair comparison. It excludes same-view recognition angles. Furthermore, we utilize the commonly referred to as large-sample training (LT) partition. This partition allocates the first 74 subjects to the training set, with the remaining 50 subjects constituting the test set. Within the test set, the gallery comprises four sequences from the NM condition (NM #1-4), while the remaining six sequences are divided into three probe subsets: NM subsets containing NM #5-6, BG subsets containing BG #1-2, and CL subsets containing CL #1-2. We extract poses from CASIA-B using the pre-trained HRNet [143] pose estimator.

OU-MVLP (Osaka University - Multi-View Large Population) [146] is an extensive multi-view gait dataset recorded in a laboratory environment. It represents the largest gait dataset in terms of sequences, facilitating comprehensive cross-view evaluation of algorithms. OU-MVLP encompasses 10,307 subjects captured by seven cameras in a fixed round-trip walking course, effectively yielding 14 views at 15° intervals (0° , 15° , ..., 90° ; 180° , 195° , ..., 270°). Each sequence contains between 18 and 35 frames, with an average of 25 frames. The dataset is partitioned into 5,153 subjects for training and 5,154 subjects for testing. During testing, sequences with index #01 constitute the gallery, while the remaining sequences form the probe set.

Subsequent to the original release featuring silhouette images, extensions incorporating pose and mesh data were introduced. OUMVLP-Pose [2] provides keypoints extracted from the (unreleased) RGB frames. Two datasets were generated using different pre-trained pose estimators, OpenPose [17] and AlphaPose [43], while maintaining consistency in frames and subjects. OUMVLP-Mesh [89] contains 3D human meshes extracted from RGB frames using the SMPL model [103]. In aggregate, the dataset offers the following modalities: silhouettes, 2D/3D skeletons, and 3D meshes.

GREW (Gait REcognition in the Wild) [183] was developed to provide a challenging dataset by transitioning gait recognition from controlled laboratory settings to *in situ* environments. While previous gait recognition datasets often originated from controlled environments with limited variability, the GREW dataset is collected from diverse and uncontrolled environments, more accurately representing real-world scenarios. Data for the GREW dataset were acquired using a network of surveillance cameras installed in public spaces, as illustrated in Figure 4.10c. The dataset aims to advance gait recognition technology by presenting challenges such as varying viewpoints, occlusions, clothing changes, and diverse carried items. It incorporates rich annotations including age, gender, carried items, and clothing types. The dataset

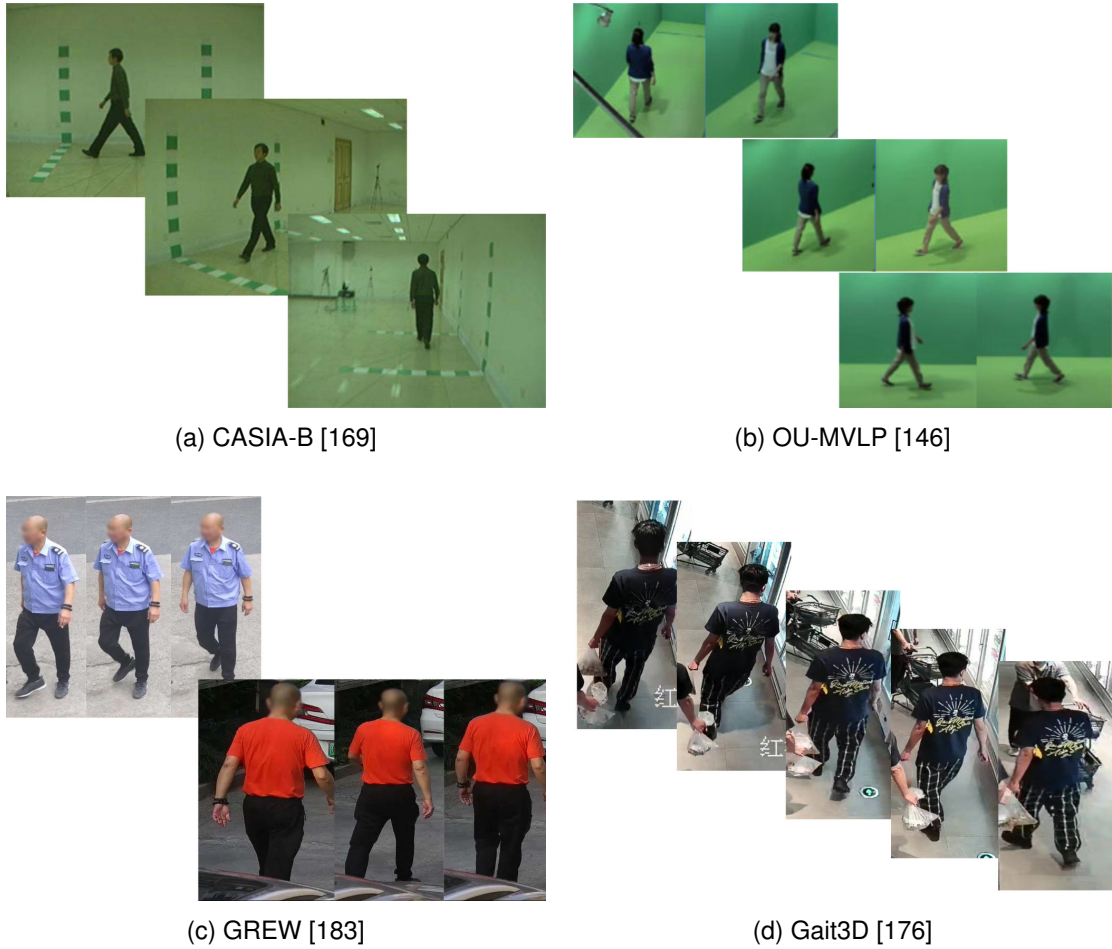


Figure 4.10.: **Example Image from Gait Datasets.** Example images from the four datasets used in this work. CASIA-B and OUMVLP are recorded in lab conditions in front of a controlled environment, and GREW and Gait3D are recorded in the wild.

also includes a distractor set comprising sequences from subjects not present in the gallery or probe set. For privacy reasons, the RGB images are not released; only silhouettes, poses, and optical flow images are provided.

The dataset includes nearly 3,500 hours of footage from 882 cameras. GREW is partitioned into training, validation, and testing sets, containing 20,000, 345, and 6,000 subjects, respectively. During testing, two sequences per subject are designated as probes, while the remaining two sequences serve as the gallery.

Gait3D [176] is a dataset acquired in an uncontrolled environment, focusing on 3D gait representations. The data were collected from 39 cameras in a supermarket setting, encompassing 4,000 subjects across more than 25,000 sequences. While smaller than GREW, the supermarket environment presents unique challenges with a greater variety of carried items and viewing angles. The dataset is partitioned into 3,000 subjects for training and 1,000 for testing. During evaluation, one sequence per subject is designated as the probe, with the remaining sequences forming the gallery.

Gait3D incorporates all contemporary modalities: 3D meshes, 2D/3D skeletons, and silhouettes. Emphasizing 3D representations, the authors employed the state-of-the-art 3D pose estimator ROMP [145] to extract 3D poses from RGB images.

Among the available datasets for gait recognition, GREW stands out as the most comprehensive, offering a large-scale dataset with diverse and challenging scenarios. It is designed to advance gait recognition technology by presenting complex situations including varying viewpoints, occlusions, clothing variations, and diverse carried items. Notably, GREW includes a distractor set comprising sequences from subjects absent from both the gallery and probe sets. It is the sole dataset featuring a private test set and a public leaderboard, facilitating fair comparisons among different approaches.

4.6.2. Evaluation Metrics

The evaluation protocol for gait recognition adheres to open-set instance retrieval paradigms, similar to person re-ID [4[†]]. The gallery encompasses all subjects, while the probe consists of a subset thereof. For each query sequence, we quantify its similarity to all gallery sequences using the cosine similarity of their respective feature vectors. We report the Rank-1 and/or Rank- N accuracy, defined as the percentage of correctly identified subjects within the top N retrieved sequences. For Gait3D, we additionally adopt mean Average Precision (mAP) and mean Inverse Negative Penalty (mINP) [166], which account for the recall of multiple instances and challenging samples.

4.6.3. Implementation Details

Our training setup utilizes an Adam optimizer [82] with a *1-cycle* learning rate schedule [137], employing a maximum learning rate of 0.005. The learning rate increases linearly to the maximum value in the initial 10% of epochs, subsequently decreasing linearly to 0 in the final 10% of epochs. At 80% of the maximum epochs, the learning rate is reduced by a factor of 0.1. All experiments were conducted on a single NVIDIA 3090 GPU using PyTorch. The embedding size is fixed at 128 dimensions. To accommodate the varying properties and sizes of the datasets, we employ different model architectures and training strategies.

Loss The loss function employed in our approach combines cross-entropy loss with supervised contrastive (*SupCon*) loss, as proposed by [81]. The loss is computed using the following equation:

$$L_{SupCon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)} \quad (4.17)$$

where I denotes the set of indices for all samples, $P(i)$ represents the set of indices of positives (same class) for sample i , $A(i)$ is the set of indices of all samples in the batch except for sample i , and z_i is the normalized embedding. The temperature

parameter τ is set to 0.01. The SupCon loss can be conceptualized as a cross-entropy between the ground-truth distribution and the contrastive distribution. In comparison to traditional contrastive losses, such as triplet or N-pair losses, this loss function considers all positive and negative samples within the batch. While this loss was state-of-the-art at the time of implementation, it was subsequently surpassed by an All-Batch Triplet loss [63], which combined the advantages of an all-batch loss with the properties of the triplet loss. The combination of this loss with cross-entropy loss has become the most prevalent approach in gait recognition, as evidenced in Table 4.1.

Sampling Strategy The compact nature of skeleton data enables the utilization of large batch sizes. For smaller controlled datasets, we employ random sampling with a batch size of 768. However, for larger datasets, random sampling does not guarantee the inclusion of another positive example in each batch. Consequently, we implement a triplet sampling strategy, constructing batches of size 256, comprising $P = 128$ persons with $K = 2$ images each.

Augmentation Our network architecture is designed to accommodate a constant input sequence length. To address this, we first apply mirror-padding to sequences shorter than the desired length in the temporal dimension. Subsequently, we randomly select a subsequence of the desired length. Additionally, we implement random left-right flipping of images, accompanied by the corresponding flipping of left and right joints. To account for potential inaccuracies in pose estimation, we introduce uniform noise to the estimated keypoints and their associated confidence values.

Evaluation & Test Time Augmentation (TTA) During the testing phase, we employ the cosine distance metric to measure the similarity between gallery and probe characteristic vectors. We apply the same sequence padding augmentation used during training, but select a sub-sequence of the required length from the center of the sequence. Furthermore, we generate two additional samples: a left-/right-flipped sample and a time-inverted sample. The resulting three embeddings are concatenated for subsequent distance calculation.

4.6.4. Main Results

This section presents a comparative analysis of our results with state-of-the-art methods across four datasets. We evaluate performance on both controlled datasets (CASIA-B and OUMVLP) and in-the-wild datasets (GREW and Gait3D). Initially, we published results exclusively for CASIA-B and OUMVLP; however, this work now includes results for the in-the-wild datasets as well. In the latter part of this section, we conduct ablation studies to analyze the various components of our approach and their impact on overall performance.

Gallery NM#1-4		0°-180°											
Probe		0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	mean
NM#5-6	GaitNet [139]	91.2	92.0	90.5	95.6	86.9	92.6	93.5	96.0	90.9	88.8	89.0	91.6
	GaitSet [20]	90.8	97.9	99.4	96.9	93.6	91.7	95.0	97.8	98.9	96.8	85.8	95.0
	GaitPart [41]	94.1	98.6	99.3	98.5	94.0	92.3	95.9	98.4	99.2	97.8	90.4	96.2
	3DLocal [73]	96.0	99.0	99.5	98.9	97.1	94.2	96.3	99.0	98.8	98.5	95.2	97.5
	PoseGait [95]	55.3	69.6	73.9	75.0	68.0	68.2	71.1	72.9	76.1	70.4	55.4	68.7
	GaitGraph [9 [†]]	85.3	88.5	91.0	92.5	87.2	86.5	88.4	89.2	87.9	85.9	81.9	87.7
	GaitGraph2 [8 [†]]	78.5	82.9	85.8	85.6	83.1	81.5	84.3	83.2	84.2	81.6	71.8	82.0
	Gait-TR [171]	95.7	96.4	97.9	97.0	96.9	95.5	95.1	96.1	96.6	96.0	92.4	96.0
BG#1-2	GaitNet [139]	83.0	87.8	88.3	93.3	82.6	74.8	89.5	91.0	86.1	81.2	85.6	85.7
	GaitSet [20]	83.8	91.2	91.8	88.8	83.3	81.0	84.1	90.0	92.2	94.4	79.0	87.2
	GaitPart [41]	89.1	94.8	96.7	95.1	88.3	84.9	89.0	93.5	96.1	93.8	85.8	91.5
	3DLocal [73]	92.9	95.9	97.8	96.2	93.0	87.8	92.7	96.3	97.9	98.0	88.5	97.5
	PoseGait [95]	35.3	47.2	52.4	46.9	45.5	43.9	46.1	48.1	49.4	43.6	31.1	44.5
	GaitGraph [9 [†]]	75.8	76.7	75.9	76.1	71.4	73.9	78.0	74.7	75.4	75.4	69.2	74.8
	GaitGraph2 [8 [†]]	69.9	75.9	78.1	79.3	71.4	71.7	74.3	76.2	73.2	73.4	61.7	73.2
	Gait-TR [171]	90.9	92.4	91.4	93.2	91.9	90.2	91.4	93.9	93.9	92.7	82.9	91.3
CL#1-2	GaitNet [139]	42.1	58.2	65.1	70.7	68.0	70.6	65.3	69.4	51.5	50.1	36.6	58.9
	GaitSet [20]	61.4	75.4	80.7	77.3	72.1	70.1	71.5	73.5	73.5	68.4	50.0	70.4
	GaitPart [41]	70.7	85.5	86.9	83.3	77.1	72.5	76.9	82.2	83.8	80.2	66.5	78.7
	3DLocal [73]	78.2	90.2	92.0	87.1	83.0	76.8	83.1	86.6	86.8	84.1	70.9	83.7
	PoseGait [95]	24.3	29.7	41.3	38.8	38.2	38.5	41.6	44.9	42.2	33.4	22.5	36.0
	GaitGraph [9 [†]]	69.6	66.1	68.8	67.2	64.5	62.0	69.5	65.6	65.7	66.1	64.3	66.3
	GaitGraph2 [8 [†]]	57.1	61.1	68.9	66	67.8	65.4	68.1	67.2	63.7	63.6	50.4	63.6
	Gait-TR [171]	86.7	88.2	88.4	89.7	91.1	90.7	93.2	93.8	93.2	91.2	83.6	90.0

Table 4.3.: **CASIA-B Results.** Averaged Rank-1 accuracies in percent on CASIA-B per probe angle excluding identical-view cases compared with other model-based methods. Each probe is grouped in silhouette-based (input size 64×44) and model-based methods.

Controlled Datasets The most popular dataset at the time GaitGraph was published was CASIA-B. This enabled our approach because it contains RGB images, and we can extract the poses with a pre-trained pose estimator. The results of CASIA-B are shown in Table 4.3. Meanwhile, CASIA-B lost popularity because of its small size and saturation results. This was also mitigated, as the OUMVLP authors published the OUMVLP-Pose dataset, which contains the same sequences as the original dataset but with the poses extracted from the RGB images. The results on OUMVLP are shown in Table 4.4. In addition to the original results, we added more recent results from the literature to provide a more comprehensive comparison.

Starting with **CASIA-B**, we could originally only compare to PoseGait [95], and GaitGraph outperformed the approach with handcrafted features by a large margin. Indicating that the learned features are more powerful than the handcrafted ones. Both approaches still suffer in performance under different walking conditions. However, the performance drop is less substantial for GaitGraph. This could be due to our approach’s more accurate pose estimator, as the bag and coat conditions are more challenging for the pose estimator. This is also supported by the results reported by Gait-TR [171], which uses a more accurate pose estimator (SimCC [91]) and suffers less performance loss under different walking conditions.

Probe Angle	0°	15°	30°	45°	60°	75°	90°	180°	195°	210°	225°	240°	255°	270°	mean
Silhouette	GEINet	11.4	29.1	41.5	45.5	39.5	41.8	38.9	14.9	33.1	43.2	45.6	39.4	40.5	35.8
	GaitSet	79.5	87.9	89.9	90.2	88.1	88.7	87.8	81.7	86.7	89.0	89.3	87.2	87.8	87.1
	GaitPart	82.6	88.9	90.8	91.0	89.7	89.9	89.5	85.2	88.1	90.0	90.1	89.0	89.1	88.7
	3DLocal	86.1	91.2	92.6	92.9	92.2	91.3	91.1	86.9	90.8	92.2	92.3	91.3	91.1	90.2
OP	CNN-Pose	8.2	13.9	18.1	22.4	21.3	18.2	10.9	7.3	13.5	12.0	20.5	17.3	13.7	14.8
	GaitGraph2	32.9	47.7	53.9	56.8	53.9	54.7	45.4	29.0	35.7	34.3	44.3	46.2	46.4	44.3
AlphaPose	CNN-Pose	14.3	22.3	27.2	30.0	28.4	23.4	17.2	7.9	13.6	15.6	25.0	24.1	20.2	20.4
	GaitGraph	42.2	54.5	58.8	60.5	59.3	59.5	56.3	39.6	45.0	43.0	55.5	54.1	53.2	52.2
	GaitGraph2	54.3	68.4	76.1	76.8	71.5	75.0	70.1	52.2	60.6	57.8	73.2	67.8	70.8	65.3
	Gait-TR	30.3	44.2	49.0	52.4	51.4	46.0	36.7	26.9	33.8	32.0	46.7	46.0	40.8	40.8
	GPGait	47.7	62.4	68.3	69.8	68.9	65.9	62.0	42.8	54.3	53.2	67.0	66.3	63.3	60.7

Table 4.4.: **OU-MVLP Results.** Averaged rank-1 accuracies on OUMVLP per probe angle. The results are grouped into silhouette-based methods (input size 64×44) and skeleton-based methods with OpenPose and AlphaPose pose estimation.

All approaches are more robust in the angles of the askew than in the angles of the strict side view, front view, or back view (0° , 90° , 180°), independent of the modality and model. But also here, Gait-TR stands out with the best performance from all angles. Possibly again, because of the better pose estimator.

Our later published approach, GaitGraph2, can not outperform GaitGraph here, which we observed during training due to the overfitting of the larger model. Later, Gait-TR builds on top of GaitGraph2 and improves performance with a stronger transformer-based backbone and a better pose estimator. Gait-TR even challenges the state-of-the-art silhouette-based methods, 3DLocal [73], and outperforms it under the coat (CL) condition. This indicates that our hope that model-based and especially skeleton-based approaches can be more invariant to the walking conditions is true but is still dependent on the pose estimator’s quality.

We can compare our results on the larger controlled dataset **OUMVLP**. Pose data was released for this dataset with a simple baseline CNN-Pose [2]. The results show that both our methods outperform the CNN-based baseline considerably. We can improve performance by $\sim 3\times$ in both keypoint types. Again, we can compare our approach to Gait-TR, and it shows that the approach does not outperform our method, showing that the better performance on CASIA-B is mostly due to the better pose estimator. Interestingly, our method also outperforms the largest model (*cf.* Figure 4.11) in this dataset and still is state-of-the-art for skeleton-based approaches on this dataset. Compared to silhouette-based approaches, our approach still needs to catch up to 3DLocal [73], and the gap is not as close as on CASIA-B.

Although the research area has moved towards more challenging datasets, we can show that our approach is still state-of-the-art on the controlled datasets. Our approach has led to a new way of gait recognition, and we are still competitive with the latest approaches.

Method	GREW				Gait3D			
	Rank-1	Rank-5	Rank-10	Rank-20	Rank-1	Rank-5	mAP	mINP
GEINet [135]	6.8	13.4	17.0	21.0	7.0	16.3	6.1	3.8
TS-CNN [161]	13.6	24.6	30.2	37.0	-	-	-	-
GaitSet [20]	46.3	63.6	70.3	76.8	42.6	63.1	33.7	19.7
GaitPart [41]	44.0	60.7	67.3	73.5	29.9	50.6	23.3	13.2
GaitGL [97]	50.8	66.6	72.2	76.8	23.5	38.5	16.4	9.2
CSTL [72]	48.5	64.1	69.9	75.4	12.2	21.7	6.4	3.3
GaitBase [38]	75.2	85.2	88.2	90.2	45.2	65.2	35.2	20.2
GaitGraph [9 [†]]	36.1	57.2	65.4	72.0	14.2	30.2	12.9	8.0
GaitGraph2 [8 [†]]	44.3	59.1	64.6	69.0	15.2	29.2	11.9	6.8
Gait-TR [171]	48.6	65.5	71.6	76.6	7.2	15.5	5.4	2.9
GPGait	51.0	68.4	74.7	79.8	20.9	37.3	16.1	10.1
SMPLGait [176]	-	-	-	-	53.2	71.0	42.4	26.0
SkeletonGait [40]	77.4	87.9	91.0	93.2	38.1	56.7	28.9	16.1
SkeletonGait++ [40]	85.8	92.6	94.3	95.5	77.6	89.4	70.3	42.6

Table 4.5.: **GREW & Gait3D Results.** Comparison of the state-of-the-art gait recognition methods on two *in the wild* datasets GREW and Gait3D. The results are grouped into silhouette-based (input size 88×128), skeleton-based, and multi-modal methods.

In the Wild Datasets Moving to the more challenging datasets **GREW** [183] and **Gait3D** [176], we can compare our approach in Table 4.5 with silhouette-based, skeleton-based and multimodal gait methods. These results are not part of our original publications but were later compared by other authors [107, 48, 176]. For this work, we reevaluated all skeleton-based approaches on these datasets using the OpenGait [39] framework for a comprehensive comparison. All other results are as reported in the original papers.

The results show that our approaches compare pretty well with later published work [171, 48] and still set the baseline for skeleton-based gait recognition. The results show that the performance on the in-the-wild datasets is lower than on the controlled datasets. This is expected as in-the-wild datasets are more challenging with more diverse walking conditions and occlusions.

The results for Gait3D are less conclusive than those for GREW, and the recognition rate is much lower. This could be due to the challenging environment with many carried items and more extreme view angles. The current state-of-the-art on Gait3D is GPGait [48] still shows strong performance on both datasets. The authors focused their approach on finding generalized human descriptors that work well in the wild.

Overall, the results show room for improvement compared to other modalities. However, new research combines two modalities to improve performance by adopting the best of both worlds. We discuss these works in the next paragraph.

Comparison with other Modalities This section presents a comparative analysis of our approach with classical silhouette-based methods and emerging multimodal approaches, showing the progress across various gait representations. Given that

controlled datasets are approaching saturation with silhouette-based methods, we focus our analysis on in-the-wild datasets. The results are summarized in Table 4.5.

We begin by examining the GEI representation, which we introduced earlier in this chapter. The initial two results, GEINet [135] and TS-CNN [161], establish a robust baseline for this representation. These findings demonstrate that while this representation and architecture perform adequately in laboratory conditions, they lack robustness to variations in view angles due to the requirement for well-aligned silhouettes.

Conversely, more recent approaches utilizing silhouette sequences [20, 41, 97, 72] exhibit superior performance on in-the-wild datasets. This improvement can be attributed to advancements in instance segmentation methods [25], which facilitate the extraction of clean silhouettes. However, these approaches were primarily designed for controlled datasets. GaitBase [38], specifically developed for large-scale in-the-wild datasets, employs a substantially larger backbone in conjunction with temporal and spatial pooling methods introduced by earlier approaches. It currently represents the state-of-the-art for single-modality approaches on in-the-wild datasets, underscoring the continued potential of silhouette-based methods.

Our skeleton-based methods demonstrate competitive performance with older silhouette-based approaches, despite their initial lag in controlled environments. However, a substantial performance gap remains between our methods and the most advanced silhouette-based techniques. Nevertheless, the emergence of novel model-based representations and architectures may alter this landscape in the future.

The final category comprises multimodal approaches that integrate two or more gait representations. SMPLGait [176] combines 3D meshes and 2D skeletons to surpass silhouette-only approaches. The SMPL [103] meshes extracted using ROMP [145] exhibit particular robustness to challenging conditions, especially in the Gait3D dataset.

SkeletonGait [40] transforms 2D skeletons into normalized keypoint heatmaps, which are subsequently processed by a CNN. The strong performance of this approach suggests that GCNs may not be the optimal architecture for capturing gait features. These gait maps bear a resemblance to silhouettes with enhanced segmentation. Furthermore, the results of SkeletonGait++, which integrates keypoint heatmaps with silhouettes, demonstrate that combining multiple modalities can yield performance improvements.

In conclusion, the field has yet to converge on an optimal gait representation, with various approaches exhibiting distinct strengths and weaknesses. The integration of multiple modalities emerges as the most promising avenue for performance enhancement. However, it is important to note that combining different modalities increases the computational cost of feature extraction and model complexity.

Model	Params	Rank-1
ST-GCN (Baseline)	3.3 M	60.7
ResGCN-N51	645 K	63.7
+ Multi Branch Input	765 K	66.5
+ Training Augmentation	"	66.6
+ Test Time Augmentation	"	67.1

Table 4.6.: **Model Ablation.** Comparison on a baseline with a vanilla ST-GCN backbone, with a stronger backbone, MBI, and training and test time augmentation. All experiments are conducted on the OUMVLP dataset.

4.6.5. Ablation Studies

Model Components To clarify the impact of various components on our approach’s overall performance, we conducted comprehensive ablation studies. As illustrated in Table 4.6, we started our analysis with the vanilla ST-GCN architecture as our baseline. The results demonstrate that the ResGCN extension significantly outperforms the ST-GCN architecture, achieving superior results with a reduced parameter count, thus exhibiting greater efficiency and efficacy for gait recognition. Subsequently, we incorporated the Multi-Branch Input (*cf.* Section 4.5.1), which enhanced performance through the addition of precomputed bone and velocity inputs. This observation aligns with other approaches that identify these additional inputs as crucial for gait recognition [48]. The subsequent two rows focus on augmentation techniques. At test time, the application of slight point noise and flipped poses yields a minimal performance improvement. However, when combined with TTA, we observe a significant 1.5% increase in performance on OUMVLP. This finding underscores the importance of augmentation techniques in enhancing model performance.

Temporal Modeling To investigate the network’s capacity for temporal information modeling, we conducted experiments utilizing sorted and shuffled sequences during training and testing. Table 4.7 presents three distinct configurations. For this ablation study, we excluded Multi-Branch inputs, as they contain precomputed temporal information that could compromise the validity of the analysis.

Our approach demonstrates a robust ability to model temporal features. We can observe a substantial performance degradation when training with sorted sequences and testing with shuffled sequences (row c). These results further corroborate our assertion of reintroducing genuine temporal features to gait recognition. Additionally, Table 4.7 illustrates the spatial modeling capabilities in row a. Despite the absence of temporal and appearance information, the network successfully learns appearance-invariant features of the individual’s underlying physique.

For comparative purposes, we note that GaitPart [41] conducted an identical ablation study, achieving NM case results of 95.6, 96.2, and 92.5 for configurations a, b, and c, respectively. This comparison suggests that GaitPart relies heavily on appearance

			CASIA-B			OU-MVLP
	Train	Test	NM	BG	CL	AlphaPose
a	Shuffle	Sort	34.7	27.0	19.0	34.1
b	Sort	Sort	72.8	60.1	44.6	63.1
c	Sort	Shuffle	34.3	27.9	18.5	14.5

Table 4.7.: **Temporal Ablation.** Control Condition: shuffle/sort the input sequence at train/test phase. All experiments conducted without MBI and TTA.

Gait3D				
	Rank-1	Rank-5	mAP	mINP
2D+	15.2	29.2	11.9	6.8
2D	7.3	16.3	5.8	3.5
3D	4.9	12.3	4.3	2.8

Table 4.8.: **Keypoint Dimensions.** Ablation on the pose input dimension on the performance of GaitGraph2. The inputs are 2D poses estimated with HRNet either with (2D+) or without (2D) keypoint confidences, and 3D poses estimated with ROMP.

information and exhibits inferior temporal modeling capabilities compared to our approach.

2D+ vs. 3D Keypoints All skeleton-based approaches presented in Table 4.5 utilize 2D poses, with the exception of PoseGait [95], which employs 3D poses. In pursuit of comprehensive 4D human perception, we evaluate our approach’s performance using 3D pose sequences instead of 2D poses in Table 4.8. Additionally, we assess the performance gain attributed to the confidence values of keypoints predicted by the 2D pose estimator. We selected Gait3D for this evaluation, as it provides actual 3D pose estimations, in contrast to GREW’s statistical 3D pose estimation.

The results reveal that our approach heavily relies on keypoint confidence, as evidenced by the 50% reduction in Rank-1 accuracy when this information is omitted. The model requires knowledge of keypoint reliability, i.e., visibility or non-occlusion. For 3D coordinates, there is currently no straightforward method to estimate keypoint confidence. The results indicate that our approach does not benefit from 3D information, which impedes progress towards 4D gait recognition. Furthermore, existing research suggests that 2D pose information already provides sufficient data to infer 3D pose [24]. Consequently, the 2D pose currently remains the preferred representation for gait recognition.

Pose Estimator Our approach employs a two-stage architecture, starting with pose estimation. No gradients propagate from the loss function to the pose estimation stage, resulting in fixed estimated poses. However, the quality of these poses significantly impacts overall performance. The OUMVLP dataset provides two distinct pose estimators: AlphaPose, a top-down approach, and OpenPose, a bottom-up approach. Both models are trained on the COCO keypoint dataset [100], but exhibit

Source Dataset	Method	Target Dataset						
		CASIA-B				OUMVLP	GREW	Gait3D
		NM	BG	CL	mean			
CASIA-B	GaitGraph [9 [†]]	87.7	74.8	66.3	76.3	0.1	0.5	0.9
	GaitGraph2 [8 [†]]	82.0	73.2	63.6	72.9	0.1	0.5	1.1
	Gait-TR[171]	94.7	89.3	86.7	90.2	0.1	0.6	1.1
	GPGait [48]	93.6	80.2	69.3	81.0	2.8	10.0	8.9
OUMVLP	GaitGraph [9 [†]]	4.9	4.8	3.9	4.5	52.2	0.7	1.5
	GaitGraph2 [8 [†]]	8.8	7.6	5.1	7.2	67.1	0.9	1.4
	Gait-TR[171]	10.1	8.3	5.2	7.8	39.8	1.1	2.6
	GPGait [48]	44.4	32.0	22.4	32.9	59.1	11.1	9.0
GREW	GaitGraph [9 [†]]	10.5	7.7	5.7	8.0	0.2	36.1	4.4
	GaitGraph2 [8 [†]]	8.9	7.2	5.1	7.1	0.2	44.3	8.3
	Gait-TR[171]	7.6	6.4	6.4	6.8	0.1	48.6	7.3
	GPGait [48]	57.9	46.0	24.2	42.7	4.3	51.0	18.5
Gait3D	GaitGraph [9 [†]]	16.5	12.2	8.3	12.3	0.3	3.1	30.2
	GaitGraph2 [8 [†]]	12.3	9.9	5.4	9.2	0.1	2.4	29.2
	Gait-TR[171]	4.5	3.9	4.0	4.1	0.1	4.4	15.5
	GPGait [48]	48.8	40.3	19.4	36.2	2.8	11.0	20.9

Table 4.9.: **Cross- and Single-Domain Performance.** Rank-1 accuracy (%) on four datasets: cross-domain and single-domain performance recent state-of-the-art skeleton-based methods, excluding identical-view case of indoor datasets.

differences in performance: AlphaPose achieves 71.0 mAP, while OpenPose reaches 64.2 mAP on the COCO test-dev dataset.

As observed in Table 4.4, the choice of keypoint estimator substantially affects gait recognition performance. The model trained on poses estimated by AlphaPose significantly outperforms the model trained on OpenPose-estimated poses. This disparity indicates a strong correlation between keypoint estimator quality and model performance. Furthermore, the results suggest that gait recognition performance scales proportionally with keypoint estimator performance, as both approaches demonstrate a $2\times$ performance scaling. This scaling behavior implies that skeleton-based methods have the potential to improve concurrently with advancements in pose estimators, potentially yielding enhanced performance in the future.

An additional observation is that the more robust pose estimator, AlphaPose, exhibits greater resilience to angles with higher occlusions (approximately 0° , 90° , and 270°), resulting in more consistent gait recognition performance across all angles. This finding suggests that the superior pose estimator is more adept at handling occlusions and can provide more reliable pose estimates under challenging conditions.

4.6.6. Cross-Domain Generalization

To conduct a comprehensive comparison, we perform cross-domain experiments on skeleton-based methods to evaluate their generalization capabilities. This involves training the model on a (source) dataset and testing it on other (target) datasets.

Table 4.9 presents the results across four popular gait datasets. An additional challenge in cross-dataset comparison arises from the varying keypoint estimators employed (*cf.* Table 4.2). While 2D keypoint estimators are trained on similar data with comparable performance, 3D keypoint estimators exhibit significant differences.

The results demonstrate that GaitGraph and Gait-TR exhibit poor generalization across datasets. Performance decreases substantially when the model is trained on one dataset and evaluated on another. This indicates that the models overfit the training dataset and struggle to generalize to unseen data. The consistency of these results across all datasets suggests that the models fail to learn general features that are common across different datasets. This is a prevalent issue in gait recognition, as datasets are collected under diverse environmental conditions, making it challenging to learn features that are invariant to these variations.

GPGait is the sole model that exhibits a degree of generalization across datasets. The performance degradation is less pronounced compared to other models, indicating that it can learn more universal features across different datasets. The authors attribute this to their introduction of Human-Oriented Transformation (HOT), which performs a series of operations to enhance the model's robustness to input data variations. Additionally, they employ a Human-Oriented Description (HOD) similar to our multi-branch input.

From a dataset perspective, Table 4.9 reveals that even laboratory datasets do not generalize well among themselves. While the performance drop is less severe compared to in-the-wild datasets, it remains substantial. GREW, being the largest and most diverse dataset, yields the best results in cross-dataset evaluation. This suggests that the model trained on GREW learns more general features that are common across different datasets.

These findings indicate that current skeleton-based methods exhibit poor generalization across datasets. The models demonstrate an excessive bias towards the training dataset and struggle to learn general features that are common across different datasets. Although GPGait shows some level of generalization, there remains significant room for improvement in this area.

4.6.7. Computational Efficiency

The computational efficiency of gait models is crucial for real-world applications; however, it is highly dependent on the gait representation and extraction method. Our method, which relies on pose keypoints, requires a keypoint extractor with varying computational costs depending on the specific method employed. Conversely, silhouette-based methods necessitate a background subtraction technique. While a simple background subtraction may suffice in controlled laboratory settings, more sophisticated methods are required for unconstrained environments. Consequently, comparing the computational efficiency of methods with different gait representations is non-trivial and becomes increasingly complex with multimodal approaches. To

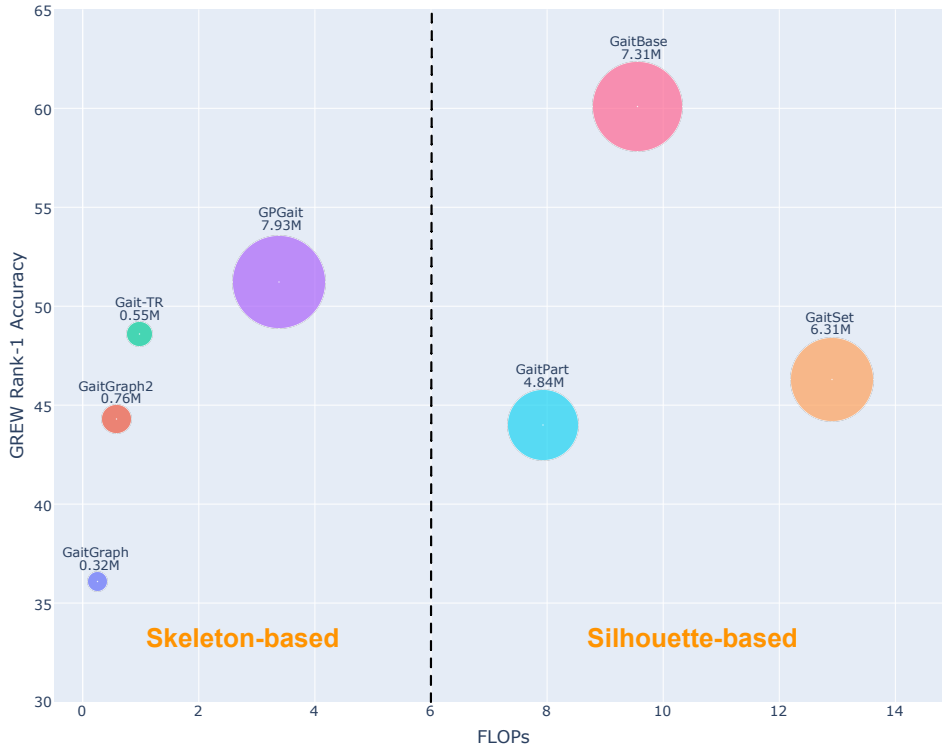


Figure 4.11.: **Analysis of Computational Efficiency.** Comparison of popular gait recognition methods with skeleton- and silhouette-based representations with their rank-1 accuracy on the GREW dataset compared to Floating Point Operations (FLOPs). The size of the bubbles denotes the number of parameters.

provide an indication of our method’s computational efficiency, we present a comparison of the number of parameters and FLOPs of our model with other methods in Figure 4.11.

In the context of skeleton-based methods, our original GaitGraph, utilizing a small ResGCN backbone, exhibits a low parameter count and FLOPs. GaitGraph2 incorporates multi-branch inputs combined with a slightly larger configuration of the ResGCN backend, resulting in increased parameters and FLOPs, but yielding a substantial performance improvement. Gait-TR builds upon GaitGraph2, replacing the backbone with a more robust transformer-based architecture, thus enhancing the Rank-1 accuracy. As anticipated with transformer architectures, they demonstrate a lower parameter count but higher FLOPs compared to graph convolutional architectures. GPGait achieves a significant performance increase, albeit at the cost of a substantial increase in parameters and FLOPs, attributable to the introduction of local and global branches and computationally expensive attention mechanisms.

Notably, our skeleton-based models demonstrate superior efficiency compared to silhouette-based models. The latter receive dense binary image inputs ($T \times 64 \times 44$), while skeleton-based models process sparse keypoint inputs ($T \times 17 \times 3$). This

disparity results in significantly fewer parameters and FLOPs for skeleton-based models, disregarding the feature extractor.

Examining the chronological progression of silhouette-based models, GaitSet pioneered the processing of silhouette images as a sequence using a convolutional architecture. GaitPart enhanced this architecture by focusing on body parts separately and reducing computational costs through late fusion of temporal information. GaitBase [39] aimed to establish a new robust baseline with a simple ResNet-based backbone and temporal pooling, featuring a higher parameter count than specialized models while maintaining competitive performance on larger outdoor datasets.

4.6.8. Gait Recognition Visualization

Understanding and interpretation the predictions of a machine learning models is especially of interest for biometric recognition [7[†]], as it makes deploying them more trustworthy. We attempt to better understand what the network uses from the skeleton data to discriminate between different gaits. Hence, we want to study which joints have the highest activation at different times of the gait cycle. We adopt Class Activation Maps (CAMs) [179] designed for image classification to GCNs that gives a simple and interpretable visualization of the network's attention. The CAM is a heatmap that highlights the regions of the input image most relevant to network prediction. A CAM M_c for a class c is computed as the weighted sum of the feature maps f of the last convolutional layer of the network:

$$M_{c,x,y} = \sum_k w_k^c f_k(x, y) \quad (4.18)$$

For skeleton data, we want to compute the activation of each joint n over the timesteps t and change Equation (4.18) to:

$$M'_{n,t} = \sum_k w_k f_k(n, t) \quad (4.19)$$

Figure 4.12 shows the normalized joint activations $M'(t, n)$ in a skeleton structure. The main observation is that the network looks mostly on the outer limbs. The keypoints on the hands and feet have the highest activation over the gait cycle, and the shoulders and face keypoints (except ears) have the lowest activation. The limb with forward motion has a higher activation than fixed parts in the temporal context. For example, in sequence a), the right leg swing from frame 5 to frame 20 shows increased foot activation the more it moves. These observations conclude that the arm swing and the leg swing are the most discriminated features in our network. The network also takes hints from head movement, as shown in the front view case c). In front view the head keypoint and head movement have strong confidence, thus the network takes motion hints from the head. The most difficult case is the back view (*cf.* Table 4.4) as shown in subfigure c). We do not plot the head keypoints here as

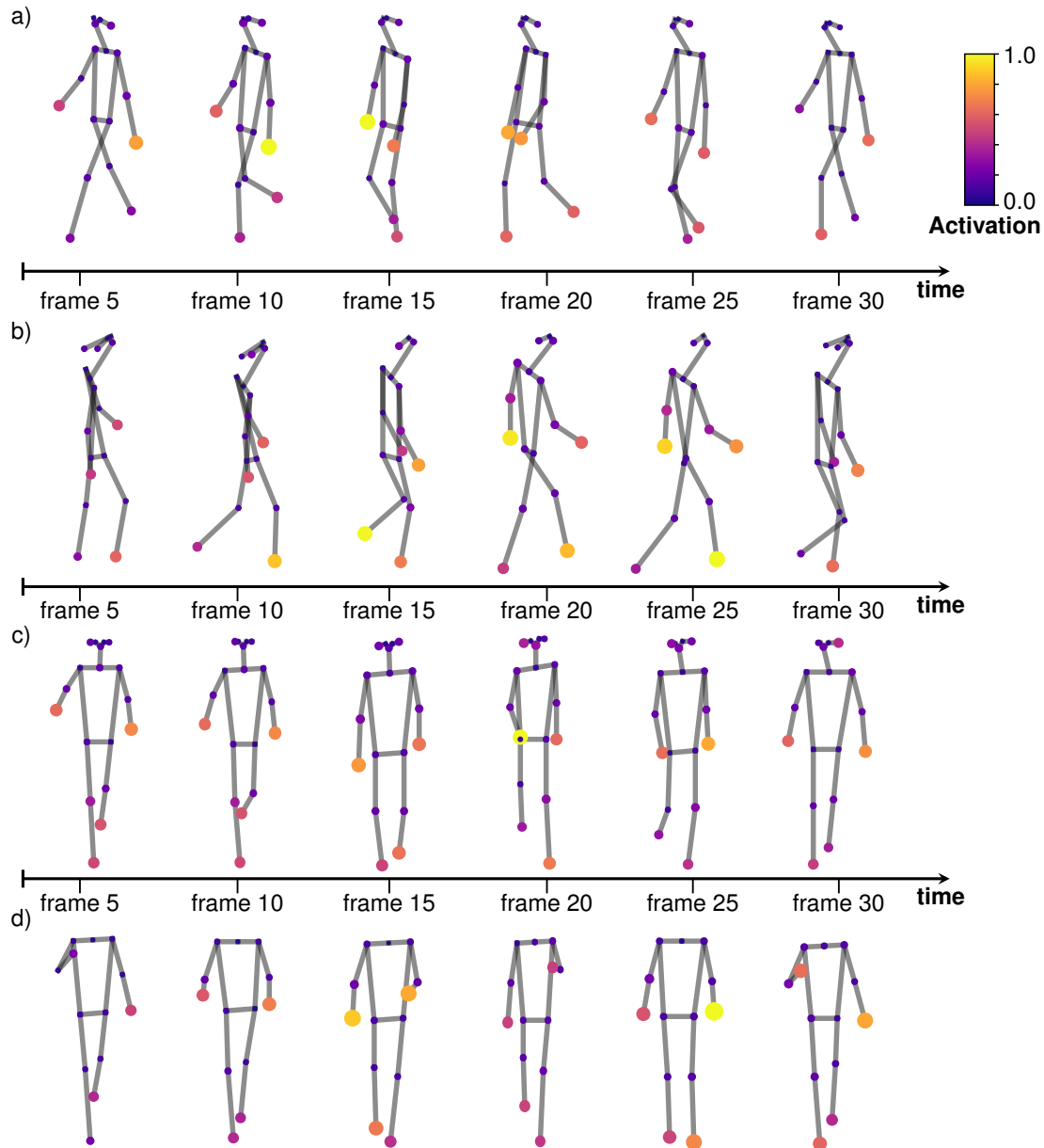


Figure 4.12.: **Gait Activation Analysis.** Activated joints of our proposed method on an example gait cycle from the OUMVLP-Pose test set. Higher activated joints are shown in a brighter color and in bigger scale. The view angles are 60° , 255° , 0° (front view), and 180° (back view) from top to bottom.

they have a confidence of lower than 2.5% and they would be very noisy. Thus, the network focuses on the outer limbs, especially the legs, to discriminate between the gaits. The missing face keypoints explain the lower recognition rate in the back view compared to the front view.

Overall our observations show that gait can be represented by a few keypoints over the temporal context. This highly aligns with the early definition of gait that we introduced earlier:

"The motion of the living body [is] represented by a few bright spots describing the motions of the main joints" – Johansson *et al.* [77]

4.7. Discussion

Ethical Considerations Gait recognition presents two primary ethical concerns: privacy infringement and algorithmic bias. The privacy issue stems from the potential utilization of gait recognition for surveillance and tracking purposes. This invasion of privacy is particularly concerning in comparison to other biometric recognition methods, as gait recognition can be conducted remotely and without explicit consent. Although the skeleton representation is less intrusive than silhouette-based approaches, our research demonstrates that it remains capable of individual identification. Nevertheless, in non-public environments, gait recognition may serve beneficial security and safety applications.

Our work further highlights that skeleton representation does not guarantee complete privacy preservation. In public applications, while the avoidance of RGB image storage in favor of skeleton representations may seem to mitigate privacy concerns, our findings indicate that these representations still enable individual identification. This issue manifested in a practical scenario involving a supermarket chain [125] that employed skeleton representations for customer tracking and analysis. Despite the company's assertion of not storing biometric information, our research demonstrates the potential for identification using these representations. This scenario underscores the complex balance between privacy protection and security enhancement in gait recognition applications.

The issue of bias in gait recognition algorithms presents another significant ethical consideration. Certain demographic groups, such as individuals with disabilities or specific ethnicities, may experience reduced recognition accuracy. This bias can originate from various sources, including the pose estimator, dataset composition, or network architecture. While the utilization of skeleton representations may potentially mitigate algorithmic bias by reducing appearance-based information, bias can persist within the pose estimator or dataset. To address this issue, it is imperative to employ diverse datasets and conduct comprehensive evaluations across various

demographic groups. The current predominance of Asian-centric datasets and evaluations necessitates the collection and integration of data from diverse geographical regions to enhance the generalizability and fairness of gait recognition algorithms.

Limitations The primary limitation of current gait recognition approaches lies in their reliance on perfect tracklets. Existing systems do not account for imperfections introduced by the detection and tracking processes necessary for obtaining tracklets. These processes can introduce errors such as occlusions, misalignment, and missing detections, which may significantly impact the performance of gait recognition systems. In real-world scenarios, gait recognition must be robust to outliers. This issue was already addressed in the previous chapter by enhancing a multi-view recognition and tracking system.

The majority of research has focused on controlled environments; only recent datasets have extended gait recognition to unconstrained settings. Recognition rates in the wild remain substantially lower than those in laboratory settings. Furthermore, generalization to unseen domains continues to pose a significant challenge for gait recognition. Consequently, current gait recognition systems require further development for practical real-world applications.

4.8. Future Work

The advancements made in model-based gait recognition through this work are promising, yet there remains substantial room for improvement. One critical area for enhancement is the pose estimator, a crucial component of the gait recognition system. The accuracy of the pose estimator directly influences the performance of the gait recognition system. Future research could focus on improving the precision of pose estimation, which would subsequently lead to enhanced gait recognition performance.

A second emerging trend is the integration of multiple modalities. While this work exclusively utilizes the skeleton representation of the human body, gait recognition can also be performed using other modalities such as silhouettes, depth images, optical flow, or RGB images. The fusion of multiple modalities has the potential to significantly improve the performance of gait recognition systems [153, 30, 117, 176]. Future work should also investigate optimal methods for fusing and normalizing these diverse modalities.

The trajectory for gait representations is moving towards 3D and model-based representations. 3D representations offer increased robustness to viewpoint changes, while model-based representations provide enhanced interpretability and can capture more detailed gait information. In the domain of skeleton-based gait recognition, all currently published works still utilize 2D keypoints. Our experiments demonstrate that achieving superior performance with 3D keypoints is non-trivial. Future research

should focus on improving the performance of 3D keypoint-based systems and their application in gait recognition.

Lastly, future work could explore the development of an end-to-end trainable model capable of learning both model-based guidance and visual features within a single network. This approach would enable the network to autonomously determine the most salient features for gait recognition from the data and potentially operate on unlabeled datasets.

Gait Recognition vs. Long-Term Person Recognition Having identified future directions for model-based gait recognition, we now turn our attention to a broader examination of gait recognition as a field. Our focus on skeleton-based gait recognition aimed to reintroduce authentic gait features into the process. We observed that contemporary gait recognition methods heavily rely on appearance information, and our objective was to model detailed temporal aspects. However, it is pertinent to question the significance of this distinction in real-world applications. Primarily, gait recognition serves the purpose of long-term person Re-Identification (re-ID).

In contrast to re-ID, which focuses on short-term person recognition, typically in non-overlapping cameras within the same scene, gait recognition emphasizes long-term person recognition over extended periods, accommodating changing appearances. Historically, gait recognition has incorporated more temporal information methods, but the two fields are increasingly converging with the advent of video re-ID [178]. It is conceivable that video re-ID approaches may implicitly learn temporal gait features. This convergence leads us to the next differentiation between these fields: representation. Gait recognition consistently aims to find a representation that obscures appearance information, while re-ID operates directly on RGB inputs. In this work, we also challenged the assumption that currently utilized representations, namely silhouettes, effectively conceal most appearance information. This further diminishes the distinctions between the two fields. The final difference lies in the data utilized. Gait recognition has historically focused on controlled laboratory environments, whereas re-ID has always emphasized real-world scenarios. With the introduction of new datasets such as GREW and Gait3D, gait recognition is also transitioning to more naturalistic settings, further narrowing the gap between the two fields.

Examining gait recognition surveillance applications, we observe an increasing merger of these two domains. The future trajectory of gait recognition appears to be moving towards re-ID, adopting similar methodologies and datasets. This shift will facilitate more robust and generalizable gait recognition systems. Additionally, in a surveillance context, we can leverage another biometric: the face. Facial recognition presents a more reliable biometric than gait and can be employed to enhance the performance of recognition systems. We discuss the challenges associated with these face recognition systems in our other works [6[†], 5[†], 7[†]].

For surveillance-targeted applications, we conclude that gait recognition should focus less on finding representations that obscure appearance information and more on developing models that generalize towards long-term recognition, incorporating appearance, gait, and facial information. While additional guidance from representations such as skeletons may improve recognition system performance, it should be considered a secondary focus.

For other applications, such as medical and forensic gait analysis, gait methods, particularly 3D model-based representations, can still prove valuable. These domains often work with significantly less data and require higher model interpretability.

4.9. Conclusion

This chapter introduces a novel skeleton-based gait representation and formulates an advanced skeleton-based recognition architecture. We propose a multi-branch architecture for gait recognition that utilizes precomputed temporal information strategically divided into three branches: joints, motion, and bones.

We conduct a comprehensive comparison of our approach with state-of-the-art methods across four prominent gait datasets. Notably, we achieve significant performance gains on OUMVLP, the largest indoor skeleton-based gait dataset, demonstrating a 3-fold performance improvement over the baseline. This performance establishes our method as the current state-of-the-art. Furthermore, our training methodology exhibits superior efficiency and stability compared to previous graph-based approaches. Consequently, our approach has been adopted as the baseline in subsequent studies [176, 183] for skeleton-based gait recognition.

In our extensive ablation study, we meticulously analyze the learned gait representation of the network, demonstrating robust temporal modeling. We also identify and elucidate the challenges inherent in full 4D gait recognition, providing evidence that 2D pose currently serves as the optimal representation for gait recognition. Our analysis further reveals the critical dependence of model performance on the quality of the keypoint estimator. Additionally, we highlight the limitations of current skeleton-based methods in cross-domain generalization.

To provide deeper insights, we visualize the network’s attention to the skeleton data, revealing a pronounced focus on the outer limbs and limbs exhibiting forward motion.

These observations substantiate that our model effectively captures *authentic* gait features, transcending mere spatial recognition. When combined with the advantages of robust pose estimation, this approach enables a broader spectrum of applications in gait recognition.

We anticipate with enthusiasm the future work that will build upon our approach, further advancing the performance and applicability of gait recognition systems.

5

Conclusion

This chapter presents a comprehensive reflection on the key findings and contributions of our research in the domains of gait recognition and multi-view tracking. We will elucidate the implications of our work for the broader field of computer vision and propose directions for future research. By synthesizing the advancements made in gait recognition and multi-view tracking, we aim to underscore the significance of our contributions and their potential impact on real-world applications.

5.1. Contributions

Skeleton Gait Representation After identifying the limitations of existing gait representation approaches, we introduced skeleton-based gait representations derived from 2D and 3D human pose estimation. Our research demonstrates that skeleton-based gait representations serve as a viable alternative to silhouette-based representations for gait recognition. Although the performance currently falls short of silhouette-based approaches, other studies indicate that when combined with silhouette-based representations, skeleton-based gait representations yield state-of-the-art performance. Furthermore, our work has paved the way for the introduction of more model-based representations in the gait recognition community.

Graph Neural Networks for Gait Recognition Following the identification of the potential of skeleton-based gait representations, we introduced a novel approach

to gait recognition based on Graph Convolutional Network (GCN). While other research areas such as skeleton-based action recognition have been advanced by Graph Neural Network (GNN), gait recognition remains predominantly dominated by Convolutional Neural Networks (CNNs). A notable strength of the Spatial-Temporal Graph Convolutional Network (ST-GCN) architecture is its robust focus on temporal information, as evidenced by our ablation studies. We demonstrate that our approach is primarily driven by temporal information, whereas other methods tend to focus more on static information. This characteristic renders our approach a *bona fide* gait recognition system.

Explainability of Gait Recognition Explainability is a critical aspect in the field of biometric recognition. Given that gait is a highly distinctive feature, explainability is crucial not only for transparency but also for fostering trust in the technology. We visualized the activations of the ST-GCN to gain insights into which skeletal components are most salient for the classification task. Our analysis reveals that our architecture primarily focuses on leg swing and arm swing movements.

Multi-View Aggregation The aggregation of multiple views is a powerful technique in detection and tracking tasks. To date, only rudimentary methods have been employed to aggregate views in Bird's Eye View (BEV). We conducted a comparative study evaluating recent aggregation methods, establishing a new state-of-the-art in single-stage multi-view detection. Additionally, we introduced a simple yet effective temporal aggregation technique to enhance results in a causal manner. We also proposed a novel lifting technique that models the 3D space in a continuous manner with high computational efficiency.

One-Stage Multi-Camera Tracking We propose the first one-stage multi-camera tracking approach for pedestrian tracking. We introduced an early fusion architecture that bypasses 2D detection and directly utilizes 3D location for tracking. This approach reduces downstream errors in both association tasks and demonstrated state-of-the-art performance on popular benchmarks. Our work also underscores that for the tracking task, temporal information is more critical than visual information.

5.2. Conclusion

This thesis makes significant contributions to the field of computer vision and provides novel insights into 4D human understanding. While both chapters are interconnected within a typical perception or surveillance pipeline, the logical progression would be to integrate both in a real-world application. This could manifest as a multi-view gait recognition system that tracks humans in 3D and recognizes them based on their 3D skeleton-based gait representation. As no public dataset currently exists for this task, we propose it as an intriguing challenge for future research.

Our work identifies a clear trend from 2D to 3D methodologies in the field of surveillance. While the majority of current research relies on 2D detections, we demonstrate that the next evolutionary step necessitates 3D detections to further advance the state-of-the-art. Another observable trend is the transition from laboratory conditions to real-world scenarios. Recent years have witnessed improvements in detection and recognition accuracy even under real-world conditions. This progress indicates the feasibility of automated surveillance systems and underscores the need for public awareness regarding the capabilities of such systems.

In conclusion, we posit that our work has significantly expanded the boundaries of gait recognition and multi-view tracking. We anticipate that our research will serve as a robust foundation for further advancements in the field and will be built upon by future researchers.

5.3. Outlook

After giving an outlook for each chapter, in this section we want to derive general trends for human perception and surveillance systems. Several key directions emerge from our research that will likely shape the future of this field:

From 2D to 4D Understanding A clear trend emerges towards more comprehensive spatial-temporal modeling. While current systems predominantly operate in 2D or 3D space, future developments will likely embrace full 4D understanding (3D space + time) of human behavior. This shift is evidenced by the increasing adoption of multi-sensor systems and temporal modeling in both tracking and recognition tasks. The integration of temporal information, as demonstrated in our work, is crucial for robust human understanding and will continue to gain importance.

Fusion of Multiple Modalities Future systems will likely move beyond purely visual information, incorporating multiple sensing modalities such as LiDAR, event cameras, and potentially other sensors. This multi-modal approach, as highlighted in our review of recent trends, can provide more robust performance across varying environmental conditions and potentially enable new capabilities in human understanding.

Privacy-Aware Systems As surveillance systems become more capable, there is an increasing need for privacy-preserving approaches. Skeleton-based representations, as explored in our work, offer an interesting direction as they capture behavioral characteristics while inherently anonymizing personal appearance. Future research should further explore the balance between system capability and privacy preservation.

Real-World Deployment The transition from laboratory conditions to real-world scenarios presents ongoing challenges that future research must address. These include handling varying lighting conditions, occlusions, and crowded scenes. Additionally, the computational efficiency of these systems needs to be improved for practical deployment, particularly for real-time applications.

Human-Centric Analysis A significant trend is the shift towards more sophisticated human-centric analysis. Future systems will likely move beyond simple detection and tracking to understand complex human behaviors, interactions between multiple people, and human-object interactions. This will require advances in both representation learning and temporal modeling.

These trends suggest that the field is moving towards more comprehensive, privacy-aware, and practically deployable systems for human understanding. Future research will need to address these challenges while maintaining high accuracy and computational efficiency. The integration of these various aspects presents exciting opportunities for innovation in computer vision and surveillance systems.

A

Tracking Metrics

Multiple Object Tracking (MOT) is the task of detecting the presence of multiple objects in video, and associating these detections over time according to object identities. In order to evaluate how well a tracker performs, we need to compare its output to a ground-truth set of tracking results. The purpose of an evaluation metric is to evaluate the similarity between the given ground-truth and the tracking results. In this appendix, we will first set the preliminaries of tracking and later discuss the most common metrics used in MOT evaluation.

This appendix is based on the work of [9] and [105].

Ground-Truth and Prediction Format Ground-truth tracks are represented as a set of detections (gtDets) in each video frame. Each gtDet has a unique ID (gtID) within a frame, consistent over time for detections from the same ground-truth trajectory (gtTraj).

For most evaluation metrics (e.g., MOTA, IDF1), a tracker's prediction follows the same format. It consists of predicted detections (prDets) in each frame, each with a predicted ID (prID), unique within a frame and consistent over time for detections from the same predicted trajectory (prTraj).

When tracking multiple object classes, each gtTraj and prTraj may have a class ID. Metrics are typically applied per class and averaged, so we can assume calculations are for a single object class at a time.

Types of Tracking Errors Errors between predicted and ground-truth tracks can be classified into three categories:

1. Detection errors: The tracker predicts detections that don't exist in the ground-truth, or fails to predict existing ground-truth detections.
2. Association errors: The tracker assigns the same prID to detections with different gtIDs, or different prIDs to detections that should have the same gtID.
3. Localisation errors: Predicted detections (prDets) are not perfectly spatially aligned with ground-truth detections (gtDets).

These three error types are widely applicable for evaluating tracking across various scenarios, and are the focus when analyzing and comparing evaluation metrics.

Measures of Similarity Most metrics require a similarity score between two detections, S , constrained between 0 and 1. When S is 1, the predicted detection (prDet) and ground-truth detection (gtDet) perfectly align; when S is 0, there is no overlap. Common similarity metrics include:

- For 2D boxes, 3D boxes, and segmentation masks: Intersection over Union (IoU)
- For point representations and human joint locations: 1 minus the Euclidean distance (with zero overlap beyond 1 meter)

The choice of similarity metric depends on the detection representation used.

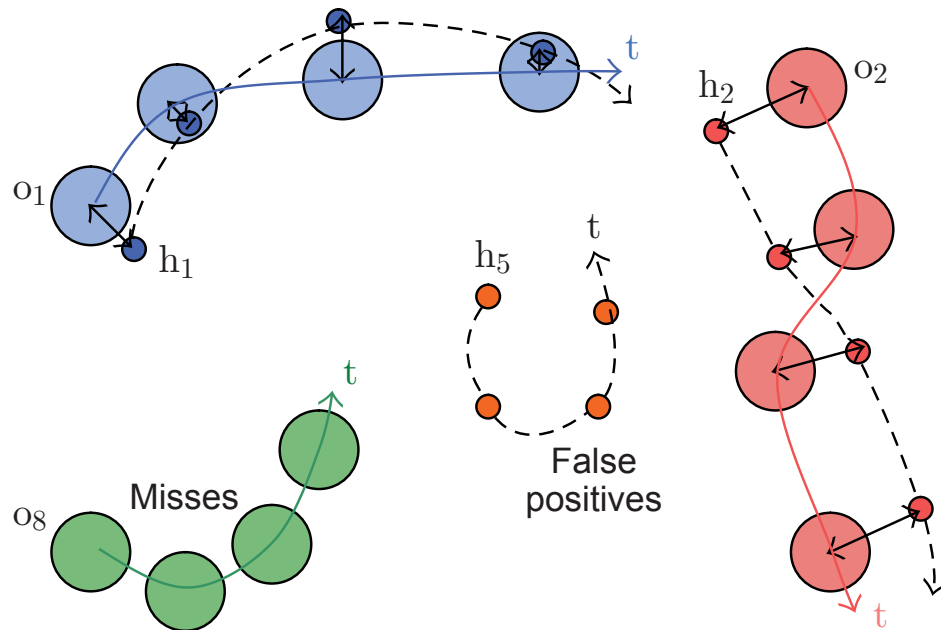


Figure A.1.: **Bijective Matching.** A common procedure in MOT evaluation metrics is to perform a bijective (one-to-one) matching between gtDets and prDets at every time step t .

Bijjective Matching A common procedure in MOT evaluation metrics (MOTA, IDF1) is to perform a bijective (one-to-one) matching between gtDets and prDets. This ensures that all gtDets are matched to at most one prDet and vice versa, penalizing extra or missed predictions. This bijective mapping is calculated using the Hungarian algorithm to optimize the sum of matching scores between all gtDet-prDet pairs. Usually, a minimum similarity $\mathcal{S} \geq \alpha$ is required for a match. After matching, we have:

- True positives (TP): Matched gtDet-prDet pairs (correct predictions)
- False negatives (FN): Unmatched gtDets (missed predictions)
- False positives (FP): Unmatched prDets (extra predictions)

FNs and FPs represent incorrect predictions.

CLEARMOT: MOTA and MOTP

MOTA (Multi-Object Tracking Accuracy) performs matching at the detection level. A bijective mapping is constructed between prDets and gtDets in each frame. Matching requires spatial similarity, defined by a similarity score $\mathcal{S}$ (e.g., IoU for 2D boxes, or distance for point representations) and threshold α . Matches are only allowed when $\mathcal{S} \geq \alpha$. The final matching optimizes MOTA and MOTP scores. MOTA measures association using Identity Switches (IDSW). An IDSW occurs when a tracker swaps object identities or reinitializes a lost track with a different identity. Formally, an IDSW is a TP with a prID different from the previous TP's prID (with the same gtID). IDSWs only consider errors relative to the previous TP, not ID transfers.

MOTA scoring measures three error types: FNs, FPs (detection errors), and IDSWs (association errors). The final MOTA score is:

$$\text{MOTA} = 1 - \frac{|\text{FN}| + |\text{FP}| + |\text{IDSW}|}{|\text{gtDets}|} \quad (\text{A.1})$$

Note that MOTA does not measure localisation error. Instead, CLEARMOT metrics define a secondary metric, MOTP (Multi-Object Tracking Precision), which solely measures localisation accuracy:

$$\text{MOTP} = \frac{1}{|\text{TP}|} \sum_{(i,j) \in \text{TP}} \mathcal{S}(i,j) \quad (\text{A.2})$$

where $\mathcal{S}(i,j)$ is the similarity score between matched gtDet i and prDet j .

The Identification Metrics: IDF1, IDP, IDR

IDF1 calculates a bijective (one-to-one) mapping between gtTrajs and prTrajs sets, unlike MOTA's detection-level matching. This defines:

- IDTPs: Matches on overlapping trajectory parts where $S \geq \alpha$
- IDFNs and IDFPs: Remaining gtDets and prDets from non-overlapping sections and unmatched trajectories

The IDF1, IDP, and IDR metrics are defined as:

$$\text{IDR} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}} \quad (\text{A.3})$$

$$\text{IDP} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFP}} \quad (\text{A.4})$$

$$\text{IDF1} = \frac{2 \cdot \text{IDTP}}{2 \cdot \text{IDTP} + \text{IDFP} + \text{IDFN}} \quad (\text{A.5})$$

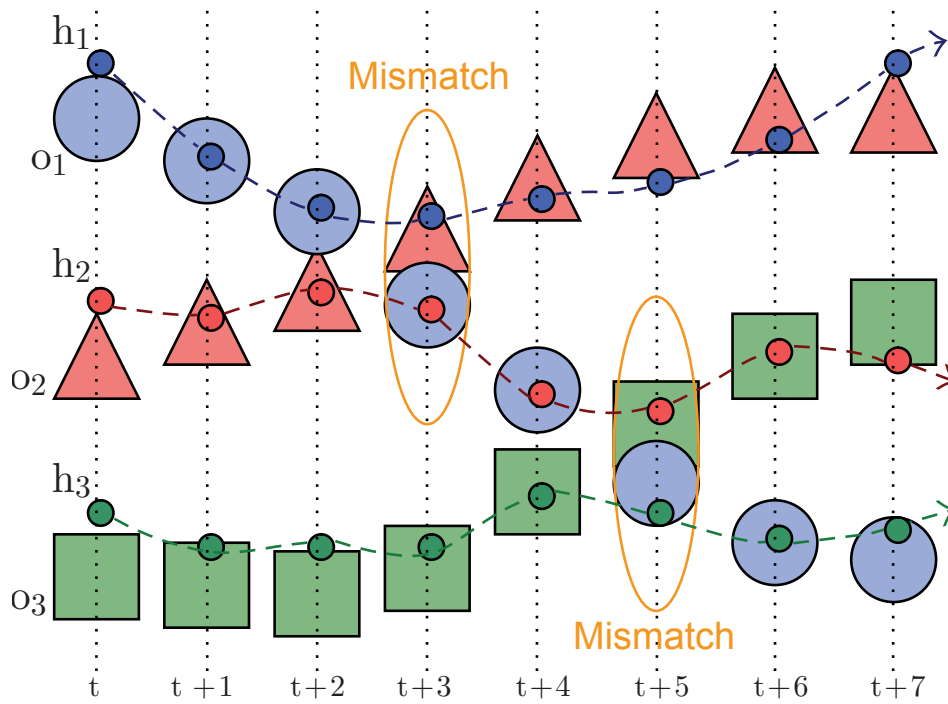


Figure A.2.: **Mismatched Tracks.** Here, h_2 is first mapped to o_2 . After a few frames, though, o_1 and o_2 cross paths and h_2 follows the wrong object. Later, it wrongfully swaps again to o_3 .

References

- [1] Timilehin B Aderinola, Tee Connie, Thian Song Ong, Andrew Beng Jin Teoh, and Michael Kah Ong Goh. Gait-based age group classification with adaptive graph neural network. *Engineering Applications of Artificial Intelligence*, 137:109081. Elsevier, 2024. → p. 65
- [2] Weizhi An, Shiqi Yu, Yasushi Makihara, Xinhui Wu, Chi Xu, Yang Yu, Rijun Liao, and Yasushi Yagi. Performance Evaluation of Model-Based Gait on Multi-View Very Large Population Database With Pose Sequences. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):421–430, 2020. → pp. 88, 89, 94
- [3] Gunawan Ariyanto and Mark S Nixon. Model-based 3D gait biometrics. In *2011 international joint conference on biometrics (IJCB)*, pp. 1–7. IEEE, 2011. → p. 72
- [4] Maryam Babaei, Linwei Li, and Gerhard Rigoll. Gait energy image reconstruction from degraded gait cycle using deep learning. In *European Conference on Computer Vision (ECCV) Workshops*, 2018. → p. 68
- [5] Maryam Babaei, Yue Zhu, Okan Köpüklü, Stefan Hörmann, and Gerhard Rigoll. Gait Energy Image Restoration Using Generative Adversarial Networks. In *IEEE International Conference on Image Processing*, pp. 2596–2600, 2019. → pp. xviii, 66, 71
- [6] Pierre Baqué, François Fleuret, and Pascal Fua. Deep occlusion reasoning for multi-camera multi-target detection. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 271–279, 2017. → p. 55
- [7] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819. IEEE, 2011. → pp. 36, 39
- [8] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 941–951, 2019. → pp. 36, 37, 38, 39

- [9] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10. Springer, 2008. → pp. xiv, 54, 115
- [10] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing*, 2016. → pp. 36, 37
- [11] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-Speed tracking-by-detection without using image information. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017. → p. 36
- [12] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, pp. 561–578. Springer, 2016. → pp. 70, 72
- [13] Esther Bosch, Raquel Le Houcq Corbí, Klas Ihme, Stefan Hörmann, Meike Jipp, and David Käthner. Frustration Recognition Using Spatio Temporal Data: A Novel Dataset and GCN Model to Recognize In-Vehicle Frustration. *IEEE Transactions on Affective Computing*, 14(4):2864–2875. IEEE, 2022. → p. 65
- [14] Guillem Brasó and Laura Leal-Taixé. Learning a Neural Solver for Multiple Object Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. → pp. 36, 37
- [15] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. → pp. xiii, 19
- [16] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020. → p. 54
- [17] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. → pp. 70, 72, 77, 78, 79, 89
- [18] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742, 2016. → p. 78
- [19] Rich Caruana. Multitask learning. *Machine learning*. Springer, 1997. → p. 13

- [20] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. GaitSet: Regarding Gait as a Set for Cross-View Gait Recognition. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 8126–8133, 2019. → pp. 66, 70, 71, 72, 73, 93, 95, 96
- [21] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024. → pp. 39, 40
- [22] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5030–5039, 2018. → pp. xiii, 31, 32, 51, 54, 56, 62
- [23] Tatjana Chavdarova and François Fleuret. Deep multi-camera people detection. In *International Conference on Machine Learning and Applications (ICMLA)*, pp. 848–853. IEEE, 2017. → p. 55
- [24] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7035–7043, 2017. → pp. 70, 72, 78, 80, 98
- [25] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4974–4983, 2019. → p. 96
- [26] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *IEEE International Conference on Multimedia and Expo*, pp. 1–6. IEEE, 2018. → p. 39
- [27] Cheng-Che Cheng, Min-Xuan Qiu, Chen-Kuo Chiang, and Shang-Hong Lai. ReST: A Reconfigurable Spatial-Temporal Graph Model for Multi-Camera Multi-Object Tracking. *arXiv preprint arXiv:2308.13229*, 2023. → pp. 32, 38, 39, 56
- [28] Adrian Cosma and Emilian Radoi. Psymo: A dataset for estimating self-reported psychological traits from gait. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 4603–4613, 2024. → p. 65
- [29] Christian Creß, Walter Zimmer, Leah Strand, Maximilian Fortkord, Siyi Dai, Venkatnarayanan Lakshminarasimhan, and Alois Knoll. A9-dataset: Multi-sensor infrastructure-based dataset for mobility research. In *IEEE Intelligent Vehicles Symposium (IV)*, pp. 965–970. IEEE, 2022. → p. 31
- [30] Yufeng Cui and Yimei Kang. Multi-modal gait recognition via effective spatial-temporal feature fusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17949–17957, 2023. → pp. 73, 74, 105

- [31] Antitza Dantcheva, Carmelo Velardo, Angela D'angelo, and Jean-Luc Dugelay. Bag of soft biometrics for person identification: New trends and challenges. *Multimedia Tools and Applications*, 51:739–777. Springer, 2011. → p. 65
- [32] Patrick Dendorfer, Vladimir Yugay, Aljosa Osep, and Laura Leal-Taixé. Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking? *Advances in Neural Information Processing Systems (NeurIPS)*, 35:15657–15671, 2022. → p. 36
- [33] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019. → p. 15
- [34] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, N. Houlsby, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. → pp. xiii, 7, 21
- [35] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017. → p. 52
- [36] Martin Engilberge, Weizhe Liu, and Pascal Fua. Multi-view Tracking Using Weakly Supervised Human Motion Prediction. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023. → p. 56
- [37] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. IEEE, 2008. → p. 39
- [38] Chao Fan, Saihui Hou, Jilong Wang, Yongzhen Huang, and Shiqi Yu. Learning gait representation from massive unlabelled walking videos: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2023. → pp. 73, 74, 95, 96
- [39] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. OpenGait: Revisiting Gait Recognition Towards Better Practicality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9707–9716, June 2023. → pp. 72, 95, 102
- [40] Chao Fan, Jingzhe Ma, Dongyang Jin, Chuanfu Shen, and Shiqi Yu. Skeleton-Gait: Gait Recognition Using Skeleton Maps. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 1662–1669, 2024. → pp. 73, 74, 95, 96
- [41] Chao Fan, Yunjie Peng, Chunshui Cao, Xu Liu, Saihui Hou, Jiannan Chi, Yongzhen Huang, Qing Li, and Zhiqiang He. GaitPart: Temporal Part-Based Model for Gait Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. → pp. 66, 70, 71, 72, 73, 74, 93, 95, 96, 97

- [42] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. → pp. 78, 79
- [43] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional Multi-Person Pose Estimation. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. → pp. 70, 72, 89
- [44] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3038–3046, 2017. → pp. 37, 38, 39
- [45] Yang Feng, Yuncheng Li, and Jiebo Luo. Learning effective Gait features using LSTM. In *International Conference on Pattern Recognition (ICPR)*, pp. 325–330, 2016. → pp. 66, 71
- [46] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *IEEE international workshop on performance evaluation of tracking and surveillance*, pp. 1–6. IEEE, 2009. → pp. 31, 51, 62
- [47] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2007. → pp. 38, 51, 55
- [48] Yang Fu, Shibei Meng, Saihui Hou, Xuecai Hu, and Yongzhen Huang. Gpgait: Generalized pose-based gait recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 19595–19604, 2023. → pp. xv, 73, 95, 97, 99
- [49] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, pp. 1231–1237. Sage Publications Sage UK: London, England, 2013. → p. 54
- [50] Justin Gilmer, Sam Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *International Conference on Machine Learning (ICML)*, 2017. → p. 20
- [51] Derek Gloudemans, Yanbing Wang, Gracie Gumm, William Barbour, and Daniel B Work. The Interstate-24 3D Dataset: a new benchmark for 3D multi-camera vehicle tracking. *arXiv preprint arXiv:2308.14833*, 2023. → p. 31
- [52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>. → p. 8
- [53] Song Guo, Jingya Wang, Xinchao Wang, and Dacheng Tao. Online Multiple Object Tracking With Cross-Task Synergy. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. → p. 37

- [54] Xianda Guo, Zheng Zhu, Tian Yang, Beibei Lin, Junjie Huang, Jiankang Deng, Guan Huang, Jie Zhou, and Jiwen Lu. Gait Recognition in the Wild: A Large-scale Benchmark and NAS-based Baseline. *arXiv e-prints*, pp. arXiv–2205, 2022. → pp. 72, 73
- [55] Jinguang Han and Bir Bhanu. Individual recognition using gait energy image. *PAMI*, 28(2):316–322. IEEE, IEEE, 2005. → pp. 66, 68, 69, 71
- [56] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What Really Matters for Multi-Sensor BEV Perception? In *International Conference on Robotics and Automation (ICRA)*, 2023. → pp. 32, 35, 45, 46, 54, 56, 57, 62
- [57] Adam W Harley, Shrinidhi K Lakshmikanth, Fangyu Li, Xian Zhou, Hsiao-Yu Fish Tung, and Katerina Fragkiadaki. Learning from unlabelled videos using contrastive predictive neural 3d mapping. *arXiv preprint arXiv:1906.03764*, 2019. → pp. 35, 46
- [58] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. → pp. 33, 44
- [59] Jules. Harvey, Adam. LaPlace. Exposing.ai, 2021. → p. 62
- [60] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, 2017. → p. 37
- [61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. → pp. 17, 18, 41, 76, 87
- [62] Tong He, Pei Sun, Zhaoqi Leng, Chenxi Liu, Dragomir Anguelov, and Mingxing Tan. LEF: Late-to-Early Temporal Fusion for LiDAR 3D Object Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1637–1644, 2023. → p. 42
- [63] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. → pp. 15, 92
- [64] Martin Hofmann, Sebastian Bachmann, and Gerhard Rigoll. 2.5D gait biometrics using the Depth Gradient Histogram Energy Image. In *IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 399–403, 2012. → pp. 68, 69, 71
- [65] Martin Hofmann, Jürgen Geiger, Sebastian Bachmann, Björn Schuller, and Gerhard Rigoll. The TUM Gait from Audio, Image and Depth (GAID) database: Multimodal recognition of subjects and traits. *Journal of Visual Communication and Image Representation*, 25(1):195–206. Elsevier, 2014. → pp. xiv, 68

- [66] Martin Hofmann and Gerhard Rigoll. Exploiting gradient histograms for gait-based person identification. In *IEEE International Conference on Image Processing*, pp. 4171–4175, 2013. → pp. 68, 69
- [67] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3650–3657, 2013. → pp. 32, 38, 39
- [68] Saihui Hou, Chunshui Cao, Xu Liu, and Yongzhen Huang. Gait lateral network: Learning discriminative and compact representations for gait recognition. In *European Conference on Computer Vision (ECCV)*, pp. 382–398. Springer, 2020. → pp. 72, 73
- [69] Yunzhong Hou and Liang Zheng. Multiview Detection with Shadow Transformer (and View-Coherent Data Augmentation). In *ACM International Conference on Multimedia*, 2021. → pp. xiii, 34, 35, 44, 54, 55, 58, 62
- [70] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview Detection with Feature Perspective Transformation. In *European Conference on Computer Vision (ECCV)*, 2020. → pp. 31, 34, 35, 44, 51, 52, 54, 55, 58, 62
- [71] Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou, and Steve Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):663–671. IEEE, 2006. → p. 39
- [72] Xiaohu Huang, Duowang Zhu, Hao Wang, Xinggang Wang, Bo Yang, Botao He, Wenyu Liu, and Bin Feng. Context-sensitive temporal feature learning for gait recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 12909–12918, 2021. → pp. 72, 73, 95, 96
- [73] Zhen Huang, Dixiu Xue, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xian-Sheng Hua. 3d local convolutional neural networks for gait recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 14920–14929, 2021. → pp. 72, 93, 94
- [74] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015. → p. 11
- [75] Haris Iqbal. HarisIqbal88/PlotNeuralNet v1.0.0, Dec. 2018. → p. xiii
- [76] Hao Jiang, Sidney S. Fels, and James J. Little. A Linear Programming Approach for Multiple Object Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. → p. 36
- [77] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211. Springer, 1973. → pp. 66, 104

- [78] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960. → p. 38
- [79] Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Kozhova, and Jing Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336. IEEE, 2008. → p. 54
- [80] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Dretakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), July 2023. → pp. 39, 50
- [81] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020. → pp. 15, 91
- [82] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. → pp. 11, 91
- [83] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. → pp. 20, 84, 85
- [84] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In *European Conference on Computer Vision (ECCV)*, pp. 734–750, 2018. → p. 42
- [85] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. → p. 36
- [86] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1987–1994. IEEE, 2012. → p. 39
- [87] Wei-Yu Lee, Ljubomir Jovanov, and Wilfried Philips. Multi-view Target Transformation for Pedestrian Detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 90–99, 2023. → pp. 34, 35, 55, 58, 62
- [88] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10863–10872, 2019. → p. 77

- [89] Xiang Li, Yasushi Makihara, Chi Xu, and Yasushi Yagi. Multi-view large population gait database with human meshes and its performance evaluation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):234–248. IEEE, 2022. → p. 89
- [90] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, Shiqi Yu, and Mingwu Ren. End-to-end model-based gait recognition. In *Asian Conference on Computer Vision (ACCV)*, 2020. → pp. 74, 75
- [91] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. Simcc: A simple coordinate classification perspective for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pp. 89–106. Springer, 2022. → p. 93
- [92] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, pp. 1–18, 2022. → pp. 32, 35, 47, 62
- [93] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *PAMI*, 25(12):1505–1518. IEEE, 2003. → pp. 66, 71
- [94] Rijun Liao, Chunshui Cao, Edel B. Garcia, Shiqi Yu, and Yongzhen Huang. Pose-Based Temporal-Spatial Network (PTSN) for Gait Recognition with Carrying and Clothing Variations. In Jie Zhou, Yunhong Wang, Zhenan Sun, Yong Xu, Linlin Shen, Jianjiang Feng, Shiguang Shan, Yu Qiao, Zhenhua Guo, and Shiqi Yu, editors, *Biometric Recognition*, pp. 474–483, Cham, 2017. Springer International Publishing. → pp. 67, 70, 72
- [95] Rijun Liao, Shiqi Yu, Weizhi An, and Yongzhen Huang. A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognition*, 98:107069, 2020. → pp. 67, 72, 73, 77, 80, 93, 98
- [96] Beibei Lin, Shunli Zhang, and Feng Bao. Gait recognition with multiple-temporal-scale 3d convolutional neural network. In *ACM International Conference on Multimedia*, pp. 3054–3062, 2020. → pp. 71, 72
- [97] Beibei Lin, Shunli Zhang, and Xin Yu. Gait recognition via effective global-local feature representation and local temporal aggregation. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 14648–14656, 2021. → pp. 72, 73, 95, 96
- [98] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, 2017. → pp. 23, 41
- [99] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2980–2988, 2017. → pp. 42, 78

- [100] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014. → pp. 77, 79, 98
- [101] Kunliang Liu, Ouk Choi, Jianming Wang, and Wonjun Hwang. CDGNet: Class Distribution Guided Network for Human Parsing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4473–4482, June 2022. → p. 71
- [102] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10012–10022, 2021. → pp. xiii, 7, 23, 41
- [103] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16. ACM, Oct. 2015. → pp. 70, 89, 96
- [104] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. RetinaTrack: Online Single Stage Joint Detection and Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. → p. 37
- [105] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129:548–578. Springer, 2021. → p. 115
- [106] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8844–8854, 2022. → p. 36
- [107] Shibei Meng, Yang Fu, Saihui Hou, Chunshui Cao, Xu Liu, and Yongzhen Huang. FastPoseGait: A Toolbox and Benchmark for Efficient Pose-based Gait Recognition. *arXiv preprint arXiv:2309.00794*, 2023. → p. 95
- [108] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. → p. 40
- [109] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M. Chang, Y. Yao, L. Zheng, M. Shaiqur Rahman, A. Venkatachalapathy, A. Sharma, Q. Feng, V. Ablavsky, S. Sclaroff, P. Chakraborty, A. Li, S. Li, and R. Chellappa. The 6th AI City Challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3346–3355, 2022. → p. 39

- [110] Duy MH Nguyen, Roberto Henschel, Bodo Rosenhahn, Daniel Sonntag, and Paul Swoboda. LMGP: Lifted Multicut Meets Geometry Projections for Multi-Camera Multi-Object Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8866–8875, 2022. → pp. xiii, 32, 38, 39
- [111] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning (ICML)*, pp. 2014–2023. PMLR, 2016. → p. 83
- [112] Jonah Ong, Ba-Tuong Vo, Ba-Ngu Vo, Du Yong Kim, and Sven Nordholm. A bayesian filter for multi-view 3d multi-object tracking with occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2246–2263. IEEE, 2020. → p. 56
- [113] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. → p. 36
- [114] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-Dense Similarity Learning for Multiple Object Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. → p. 37
- [115] Hamza Reza Pavel, Enamul Karim, Ashish Jaiswal, Sneha Acharya, Gaurav Nale, Michail Theofanidis, and Fillia Makedon. Assessment of cognitive fatigue from gait cycle analysis. *Technologies*, 11(1):18. MDPI, 2023. → p. 65
- [116] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-Tracker: Chaining Paired Attentive Regression Results for End-to-End Joint Multiple-Object Detection and Tracking. In *European Conference on Computer Vision (ECCV)*, 2020. → p. 36
- [117] Yunjie Peng, Kang Ma, Yang Zhang, and Zhiqiang He. Learning rich features for gait recognition by integrating skeletons and silhouettes. *Multimedia Tools and Applications*, 83(3):7273–7294. Springer, 2024. → pp. 73, 74, 105
- [118] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision (ECCV)*, pp. 194–210. Springer, 2020. → pp. 32, 35, 46, 62
- [119] Walter Pirker and Regina Kitzenschlager. Gait disorders in adults and the elderly: A clinical guide. *Wiener Klinische Wochenschrift*, 129(3):81–95. Springer, 2017. → p. xiv
- [120] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. → p. 36

- [121] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *International Conference on Pattern Recognition (ICPR)*, pp. 694–701. Springer, 2021. → pp. 73, 76
- [122] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. → p. 11
- [123] Rui Qiu, Ming Xu, Yuyao Yan, Jeremy S Smith, and Xi Yang. 3d random occlusion and multi-layer projection for deep multi-camera pedestrian localization. In *European Conference on Computer Vision (ECCV)*, pp. 695–710. Springer, 2022. → pp. 35, 55, 58
- [124] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. → p. 37
- [125] Markus Reuter. Panoptischer Rewe-Supermarkt: Einkauf mit skelettkontrolle, Jan 2024. → p. 104
- [126] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision (ECCV)*, pp. 17–35. Springer, 2016. → p. 54
- [127] Herbert Robbins. An empirical Bayes approach to statistics. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 1:157–163, 1951. → p. 10
- [128] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386. American Psychological Association, 1958. → p. 8
- [129] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko. A Gentle Introduction to Graph Neural Networks. *Distill*, 2021. <https://distill.pub/2021/gnn-intro>. → p. xiv
- [130] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. → pp. 14, 15
- [131] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in Neural Information Processing Systems (NeurIPS)*, 16, 2003. → p. 71
- [132] Jenny Seidenschwarz, Guillem Brasó, Víctor Castro Serrano, Ismail Elezi, and Laura Leal-Taixé. Simple Cues Lead to a Strong Multi-Object Tracker. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13813–13823, 2023. → p. 39

- [133] Chuanfu Shen, Chao Fan, Wei Wu, Rui Wang, George Q. Huang, and Shiqi Yu. LidarGait: Benchmarking 3D Gait Recognition With Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1054–1063, June 2023. → pp. 68, 75
- [134] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12026–12035, 2019. → p. 76
- [135] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Geinet: View-invariant gait recognition using a convolutional neural network. In *2016 international conference on biometrics (ICB)*, pp. 1–8. IEEE, 2016. → pp. 71, 73, 95, 96
- [136] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Multi-commodity network flow for tracking multiple people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1614–1627. IEEE, 2013. → p. 39
- [137] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, p. 1100612. International Society for Optics and Photonics, 2019. → p. 91
- [138] Anna Sokolova and Anton Konushin. Pose-based deep gait recognition. *IET Biometrics*, 8(2):134–143. IET, 2018. → p. 67
- [139] Chunfeng Song, Yongzhen Huang, Yan Huang, Ning Jia, and Liang Wang. GaitNet: An end-to-end network for gait based human identification. *Pattern Recognition*, 96:106988. Elsevier, 2019. → pp. 66, 70, 71, 73, 75, 93
- [140] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6049–6057, 2021. → pp. 34, 35, 55, 62
- [141] Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *ACM International Conference on Multimedia*, pp. 1625–1633, 2020. → pp. xiv, 75, 76, 85, 86
- [142] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. → p. 13
- [143] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep High-Resolution Representation Learning for Human Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019. → pp. 66, 77, 78, 79, 89

- [144] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. TransTrack: Multiple-Object Tracking with Transformer. *arXiv preprint arXiv:2012.15460*, 2020. → p. 36
- [145] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, One-stage, Regression of Multiple 3D People. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. → pp. 70, 72, 78, 80, 91, 96
- [146] Noriko Takemura, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition. *IPSJ Transactions on Computer Vision and Applications*, 10(1):4, Feb 2018. → pp. xiv, 88, 89, 90
- [147] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8797–8806, 2019. → pp. 31, 39, 52, 63
- [148] Ying Tao, Chip Hong Chang, Shengyu Gao, et al. GaitSpike: event-based gait recognition with spiking neural network. In *IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, pp. 357–361. IEEE, 2024. → p. 75
- [149] A. Torralba, P. Isola, and W.T. Freeman. *Foundations of Computer Vision*. Adaptive Computation and Machine Learning series. MIT Press, 2024. → p. xiii
- [150] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. → pp. 21, 22
- [151] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018. → p. 20
- [152] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. MOTs: Multi-Object Tracking and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. → p. 37
- [153] Likai Wang and Jinyan Chen. A Two-Branch Neural Network for Gait Recognition. *arXiv preprint arXiv:2202.10645*, 2022. → pp. 74, 105

- [154] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Ganzorig Batnasan, Fady Alnajjar, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th AI City Challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024. → p. 39
- [155] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection. *arXiv preprint arXiv:2303.11926*, 2023. → p. 33
- [156] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. EV-gait: Event-based robust gait recognition using dynamic vision sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6358–6367, 2019. → p. 75
- [157] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision (ECCV)*, pp. 107–122. Springer, 2020. → pp. 37, 38, 39
- [158] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10359–10366. IEEE, 2020. → p. 54
- [159] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. In *IEEE International Conference on Image Processing*, pp. 3645–3649. IEEE, 2017. → pp. 37, 39
- [160] Thomas Wolf, Mohammadreza Babaei, and Gerhard Rigoll. Multi-view Gait Recognition using 3D convolutional neural networks. In *IEEE International Conference on Image Processing*, pp. 4165–4169. IEEE, 2016. → pp. 66, 70, 71
- [161] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):209–226. IEEE, 2016. → pp. 71, 73, 89, 95, 96
- [162] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4256–4265, 2016. → pp. 39, 51, 55, 62
- [163] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to Train Your Deep Multi-Object Tracker. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. → p. 37

- [164] ChewYean Yam, Mark S Nixon, and John N Carter. Automated person recognition by walking and running via model-based approaches. *Pattern Recognition*, 37(5):1057–1072. Elsevier, 2004. → p. 72
- [165] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*, 2018. → pp. 20, 75, 76, 81
- [166] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *PAMI*, 44(6):2872–2893. IEEE, 2021. → p. 91
- [167] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11784–11793, 2021. → p. 33
- [168] Quanzeng You and Hao Jiang. Real-time 3d deep multi-camera tracking. *arXiv preprint arXiv:2003.11753*, 2020. → p. 56
- [169] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition. In *International Conference on Pattern Recognition (ICPR)*, volume 4, pp. 441–444, 2006. → pp. xiv, 72, 73, 88, 89, 90
- [170] Fangao Zeng, Bin Dong, Tiancai Wang, Cheng Chen, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-End Multiple-Object Tracking with TRansformer. In *European Conference on Computer Vision (ECCV)*, 2022. → p. 36
- [171] Cun Zhang, Xing-Peng Chen, Guo-Qiang Han, and Xiang-Jie Liu. Spatial transformer network on skeleton-based gait recognition. *Expert Systems*, 40(6):e13244. Wiley Online Library, 2023. → pp. 73, 93, 95, 99
- [172] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. → p. 36
- [173] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1112–1121, 2020. → pp. 76, 77, 86
- [174] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087. Springer, 2021. → pp. 36, 37, 38, 39
- [175] Ziyuan Zhang, Luan Tran, Xi Yin, Yousef Atoum, Xiaoming Liu, Jian Wan, and Nanxin Wang. Gait Recognition via Disentangled Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. → pp. 70, 89

- [176] Jinkai Zheng, Xincheng Liu, Wu Liu, Lingxiao He, Chenggang Yan, and Tao Mei. Gait recognition in the wild with dense 3d representations and a benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20228–20237, 2022. → pp. xiii, xiv, 70, 73, 74, 88, 90, 95, 96, 105, 107
- [177] Jinkai Zheng, Xincheng Liu, Shuai Wang, Lihao Wang, Chenggang Yan, and Wu Liu. Parsing is all you need for accurate gait recognition in the wild. In *ACM International Conference on Multimedia*, pp. 116–124, 2023. → pp. 71, 72
- [178] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision (ECCV)*, pp. 868–884. Springer, 2016. → p. 106
- [179] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016. → p. 102
- [180] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision (ECCV)*, pp. 474–490. Springer, 2020. → pp. 36, 37, 38, 39, 62
- [181] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as Points. In *arXiv preprint arXiv:1904.07850*, 2019. → pp. 36, 37, 42
- [182] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *International Conference on Learning Representations*, 2021. → pp. xiii, 24
- [183] Zheng Zhu, Xianda Guo, Tian Yang, Junjie Huang, Jiankang Deng, Guan Huang, Dalong Du, Jiwen Lu, and Jie Zhou. Gait recognition in the wild: A benchmark. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 14789–14799, 2021. → pp. xiv, 70, 88, 89, 90, 95, 107

Publications

- [1[†]] Johannes Gilg, Torben Teepe, Fabian Herzog, and Gerhard Rigoll. The box size confidence bias harms your object detector. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1471–1480, 2023. → p. 4
- [2[†]] Johannes Gilg, Torben Teepe, Fabian Herzog, Philipp Wolters, and Gerhard Rigoll. Do We Still Need Non-Maximum Suppression? Accurate Confidence Estimates and Implicit Duplication Modeling with IoU-Aware Calibration. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 4850–4859, 2024. → p. 4
- [3[†]] Fabian Herzog, Junpeng Chen, Torben Teepe, Johannes Gilg, Stefan Hörmann, and Gerhard Rigoll. Synthehicle: multi-vehicle multi-camera tracking in virtual cities. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–11, 2023. → pp. xiv, 4, 31, 32, 38, 51, 52, 56, 62, 63
- [4[†]] Fabian Herzog, Xunbo Ji, Torben Teepe, Stefan Hörmann, Johannes Gilg, and Gerhard Rigoll. Lightweight multi-branch network for person re-identification. In *IEEE International Conference on Image Processing*, pp. 1129–1133. IEEE, 2021. → pp. 4, 14, 37, 65, 72, 91
- [5[†]] Stefan Hörmann, Tianlin Kong, Torben Teepe, Fabian Herzog, Martin Knoche, and Gerhard Rigoll. Face Morphing: Fooling a Face Recognition System Is Simple! *arXiv preprint arXiv:2205.13796*, 2022. → pp. 14, 65, 106
- [6[†]] Stefan Hörmann, Zeyuan Zhang, Martin Knoche, Torben Teepe, and Gerhard Rigoll. Attention-based partial face recognition. In *IEEE International Conference on Image Processing*, pp. 2978–2982. IEEE, 2021. → pp. 14, 65, 106
- [7[†]] Martin Knoche, Torben Teepe, Stefan Hörmann, and Gerhard Rigoll. Explainable model-agnostic similarity and confidence in face verification. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 711–718, 2023. → pp. 65, 102, 106
- [8[†]] Torben Teepe, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. Towards a deeper understanding of skeleton-based gait recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1569–1577, 2022. → pp. xiv, 4, 68, 73, 74, 93, 95, 99

- [9[†]] Torben Teepe, Ali Khan, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. GaitGraph: Graph convolutional network for skeleton-based gait recognition. In *IEEE International Conference on Image Processing*, pp. 2314–2318. IEEE, 2021. → pp. xiv, xviii, 4, 68, 73, 93, 95, 99
- [10[†]] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. EarlyBird: Early-Fusion for Multi-View Tracking in the Bird’s Eye View. In *IEEE Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 102–111, 2024. → pp. xiii, xiv, 4, 32, 33, 38, 43, 56, 58
- [11[†]] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. Lifting Multi-View Detection and Tracking to the Bird’s Eye View. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 667–676, June 2024. → pp. xiii, xiv, 4, 33, 38, 43, 56, 58
- [12[†]] Philipp Wolters, Johannes Gilg, Torben Teepe, Fabian Herzog, Anouar Laouichi, Martin Hofmann, and Gerhard Rigoll. Unleashing HyDRa: Hybrid Fusion, Depth Consistency and Radar for Unified 3D Perception. In *International Conference on Robotics and Automation (ICRA)*, 2025. → pp. 4, 32

Supervised Student Works

- [1⁺] Yassine Hamila. Enhancing Bird's Eye View Perception with Roadside Camera-LiDAR Sensor Fusion. *Master's Thesis*. Technical University of Munich, 2024. → p. 33
- [2⁺] Ali Khan. Skeleton-based Gait Recognition using Spatial-Temporal Graph Convolutional Networks. *Master's Thesis*. Technical University of Munich, 2020. → pp. xviii, 68
- [3⁺] Luca Klingenberg. Spatio-Temporal Action Recognition with Graph Neural Networks. *Master's Thesis*. Technical University of Munich, 2022.
- [4⁺] Zongyue Li. Towards GCN-based Recommendation System via Contrastive Learning. *Master's Thesis*. Technical University of Munich, 2022.
- [5⁺] Samuel Schedler. Graph Neural Solver for Multiple Person Pose Tracking. *Master's Thesis*. Technical University of Munich, 2022.
- [6⁺] Yuyin Wang. Multi-View 3D Detection and Tracking in Bird's-Eye View. *Master's Thesis*. Technical University of Munich, 2023. → p. 33

