Herbert Woisetschläger

Efficient Federated Learning Systems in Resource-Constrained Environments under Emerging AI Regulation

Technical
University
of Munich

Technische Universität München
TUM School of Computation, Information and Technology

Efficient Federated Learning Systems in Resource-Constrained Environments under Emerging AI Regulation

Herbert Woisetschläger

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

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Prof. Dr. Rüdiger Westermann

Prüfende der Dissertation:

1. Prof. Dr. Hans-Arno Jacobsen
2. Prof. Dr. Florian Matthes
3. Assoc. Prof. Christopher G. Brinton, Ph.D.

Die Dissertation wurde am 14.04.2025 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 26.10.2025 angenommen.

Abstract

This thesis examines the challenges and opportunities of implementing Federated Learning (FL) systems on embedded hardware, with a focus on scalability, efficiency, and regulatory compliance. Through the development of a novel benchmarking framework called FLEdge, we systematically evaluate the performance limitations of FL clients on edge devices, revealing significant memory bottlenecks that result in processing times up to four times longer than modern data center GPUs. We demonstrate that, contrary to established practices in high-performance computing, scaling batch sizes on embedded devices does not improve computational efficiency due to hardware-specific constraints. Further investigation into scaling FL systems to accommodate Large Language Models identifies critical performance challenges at the network edge. Our experiments with the FLAN-T5 model family on NVIDIA Jetson AGX Orin devices establish that memory bandwidth limitations significantly impact computational efficiency. We introduce an energy efficiency metric as a practical real-time measure for assessing computational effectiveness without requiring detailed hardware specifications, showing strong correlation with traditional Model FLOP Utilization metrics across tested models. Our systematic analysis of computational and communication efficiency in FL applications reveals a significant disconnect between these domains. While computational efficiency methods for foundation models exist, communication efficiency techniques predominantly target full-model training, creating a methodological misalignment that grows more challenging as models scale to billions of parameters. Parameter-Efficient Fine-Tuning techniques show promise for reducing both computational and communication intensity but exhibit increased sensitivity to data heterogeneity. Finally, our interdisciplinary analysis positions FL advantageously within emerging regulatory frameworks, particularly the EU AI Act. By maintaining data locality and minimizing raw data transfers, FL inherently satisfies requirements for data protection by design. However, we identify significant gaps between current technical capabilities and regulatory expectations regarding energy efficiency, data quality assurance, and responsibility allocation among system participants. This thesis provides insights for making FL systems more efficient, scalable, and regulatory-compliant, establishing a clear path for future research addressing legal priorities while improving training efficiency of FL clients on the network edge and optimizing communication for large deep learning models.

Zusammenfassung

In dieser Arbeit werden die Herausforderungen und Möglichkeiten der Implementierung von Federated Learning-Systemen (FL) auf eingebetteter Hardware untersucht, wobei der Schwerpunkt auf Skalierbarkeit, Effizienz und Einhaltung von rechtlichen Vorschriften liegt. Durch die Entwicklung eines neuartigen Benchmarking-Frameworks, FLEdge, evaluieren wir systematisch die Leistungsbeschränkungen von FL-Clients auf Edge-Geräten und decken dabei erhebliche Speicherengpässe auf, die zu Verarbeitungszeiten führen, die bis zu viermal länger sind als bei modernen GPUs in Rechenzentren. Wir zeigen, dass die Skalierung von Minibatches auf eingebetteten Geräten im Gegensatz zu etablierten Praktiken im High-Performance-Computing aufgrund hardwarespezifischer Einschränkungen nicht zu einer Verbesserung der Berechnungseffizienz führt. Weitere Untersuchungen zur Skalierung von FL-Systemen für größere Sprachmodelle zeigen kritische Leistungsprobleme an der Netzwerk-Edge auf. Unsere Experimente mit der FLAN-T5-Modellfamilie auf NVIDIA Jetson AGX Orin-Geräten zeigen, dass die begrenzte Speicherbandbreite die Recheneffizienz erheblich beeinträchtigt. Wir führen eine Energieeffizienzmetrik als praktisches Echtzeitmaß zur Bewertung der Recheneffizienz ein, ohne dass detaillierte Hardwarespezifikationen erforderlich sind, und zeigen bei allen getesteten Modellen eine starke Korrelation mit der weit verbreiteten Model-FLOP-Utilization Metrik. Unsere systematische Analyse der Rechen- und Kommunikationseffizienz in FL-Anwendungen zeigt eine signifikante Diskrepanz zwischen diesen Bereichen.

Prinzipiell ist die Skalierung der von Modellen auf mehrere Milliarden Parameter durch Parameter Efficient Fine-Tuning Techniken vielversprechend, um sowohl die Rechen- als auch die Kommunikationsintensität zu reduzieren, jedoch reagieren diese Techniken empfindlich auf Datenheterogenität. Schließlich positioniert unsere interdisziplinäre Analyse FL vorteilhaft innerhalb neuer gesetzlicher Rahmenbedingungen, insbesondere dem europäischen KI Gesetz. Durch die Beibehaltung der Datenlokalität und die Minimierung des Rohdatentransfers erfüllt FL von Natur aus die Anforderungen des Datenschutzes durch Design. Wir stellen jedoch fest, dass zwischen den derzeitigen technischen Möglichkeiten und den Erwartungen der Regulierungsbehörden in Bezug auf Energieeffizienz, Datenqualitätssicherung und Verantwortungszuweisung zwischen allen Parteien in einem FL-System erhebliche Lücken bestehen. Diese Arbeit liefert Erkenntnisse, um FL-Systeme insgesamt effizienter und gesetzeskonform zu gestalten, und zeigt einen klaren Weg für künftige Forschungen auf, die sowohl die Energieeffizienz als auch regulatorische Compliance von FL-Anwendungen weiter verbessern.

Acknowledgements

I am deeply grateful for the support, mentoring, and guidance of my advisor, Prof. Hans-Arno Jacobsen. Your constant encouragement and constructive feedback have enabled me to grow academically and personally. Thank you for providing me the freedom to explore a broad range of ideas in my research and for ample opportunities to engage with the research community, be it at conferences, workshops, or joint research projects.

I owe a heartfelt thank you to Prof. Shiqiang Wang, who has become a close mentor and friend along the way. Your invaluable advice, leadership by example, and our hour-long discussions have made the difference in my research journey. You changed my perspective on research for good and have greatly motivated me to go above and beyond. Thank you!

I would like to thank Prof. Florian Matthes and Prof. Christopher G. Brinton for agreeing to serve on my examination committee. Thank you, Prof. Rüdiger Westermann, for chairing the committee.

Thank you to my close collaborators, Prof. Ruben Mayer and Dr. Alexander Erben, who provided invaluable support for my research projects, especially in the beginning. Your feedback and help in designing experiments have had a substantial impact, and without you, the projects would not have gotten where they are. Also, Alex, thank you for the endless hours we spent together developing and maintaining our compute cluster.

I also want to thank all the collaborators from academia and industry I met along the way and cannot all mention individually. I had great fun working on our joint research projects. Thank you for sharing parts of the journey with me.

Thank you to my family for their incredible support, my parents, Alexandra and Herbert, and my brother Johannes. My journey would certainly not have been possible without you. I guess, this is another major milestone in the marathon of life...
I am forever grateful for all you have done for me!

Finally, thank you to my fantastic wife, Lisa, for your unwavering support when I was chasing deadline after deadline. I am forever grateful for your patience and constant encouragement along the journey, especially during the research internships abroad and the many long days and weekends filled with work. You mean the world to me. I love you!

Contents

List of Publications

This publication-based thesis contains three *core publications* that have all been peer-reviewed and are published in proceedings. Further, the thesis includes two *non-core publications*, both of which have been accepted to peer-reviewed workshops. This thesis includes excerpts, summaries, and key results from all publications without explicit reference. The published versions and the respective copyright licenses of all core and non-core papers can be found in Appendices A to E.

Core Paper 1 (sole first authorship) Herbert Woisetschläger, Alexander Erben, Ruben Mayer, Shiqiang Wang, and Hans-Arno Jacobsen. "FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems." In: *Proceedings of the 25th International Middleware Conference*. Middleware '24. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 88–102. ISBN: 9798400706233. DOI: 10.1145/3652892.3700751. URL: https://doi.org/10.1145/3652892.3700751

Core Paper 2 (sole first authorship) Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly." In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. Santiago, Chile: Association for Computing Machinery, 2024, pp. 39–50. ISBN: 9798400706110. DOI: 10.1145/3650203.3663331. URL: https://doi.org/10.1145/3650203.3663331

Core Paper 3 (sole first authorship) Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "A survey on efficient federated learning methods for foundation model training." In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. IJCAI '24. Jeju, Korea, 2024, pp. 8317–8325. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/919. URL: https://doi.org/10.24963/ijcai.2024/919

Non-core Paper 1 (sole first authorship) Herbert Woisetschläger, Alexander Erben, Bill Marino, Shiqiang Wang, Nicholas D. Lane, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning Priorities Under the European Union Artificial Intelligence Act." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2402.05968. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/48.pdf

Non-core Paper 2 (shared first authorship) Herbert Woisetschläger*, Simon Mertel*, Christoph Krönke, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2407.08105. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/16.pdf (* indicates shared first authorship)

In addition, the author has contributed significantly to the following publications (in reverse chronological order) that are not part of this thesis and must not be evaluated.

Daouda Sow, Herbert Woisetschläger, Saikiran Bulusu, Shiqiang Wang, Hans-Arno Jacobsen, and Yingbin Liang. "Dynamic Loss-Based Sample Reweighting for Improved Large Language Model Pretraining." In: *The Thirteenth International Conference on Learning Representations*. ICLR'25. 2025. URL: https://openreview.net/forum?id=gU4ZgQNsOC

Hajar Emami Gohari, Swanand Ravindra Kadhe, Syed Yousaf Shah, Constantin Adam, Abdulhamid Adebayo, Praneet Adusumilli, Farhan Ahmed, Nathalie Baracaldo Angel, Santosh Borse, Yuan-Chi Chang, Xuan-Hong Dang, Nirmit Desai, Ravital Eres, Ran Iwamoto, Alexei Karve, Yan Koyfman, Wei-Han Lee, Changchang Liu, Boris Lublinsky, Takuyo Ohko, Pablo Pesce, Maroun Touma, Shiqiang Wang, Shalisha Witherspoon, Herbert Woisetschläger, David Wood, Kun-Lung Wu, Issei Yoshida, Syed Zawad, Petros Zerfos, Yi Zhou, and Bishwaranjan Bhattacharjee. "GneissWeb: Preparing High Quality Data for LLMs at Scale." Feb. 2025. URL: https://huggingface.co/datasets/ibm-granite/GneissWeb

Ryan Zhang, Herbert Woisetschläger, Shiqiang Wang, and Hans Arno Jacobsen. "MESS+: Energy-Optimal Inferencing in Language Model Zoos with Service Level Guarantees." In: *Workshop on Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning in Conjunction with NeurIPS'24*. 2024. URL: https://openreview.net/forum?id=OoReeQpwmW

Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschläger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. "A survey on dataset distillation: approaches, applications and future directions." In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI '23. Macao, P.R.China, 2023. ISBN: 978-1-956792-03-4. DOI: 10.24963/ijcai.2023/741. URL: https://doi.org/10.24963/ijcai.2023/741

Chapter 1

Introduction

The emergence of federated learning (FL) represents a significant paradigm shift in contemporary machine learning (ML) methodologies, addressing critical challenges inherent in modern data ecosystems [10]. FL is particularly effective in scenarios where traditional centralized learning frameworks encounter limitations due to data control and regulatory requirements [11, 12, 13].

Recent developments in legislation (e.g., the European Union AI Act) and heightened awareness of privacy or copyright concerns have necessitated fundamental modifications to established machine learning practices [14]. FL presents a methodological framework that enables collaborative model development while maintaining strict data localization, thereby addressing the inherent tensions between model performance and data ownership [15, 16]. FL enables data owners to retain full control of their data and thus leaves the decision of what data to use during training on a granular level to the data owner [17, 18, 19]. This lifts barriers for use cases where data sensitivity and trust have previously been key challenges.

For instance, FL enables model training on sensitive data between entities and jurisdictions since raw training data never leaves the owner's premises. This is advantageous in three ways. First, access to domain-specific data becomes easier, which enables training over larger amounts of data and likely results in a higher model quality [20, 21, 22, 23]. Second,

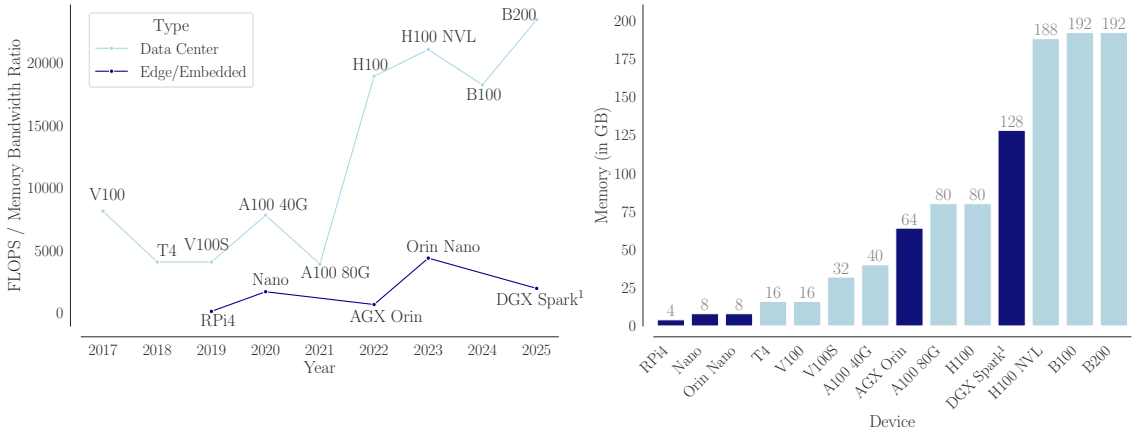


Figure 1.1: The computational capabilities (measured as the FLOPS/Memory Bandwidth Ratio) of embedded hardware are catching up to data center devices, enabling the training of state-of-the-art deep learning models in the billion parameter regime[1].

learning across institutions is simplified. For instance, the European Health Data Space (EHDS), a new bill recently adopted by the European Council, requires EU member states to use electronic patient data and make available the data in a secure and anonymous way to enhance data-driven research [24]. In the case of Germany, this requires the different healthcare insurers to make their patients' information accessible. Third, multi-national entities can learn models across jurisdictions since sensitive training data is never transferred between regions. This is particularly useful to bolster compliance with data protection regulations such as GDPR [25]. It can also reduce data fragmentation in large organizations by making data accessible without the need to see and directly work with raw input data.

In addition, FL excels at harnessing spare computing capacity by users to contribute their computing resources flexibly based on their current state and availability [26, 27]. For example, hardware-accelerated embedded devices (e.g., Nvidia Jetson AGX Orin) are capable of performing model training when they have spare computing capacity, sufficient network bandwidth, and are not actively being used for higher priority tasks (Figure 1.1). This opportunistic scheduling enables the system to tap into otherwise idle computing power without impacting device performance or user experience. FL can automatically adjust training schedules based on a wide variety of environmental factors ensuring efficient resource utilization [28, 29].

[1]Nvidia DGX Spark statistics are based on Nvidia's product announcement (April 2025) [30].

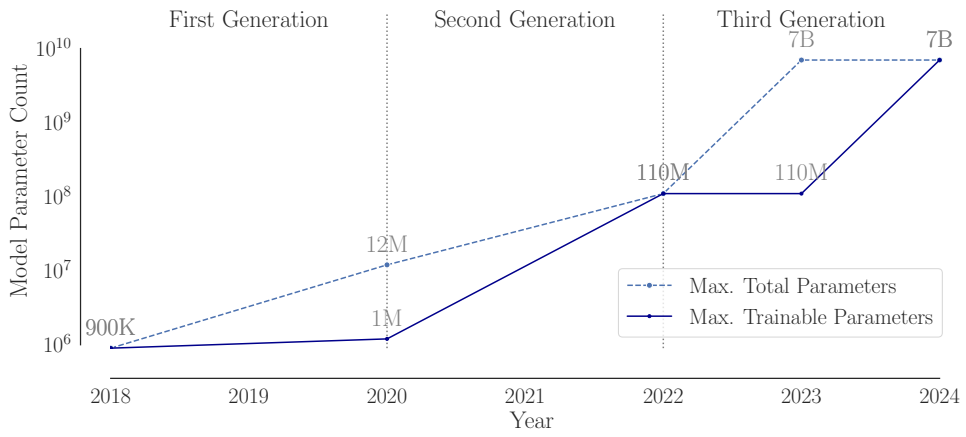


Figure 1.2: The size of models trained with FL has grown three orders of magnitude within seven years. FL models are following a similar growth trajectory as their counterparts trained with traditional centralized approaches. Yet, FL still exhibits significant shortcomings with regard to overall computational and communication efficiency.

1.1 Motivation

While FL offers advantages in leveraging distributed data sources, enabling learning from sensitive domain-specific data, and utilizing decentralized computing resources, it faces substantial operational challenges that warrant careful consideration. These challenges primarily manifest in two critical areas: the computational and communication overheads inherent in the distributed learning process and the need to manage hardly predictable client behavior that can significantly compromise training reliability. Trends indicate that FL applications will incorporate increasingly larger models with expanding parameter counts and enhanced capabilities, increasing the cost of failure during training.

This is evident in the evolution of FL applications (Figure 1.2). The first generation of FL applications included small models ($<$ 1M parameters) that could be trained quickly and without the need for extensive hardware acceleration. Yet, the use cases were limited to simplistic applications like character recognition in images or next-character prediction for text [31]. In the transition between first and second-generation applications, the parameter count of models grew by an order of magnitude to approx. 12M with the deployment of ResNet [32, 33]. The second generation was mainly characterized by transformer models [34] as the release of GPT-3 [35] has drawn significant public attention and research interest at the time. Again, the total model parameter count of the

largest models that were trained with FL grew by 10×. Yet, the most notable difference compared to the first generation is the number of trainable parameters grew by two orders of magnitude. The same is true for the third and current generation, where FL is used to train state-of-the-art LLM architectures with up to 7B parameters [36, 37, 38]. As models grow, the need for better computational and communication efficiency grows.

At the same time, global legislation on AI creates new directives and guardrails on how to train, evaluate, and deploy deep learning models. This can introduce complexities in addition to already existing factors that hinder practical adoption of FL today [39]. Overall, the legal and technical challenges create a fundament for interdisciplinary FL research and how legislation can help drive practical adoption of FL.

1.2 Problem Statement

As FL continues to evolve and expand into new domains, several critical challenges emerge at the intersection of system design, scalability, efficiency, and regulatory compliance. This thesis addresses four interconnected problems in the FL landscape: First, we examine the fundamental gap in understanding how FL clients perform on edge devices, moving beyond traditional server-centric benchmarks to assess real-world deployment scenarios. Second, we investigate the complexities of scaling FL systems to accommodate the growing demands of large language models, particularly when dealing with resource-constrained edge devices. Third, we explore the dual challenge of optimizing both computational and communication efficiency in FL applications, especially when working with billion-parameter models that strain existing frameworks. Finally, we analyze the technical implications of emerging AI regulations, using the EU AI Act as a framework to understand how FL can help build regulatory compliant systems while maintaining operational effectiveness. Together, these challenges raise the question *how we can design scalable, efficient, and regulatory-compliant FL systems.*

1.2.1 Client Behavior in Federated Learning Applications

While FL has become an established middleware abstraction for distributed and data-ownership-oriented deep learning, existing benchmarks primarily target server-grade hardware and rely on simulations [31, 33, 40, 41]. As such, they neglect the characteristics of FL clients that are often deployed on the network edge. Devices on the network edge are typically characterized by diverse hardware capabilities, lower reliability compared to data center hardware, varying network quality, and energy constraints [29, 42]. Additionally, FL workloads in these environments are prone to a high degree of data heterogeneity [43, 44]. This creates a complex set of challenges and interdependency between the various factors. Overall, this increases the complexity when designing and deploying FL applications. It is also a potential limiting factor for broader adoption of FL in practice, raising the question *how well state-of-the-art FL workloads respond to deployment in edge computing systems.*

1.2.2 Training Large Models in Federated Learning Systems

With the growing adoption of LLMs in both academia and industry, there is an increasing need to fine-tune these models on domain-specific data to improve their performance on downstream tasks. While FL offers a promising approach to accessing distributed data sources for model fine-tuning, a major challenge arises from the fact that this data often resides on edge devices with significantly constrained resources. The computational capabilities of edge devices are orders of magnitude lower than data center hardware, with the latest edge devices like the NVIDIA Jetson AGX Orin providing only 13% of the theoretical computational power of a state-of-the-art data center GPU [45, 46]. This limitation is particularly challenging for LLM fine-tuning, as these models are inherently more challenging to train than smaller models due to their size and complexity, with higher risks of gradient explosion or vanishing during the training process. Additionally, the memory bandwidth constraints on edge devices create significant bottlenecks for operations that are critical to the training process. This raises the question: *How can we effectively enable federated fine-tuning of LLMs on resource-constrained edge devices, and what are the key factors that influence the efficiency of such systems?*

1.2.3 Computational and Communication Efficiency

While FL could enable access to diverse, sensitive datasets needed for LLM fine-tuning, current FL approaches are primarily designed for small models and full model training, creating significant computational and communication bottlenecks when applied to billion-parameter models [47]. This efficiency challenge manifests in both computational and communication dimensions [48, 49]. The computational inefficiency arises from the substantial resources required for training and fine-tuning LLMs across distributed clients in FL systems [50, 51]. Simultaneously, significant bandwidth and network demands of transmitting large model updates between clients and servers result in communication bottlenecks [52]. Existing FL frameworks typically incorporate methods for efficient model aggregation but do not provide solutions for improving the computational and communication efficiency of a training process, especially when looking at edge deployments [53]. Since these challenges are interlinked when working with LLMs, they must be addressed jointly to enable the practical deployment in FL settings. This leads to the research question: *How can computational and communication efficiency be jointly optimized to enable the practical deployment of LLMs in FL systems?*

1.2.4 Emerging AI Regulation in the European Union

System efficiency is at the core of globally emerging AI regulation. At a global level, organizations like the OECD have established principles and guidelines for responsible AI development, creating a foundation for more specific regulatory frameworks worldwide [14, 54, 55]. However, the practical implementation of these principles into binding legislation creates significant technical and operational challenges for AI development [56]. Often this also involves extensive compliance and legal clearance processes [57]. The European Union AI Act, as the first comprehensive AI legislation globally, serves as a precedent and blueprint for how these broader principles are being translated into concrete regulatory requirements. There is an inherent misalignment between traditional AI development approaches and emerging regulatory requirements. Current AI development methods, which rely heavily on centralized data collection and processing, face substantial challenges in meeting new regulatory standards around data

governance, privacy protection, security requirements, and environmental sustainability. This creates a fundamental tension between regulatory compliance and effective AI system development, particularly for high-risk applications that face the most stringent oversight [14]. Overall, this yields the question: *How can AI systems be developed and deployed to achieve compliance with emerging global AI regulations, using the EU AI Act as a framework, while maintaining technical effectiveness and operational feasibility?* This question is particularly relevant as other jurisdictions begin developing their own AI regulations based on similar principles, making the EU AI Act a critical case study for understanding how technical solutions can address regulatory requirements that are likely to become more common globally [58, 59]. The problem is time-sensitive given the implementation timeline of the EU AI Act and the growing momentum for AI regulation worldwide, requiring solutions that can bridge the gap between current development and deployment practices and emerging regulatory frameworks.

1.3 Approach

Our approach spans four main areas: (1) FLEdge, a *benchmarking framework* that systematically assesses client behavior, communication patterns, energy efficiency, and hardware diversity in edge computing environments; (2) a methodology for *scaling FL systems to train large language models*, focusing on computational efficiency, energy consumption, communication costs, and model performance through metrics like Model FLOP Utilization and energy efficiency ratios; (3) a *structured analysis of computational and communication efficiency in FL applications*, examining the intersection of parameter-efficient fine-tuning (PEFT), model compression, and distributed optimization techniques; and (4) an *interdisciplinary framework for ensuring regulatory compliance of FL systems*, particularly with respect to the EU AI Act, through careful consideration of data governance, energy efficiency, and quality management.

1.3.1 Benchmarking Behavior and Resource Efficiency of FL Clients

We begin with our benchmarking framework – FLEdge – that evaluates FL workloads in edge computing environments through the lens of four fundamental dimensions: client behavior, communication efficiency, energy efficiency, and hardware diversity. Our approach complements existing FL benchmarks by shifting focus from simulation-based evaluation to systematic assessment of real-world deployment challenges. The framework builds upon established FL infrastructure while introducing new mechanisms for comprehensive system evaluation.

The conceptual foundation of our benchmark rests on modeling and measuring key system characteristics that impact FL performance at the edge. For client behavior, we incorporate probabilistic modeling of device reliability through independent binomial distribution, enabling systematic evaluation of client dropouts and their interaction with differential privacy guarantees. The privacy aspect is handled through an adaptive user-level differential privacy mechanism that accounts for varying client participation. Communication patterns are evaluated through a granularity-based approach that quantifies the relationship between computation and communication costs, considering both wired and wireless network scenarios. This is complemented by a per-bit communication cost model that estimates energy consumption across network hops. Energy efficiency is assessed through a combination of direct hardware measurements and theoretical modeling of network energy consumption, providing insights into system-wide energy characteristics. We measure this through throughput in samples per second divided by power draw, enabling non-intrusive detection of computational bottlenecks.

Our approach unifies these concepts through a systematic evaluation pipeline that considers both local device capabilities and global system behavior. For hardware assessment, we employ micro-benchmarking techniques that analyze training step times, memory bandwidth utilization, and the impact of different GPU architectures. The framework employs standardized metrics across different hardware platforms, enabling comparative analysis of FL workload behavior through controlled network conditions and client participation patterns. This approach allows for the evaluation of both system-level properties like client reliability and hardware-specific characteristics like computational efficiency, providing a comprehensive view of FL workload behavior in edge environments.

The evaluation pipeline supports various FL strategies and is designed to be extensible for new hardware platforms and workload types, making it a valuable tool for understanding the practical implications of deploying FL systems at the edge.

FLEdge is built upon the Flower Framework [60], a popular library for implementing and orchestrating efficient FL applications. Thus, FLEdge can be easily deployed and extended in the same way as Flower, making it intuitive for researchers and practitioners to use.

1.3.2 Scaling FL Systems to Train Large Language Models

We develop a systematic approach to evaluate FL at the network edge, focusing particularly on the intersection of LLMs and resource-constrained environments. Our methodology is built on four fundamental analytical pillars: computational efficiency, which examines hardware utilization and processing capabilities through metrics like Model FLOP Utilization (MFU); energy efficiency, which quantifies the relationship between computational throughput and power consumption through our proposed energy efficiency metric η_e; communication efficiency which analyzes the balance between computation and communication costs using granularity measurements; and model performance which evaluates convergence and quality metrics. This framework is designed to provide a comprehensive understanding of the challenges and constraints in edge-based FL systems, taking into account emerging regulatory requirements and practical deployment considerations through real-world hardware and network configurations.

The core of our methodology centers on comparative analysis between edge and data center environments, utilizing well-established metrics that we complement with new measures to account for the fundamentally different properties of embedded hardware. We develop our evaluation strategy around the concept of performance ratios - comparing computational throughput against theoretical hardware limits through MFU analysis, η_e measurements, and communication overhead against computation time through granularity assessments. This approach allows us to establish quantifiable benchmarks that can be consistently applied across different hardware configurations and model architectures. We pay particular attention to memory bandwidth constraints, network communication patterns, and energy consumption profiles, as these factors significantly

impact the feasibility of edge deployments. Our evaluation framework incorporates both theoretical performance limits and practical operational constraints, allowing us to identify bottlenecks and optimization opportunities across the entire system stack.

Our experimental framework is structured to evaluate these metrics under realistic FL conditions, incorporating parameter-efficient fine-tuning techniques like low-rank adapters (LoRA) [61], varying client participation patterns through Dirichlet distribution sampling, and different network connectivity scenarios, including both wireless and wired configurations. We design the study to examine both the individual components of edge-based FL systems and their interactions, providing insights into system-level behaviors and constraints. This multi-faceted approach enables us to create a comprehensive evaluation approach that can be applied to assess the practical viability of edge-based FL deployments. The framework accommodates various federated optimization strategies, from simple averaging to adaptive techniques, and considers the impact of data heterogeneity and communication patterns on system performance. By incorporating both system-level metrics and model-specific measurements, our approach provides a holistic view of edge FL systems that account for computational, communication, and energy constraints while maintaining practical applicability.

1.3.3 Efficiency in FL Systems for Large Language Models

We introduce a structured analytical approach to understand the convergence of computational and communication efficiency in FL systems, particularly for LLMs. Our approach is built upon a novel taxonomic framework that maps the relationship between efficiency methods (e.g., PEFT, prompt tuning, instruction tuning) and their practical implementations in distributed environments. This structure examines how traditionally distinct optimization approaches intersect and complement each other, focusing on the interplay between model compression techniques, gradient quantization, and adaptive training strategies in modern distributed learning environments.

The core of our approach decomposes efficiency into computational and communication dimensions through categorical analysis. The computational dimension encompasses full model training, parameter-efficient techniques, and various tuning approaches, while the communication dimension focuses on model pruning and compression techniques, including quantization, sparsification, and gradient projection. We establish clear criteria

for classification and evaluation based on key metrics such as parameter reduction ratios, communication overhead, and computational complexity. Each dimension is examined through theoretical foundations and practical implementations.

To comprehensively address computational and communication efficiency in FL applications, we investigate four key areas: (1) existing methods for achieving computational efficiency (e.g., LoRA, BitFit, and adapter-based approaches) [61, 62, 63] and communication efficiency (e.g., FedPAQ, FedKSeed) [64, 65] in FL systems working with LLMs and their limitations; (2) quantifiable trade-offs between different efficiency optimization approaches, examining metrics like model accuracy, communication bandwidth, training time, and resource utilization; (3) architectural readiness of current FL frameworks (e.g., NVIDIA FLARE, FederatedScope, Flower) to support efficient LLM training and fine-tuning, particularly focusing on their ability to handle large-scale models and implement advanced efficiency techniques; and (4) methodological gaps preventing efficient scaling of LLMs in FL systems, such as the challenges in maintaining model quality while reducing communication overhead and computational requirements. This analysis considers both theoretical foundations in distributed optimization and real-world implementation challenges while identifying critical research gaps that must be addressed to make FL with LLMs practically feasible at scale.

1.3.4 Regulatory Compliant FL Applications in High-Risk Regimes

Emerging AI regulation prioritizes resource efficiency and sets standards for AI application monitoring [14, 66]. We introduce a novel interdisciplinary framework for analyzing the intersection of FL and regulatory compliance, specifically focusing on the EU AI Act. The analytical framework bridges legal requirements and technical capabilities through a dual-lens approach: examining both the regulatory obligations outlined in the AI Act and the inherent characteristics of FL systems. We carefully analyze both Articles and Recitals from the AI Act to distinguish between binding legal requirements and interpretative guidance, mapping both to the technical capabilities of FL. With our interdisciplinary approach we outline how FL can bolster legal compliance of AI applications and what future research is needed to overcome existing hurdles. Our approach acknowledges the unique challenges of FL systems, where data never leaves its source and model training occurs across distributed clients.

The framework divides the analysis into three conceptual pillars: data governance, energy efficiency, and robustness and quality management. For each pillar, we developed evaluation criteria that map regulatory requirements to technical capabilities. The data governance pillar examines privacy preservation mechanisms, including differential privacy, secure multi-party computation, and homomorphic encryption, alongside bias mitigation approaches in distributed settings. The energy efficiency pillar considers both computational and communication costs across distributed systems, analyzing the energy impact of client-server architectures, model parameter transmission, and privacy-preserving computation techniques. The robustness pillar evaluates quality assurance and monitoring capabilities in privacy-preserving contexts, including validation strategies, model performance tracking, and bias detection without direct data access. This structured approach allows for the systematic identification of both alignment opportunities and compliance challenges while considering the inherent trade-offs between privacy, efficiency, and model quality.

To operationalize our analysis framework, we combine quantitative experimentation with qualitative regulatory evaluations. The quantitative component uses representative high-risk applications to explore technical feasibility, examining aspects such as computational overhead, communication costs, and the impact of privacy-preserving mechanisms on model performance. We developed comprehensive evaluation metrics that consider both client-side and server-side resource utilization, privacy guarantees, and model quality indicators. The qualitative component examines non-measurable system characteristics through careful analysis of legal requirements, including aspects like data lineage, bias prevention capabilities, and compliance with fundamental rights. By integrating these perspectives, our framework provides a comprehensive approach to evaluating privacy-preserving machine learning systems in regulatory contexts. Our framework considers both current technical capabilities and future requirements, making it adaptable to emerging privacy-preserving techniques and evolving regulatory landscapes. This framework can be extended to assess other distributed learning approaches and adapted as regulatory requirements evolve.

1.4 Contributions

This thesis is based on five papers that make contributions to benchmarking FL clients, accommodating large deep learning models in FL systems, showing computational and communication efficiency as a joint optimization goal for LLMs, and designing regulatory-compliant FL applications. Overall, the objective is to improve the scalability and energy efficiency of FL systems involving embedded computing hardware and wide-area networks. Individually, each paper contribution can be summarized as follows:

1. We introduce a comprehensive benchmarking framework focused on FL clients at the network edge. Through extensive experimentation, we demonstrate that state-of-the-art embedded AI accelerators face significant memory bottlenecks. Paired with the low reliability of FL clients on the network edge due to timeouts (e.g., network interruption) and the need for private computing techniques to keep model updates secure, this can yield significantly prolonged processing times. Our detailed analysis of hardware heterogeneity demonstrates that contrary to established practices, scaling batch sizes on embedded devices does not improve computational efficiency. Profiling small transformer models on embedded hardware reveals that CPU-GPU data movement and matrix multiplication operations can take up to 32× longer than on data center GPUs. From a practical perspective, we extended the Flower FL library with modules for controlling client behavior, implementing adaptive differential privacy, emulating network conditions, and monitoring computational efficiency on dedicated edge devices. Overall, our findings serve as guidance for optimizing FL workload deployment in edge computing systems and provide a basis for building more efficient FL applications.
2. We examine the challenges and opportunities of implementing FL for fine-tuning LLMs on edge devices, focusing on three key aspects: computational efficiency, energy consumption, and communication costs. Through extensive experimentation with the FLAN-T5 encoder-decoder model family on NVIDIA Jetson AGX Orin devices, we analyze hardware limitations of edge computing for deep learning workloads, propose new metrics for monitoring system efficiency in federated settings, and evaluate various optimization strategies to improve training performance, i.e., enhancing sample efficiency during training resulting in lower communication cost. Our work provides a comprehensive framework for understanding the practical

considerations of FL system deployment at the network edge.

3. We present a comprehensive analysis of the challenges and opportunities in integrating LLMs with FL systems. In our survey, we develop a novel taxonomy that systematically examines the interplay between computational and communication efficiency, revealing critical gaps in current approaches that must be addressed. Through rigorous evaluation of existing approaches, we provide detailed insights into the trade-offs and limitations of pre-training and fine-tuning LLMs in federated environments. We outline essential research directions necessary for practical implementation, with particular emphasis on evaluation frameworks, privacy considerations, and efficient communication protocols. Together, these contributions establish a basis for future research in FL with LLMs.
4. We examine how the EU AI Act will impact FL and identify necessary changes to make FL viable for regulatory compliance. Through our interdisciplinary legal and technical analysis, we evaluate the current capabilities and limitations of FL in meeting the AI Act requirements around data governance, privacy, energy efficiency, and model quality. While we find that the data-ownership-oriented architecture of FL provides inherent advantages, our analysis reveals significant gaps between current FL approaches and regulatory requirements, particularly in areas like energy efficiency, data quality monitoring, and lifecycle management. We outline key research priorities to address these gaps and argue that with appropriate focus on these priorities, FL could become a leading approach for building AI systems that comply with the AI Act and beyond.

1.5 Structure

This thesis is organized as follows. In Chapter 2, we outline our methodology and introduce relevant background. We provide brief summaries of each publication that comprises this thesis in Chapter 3. Each summary provides an overview of the paper's significant findings and research contributions, highlighting the thesis author's individual role in advancing the work. Thereafter, we discuss the main finding and limitations of this thesis in Chapter 4. We conclude in Chapter 5. The papers discussed in this thesis are available in full length in Appendices A to E.

Chapter 2

Methodology

2.1 Background

2.1.1 Federated Learning

In traditional machine learning, we train models on centralized datasets. However, FL introduces a paradigm shift by enabling model training across decentralized devices while keeping data private [10]. This approach represents a fundamental change in how we think about distributed machine learning, moving from model-centricity to user-centricity.

As this thesis investigates efficiency in FL systems for large deep learning models at the network edge, it is imperative to introduce a set of prerequisites. Those include adaptive optimization, root causes for computational and communication bottlenecks, and data heterogeneity. Adaptive optimization in FL tackles the challenge of training models across diverse devices by dynamically adjusting optimization parameters. This approach ensures robust model convergence despite varying device capabilities and participation patterns. The system must continuously adapt to changing conditions while maintaining training stability.

Computational inefficiencies emerge as a critical challenge due to the heterogeneous nature of participating devices. With varying processing capabilities, memory constraints, and energy limitations, these differences significantly impact the overall training process and necessitate careful consideration of resource allocation.

Communication overhead represents a substantial bottleneck in FL systems. The frequent exchange of model updates between devices and the central server demands considerable bandwidth, particularly challenging in environments with limited connectivity or when working with large model architectures.

Data heterogeneity introduces unique complications as the distribution of data across devices typically follows non-IID patterns. This statistical diversity can impair model convergence and performance, requiring specialized approaches to handle varying data distributions while maintaining model effectiveness.

Preliminaries

Consider a system with K clients, where each client k has its local dataset D_k of size n_k. The total number of samples across all clients is $N = \sum_{k=1}^{K} n_k$. The goal is to find model parameters w that minimize the global objective function:

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{N} F_k(w), \tag{2.1.1}$$

where $F_k(w)$ is the local objective function for client k, $F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} \ell\ (w; x_i, y_i) + \lambda\ R(w)$. Here, $\ell\ (\cdot)$ is the loss function, (x_i, y_i) are data samples, and $R(w)$ is a regularization term with parameter λ. The optimization process [10] typically follows these steps in each round t:

- The server broadcasts the current global model parameters w^t to a subset of clients $S_t \subseteq [K]$.
- Each selected client $k \in S_t$ performs local training for $\mathcal{B}$ steps using mini-batch SGD: $w_k^{t+1} = w^t - \eta \sum_{b=1}^{B} \nabla F_k(w^t; \mathcal{B}_b)$ where η is the learning rate and $\mathcal{B}_b$ represents the b-th mini-batch.

- The server aggregates the updated models using weighted averaging, typically in the form of federated averaging (FedAvg):

$$w^{t+1} = \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} w_k^{t+1}. \tag{2.1.2}$$

The convergence rate of this process depends on several factors. The degree of non-IID data distribution across clients, quantified by $\gamma = \max_{k,j} |\nabla F_k(w) - \nabla F_j(w)|$. It is also determined by the fraction of clients selected per round, $C = |S_t|/K.$, client availability, and dropout rates. Communication rounds T and $\mathcal{B}$ local steps have substantial effects on the training performance as well [31, 67].

Non-IID Data Distribution. Local objectives F_k deviate from global objective F due to heterogeneous data, measured by gradient diversity:

$$\zeta = \frac{|\sum_{k=1}^{K} p_k \nabla F_k(w)|^2}{\sum_{k=1}^{K} p_k |\nabla F_k(w)|^2} \tag{2.1.3}$$

The gradient diversity metric $\zeta \in [0, 1]$ quantifies the alignment between local and global optimization directions. When $\zeta = 1$, all local gradients are perfectly aligned, indicating IID-like conditions. As ζ decreases toward 0, local updates increasingly conflict, signifying statistical heterogeneity and hindering the model convergence process [68].

Communication Efficiency. Limited bandwidth and high latency between server and clients constrain the model size and update frequency [69]. Modern approaches address these constraints through adaptive compression techniques [70], asynchronous aggregation protocols [27, 71], and hierarchical communication topologies [72].

System Heterogeneity. Varying computational capabilities and network conditions across clients affect training time and reliability. We can quantify system heterogeneity as the interplay of computational capacity of a device relative to a benchmark device $c_k = \frac{\text{FLOPS}_k}{\text{FLOPS}_{\text{ref}}}$ [73], network bandwidth over time relative to the overall average in the FL system $z_k(t) = \frac{Z_k(t)}{Z_{\text{ref}}}$ [74], and the client reliability across training rounds $r_k = \text{P}(\text{completion} \mid \text{selected})$ [75].

Adaptive Optimization in FL

Adaptive optimization can help reduce the variance in client model updates that may result from one of the key challenges above by taking the gradient history into account. This can improve the training effectiveness, enable training of bigger models, and, overall, reduce the training time.

On the client side, adaptive optimization extends beyond basic stochastic gradient descent (SGD) by incorporating momentum and variance adaptation mechanisms, drawing inspiration from the Adam optimizer used in centralized learning. This approach maintains exponentially decaying estimates of both the first and second moments of the gradients, β_1 and β_2, respectively. This allows for more nuanced parameter updates [76]. The first moment estimate m_k^t tracks the mean of the gradients, acting as a momentum term that helps accelerate training in consistent directions while dampening oscillations [77]. This estimate is updated using a decay rate β_1 that determines how much historical information is retained:

$$m_k^t = \beta_1 m_k^{t-1} + (1 - \beta_1)\nabla F_k(w_k^t). \tag{2.1.4}$$

Simultaneously, the algorithm maintains a second moment estimate v_k^t that captures the variance of the gradients, enabling adaptive scaling of updates based on the historical magnitude of each parameter's gradients [78]:

$$v_k^t = \beta_2 v_k^{t-1} + (1 - \beta_2)(\nabla F_k(w_k^t))^2. \tag{2.1.5}$$

These moments are then combined to create an adaptive learning step. The update rule scales the learning rate for each parameter inversely proportional to the square root of its second moment estimate, with an added small constant ϵ to prevent division by zero:

$$w_k^{t+1} = w_k^t - \eta \frac{m_k^t}{\sqrt{v_k^t} + \epsilon}. \tag{2.1.6}$$

This adaptive scaling helps manage varying gradient magnitudes across different model

parameters and training stages, making the optimization more robust to the heterogeneous nature of FL environments.

Similarly, the server-side model aggregation process can be extended with a momentum term as well [76]. Here, we compute the first order momentum over the aggregated model parameters m^t, i.e., we post-process the FedAvg result as

$$m^t = \beta m^{t-1} + (1 - \beta)F(w^t). \tag{2.1.7}$$

The weights for the next training round are then $w^{t+1} = w^t + \eta m^t$. While adaptive optimization can help address data heterogeneity and improve the effectiveness of the FL process, several challenges around computational and communication efficiency remain open.

2.1.2 System Inefficiencies in FL Applications

While FL represents a paradigm shift towards user-centricity by enabling model training on client hardware over wide-area networks, it introduces significant computational and communication challenges. The distributed nature of FL can create substantial bottlenecks on client devices, resulting in excessive consumption of computational resources and heightened processing demands. These fundamental challenges necessitate a careful examination of both computational and communication inefficiencies that arise in federated systems, particularly as they overlap with the practical constraints of edge computing environments.

Computational Inefficiencies

Client resource heterogeneity. Resource heterogeneity among FL clients imposes strict limitations on client operations, including varying maximum batch size $B_{\max}$, model sparsity requirements $|w| \leq s_{\min}$ where the number of model parameters has to be below a minimum sparsity requirement (as measured by the number of parameters), and memory usage bounds. These constraints reflect the practical limitations of edge devices such

as mobile phones and IoT sensors [79, 80]. The variation in computational capabilities spans several orders of magnitude - from high-end mobile devices with dedicated AI accelerators to basic IoT sensors with minimal processing power [81].

Straggler effects. When heterogeneous compute capabilities across clients are involved the overall round time T_{round} is determined by the slowest participating client's computation time T_k^{compute}, a phenomenon that can substantially delay training convergence in real-world deployments [79, 82]. This synchronization bottleneck is particularly pronounced in cross-device FL, where client devices may span multiple hardware generations and operating conditions [80].

Communication Inefficiencies

Efficient communication in distributed learning systems has become increasingly critical as deep learning models grow in size and complexity. Communication overhead, often the primary bottleneck in distributed training, is governed by three key factors.

System heterogeneity. Varying communication times across clients introduce system heterogeneity, a reality in real-world deployments where the total communication time T_{comm} depends on individual client bandwidths Z_k, summed over the set of participating clients S_t. This heterogeneity can lead to stragglers and significantly impact training efficiency.

Bandwidth limitations. Significant bidirectional costs proportional to model size and bit precision originate from bandwidth limitations, with both upload and download costs expressed as $C_{\text{up}} = C_{\text{down}} = |w| \cdot b$.[2] This becomes particularly challenging in FL scenarios where edge devices may have limited capabilities such as limited speed or poor network link reliability.

To address these challenges, researchers have developed various communication compression techniques. These include quantization, which rounds weights to the nearest multiple of a scaling factor, effectively reducing the bits needed per parameter while maintaining

[2]Symmetrical communication costs assume the same energy cost sending and receiving data. Often, sending data from a mobile client incurs more energy than receiving, making the client outbound communication significantly more expensive [83].

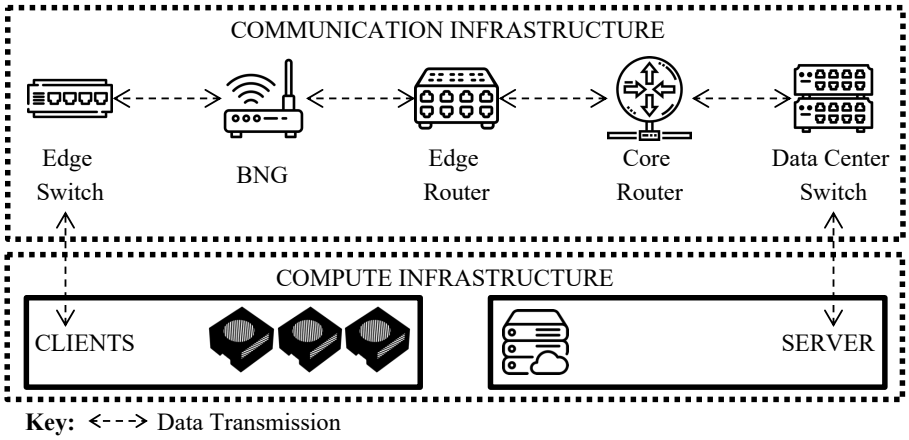


Figure 2.1: Typical architecture of an FL applications implemented over a wide-area network where multiple stakeholders are involved (e.g., end-user, ISP, cloud operator).

statistical properties [64]. Sparsification through top-K selection of weight updates reduces communication volume by transmitting only the most significant updates [84]. Additionally, error feedback mechanisms track the difference between original and compressed updates as $e^{t+1} = w^{t+1} - \hat{w}^{t+1}$ for subsequent compensation, ensuring that no information is permanently lost during compression [85, 86]. These techniques, when properly implemented, can significantly reduce communication overhead while preserving model convergence properties.

2.1.3 Federated Learning on Embedded Hardware

Embedded computing environments present unique challenges for deep learning implementation due to their constrained resources. These systems typically operate with limited memory, processing power, and energy budget [87]. Traditional deep neural networks, which often require gigabytes of memory and substantial computational resources, must be carefully optimized for embedded deployment. This has led to the development of specialized techniques like model quantization, where floating-point weights are converted to lower-precision integers, and network pruning, which removes redundant connections while maintaining model performance [88].

FL in embedded systems. Often, embedded FL applications are implemented over wide-area networks (Figure 2.1). The communication infrastructure typically follows a

hierarchical nature and significantly impacts system design and performance. The path from client to server traverses multiple network layers: client devices initially connect through edge switches, which interface with Broadband Network Gateways (BNG) for access management. The traffic then passes through edge routers that handle boundary routing decisions, followed by core routers managing the backbone traffic distribution. Finally, data center switches facilitate server connectivity. This topology introduces specific technical considerations: each network hop adds latency to the model update transmission, potentially affecting the synchronization of federated rounds. At the data center level, switches must handle burst traffic when scaling the number of clients per training round [83, 89]. Overall, the number of network hops can negatively impact the communication reliability and might lead to network-related client dropouts during training [88, 90].

Resource management and scheduling. Embedded systems running deep learning workloads require sophisticated resource management strategies. Task scheduling must balance multiple competing objectives: meeting real-time processing requirements, minimizing energy consumption, and maintaining thermal constraints [91, 92]. Memory management becomes particularly critical, as embedded systems typically lack the memory capabilities of traditional computing environments [93].

Hardware acceleration and optimization. Modern embedded systems increasingly incorporate specialized hardware accelerators for deep learning operations. These may include neural processing units (NPUs) [94], field-programmable gate arrays (FPGAs) [95], or application-specific integrated circuits (ASICs) [96]. These accelerators are designed to efficiently execute common deep learning operations like convolutions and matrix multiplications while minimizing power consumption. The challenge lies in effectively facilitating hardware/software co-design, maintaining real-time performance requirements [97].

Real-world applications and challenges. Practical applications of embedded deep learning span numerous domains, from autonomous vehicles to smart home devices. These systems must handle varying workloads while maintaining responsive performance. Challenges include handling concept drift, where the statistical properties of the input data change over time, and managing the trade-off between model accuracy and resource

utilization. Real-time requirements add another layer of complexity, as processing must be completed within strict timing constraints [98].

Large Language Models on Embedded Hardware

Given the complexity of implementing FL applications on embedded systems over wide-area networks, scaling to LLM-size models comes with additional challenges mainly rooted in the limited resource availability and the behavioral patterns of embedded clients. The versatility of LLMs offers great potential to use one model for multiple tasks in different FL systems [99].

CPU-only embedded hardware. Recent developments in the embedded computing space such as the Raspberry Pi 5 with up to 16 GB of DDR5 memory have rendered CPU-only embedded devices capale of holding transformer-based models in memory [100]. Together with inference optimization frameworks such as Ollama [101] or vLLM [102] have made it possible to run LLMs on compute limited hardware. Here, techniques like model quantizaton are key to enabling inference on embedded hardware since both memory capacity and memory bandwidth (up to 60 GB/s) are limited. Yet, matrix-multiplication-intensive back-propagation is still infeasible on CPU-only embedded devices.

Inference-only embedded hardware. The intermediate generation of edge AI accelerators introduced specialized neural processing architectures that addressed some of these computational bottlenecks. The Google Coral's Edge TPU, operating at 1 TOPS, demonstrated improved efficiency in processing attention mechanisms through dedicated matrix multiplication units [103]. However, the device's fundamental limitation lies in its on-chip memory capacity of only 8MB, necessitating frequent data transfers between external DRAM and the processing unit. This memory hierarchy poses significant challenges for LLM inference, where transformer layers require continuous access to large parameter matrices. The Intel Neural Compute Stick 2, with its Myriad X VPU, offers 4.1 TOPS of compute performance but faces similar memory constraints, limiting its practical application to heavily pruned and quantized language models [104].

Embedded hardware for training. The NVIDIA Jetson AGX Orin represents a paradigm shift in embedded AI computing capabilities, introducing an architecture including special-

ized Tensor Cores. With up to 64 GB of memory connected at 204.8 GB/s, allows for more efficient handling of transformer-based models. Yet, while the Orin is generally capable of holding large models, the maximum compute capability of 68.5 TOPS (BF16) [45] is rather limited compared to state-of-the-art data center deep learning accelerators [96]. This generally enables training of large models but can require numerous optimizations for efficient operations, such as gradient checkpointing, quantization, and PEFT, if possible.

2.1.4 Emerging AI Regulation

Global AI regulation initiatives are setting data governance, model quality, and resource efficiency requirements. Key inquires of legislators around the globe are enabling the safe and secure deployment of AI applications for a broad audience.

European Union. The European Union has established itself as a global frontrunner in comprehensive AI regulation through the implementation of the EU AI Act, which became effective in February 2025. This legislation introduces a risk-based classification system that categorizes AI technologies into unacceptable, high-risk, limited-risk, and minimal-risk tiers. For systems classified as high-risk, the Act mandates rigorous compliance protocols including comprehensive risk assessments, provisions for human oversight, and extensive transparency requirements [14, 105].

United Kingdom. The United Kingdom has proposed a regulatory strategy in its AI White Paper of March 2023 to complement existing legislation. This approach leverages existing regulatory frameworks and sector-specific guidance rather than implementing new legislation as it has been done in the EU. The UK strategy is anchored in five core principles: safety, transparency, fairness, accountability, and contestability [106].

United States. The United States implements a sector-specific regulatory approach instead of enacting comprehensive AI legislation. While the 2023 Executive Order on Safe, Secure, and Trustworthy AI [107, 108] has been rescinded, the concurrent Executive Order AI as a strategic priority remains in place warranting substantial investments in AI development and infrastructure [109, 110]. Several US states are implementing local laws to install checks and balances for AI applications [58, 59].

China. China has enacted specialized regulatory measures targeting specific AI applications. These include the Measures for the Administration of Internet Information Service Algorithm Recommendation, which governs algorithmic systems, and the Administrative Provisions on Deep Synthesis Internet Information Services, which regulates deepfakes and synthetic media technologies. These regulations prioritize national security considerations, social stability maintenance, and alignment with state-defined objectives [111].

International Frameworks. Multiple international organizations are working to develop common frameworks for AI governance. The OECD AI Principles, which have received endorsement from 42 countries, provide guidelines for trustworthy AI development. The Global Partnership on Artificial Intelligence serves as a platform for international collaboration on AI policy development and responsible implementation practices [66].

Comparative Analysis. These regulatory approaches share common fundamental concerns regarding AI safety, transparency, and accountability, though they differ significantly in implementation methodology and regulatory philosophy. The EU adopts the most prescriptive approach with legally binding requirements and clear categorization of risks, while the UK and US favor more flexible frameworks that leverage existing regulations and industry self-governance. China's approach is distinctive in its emphasis on alignment with state objectives and social stability considerations. The EU and OECD frameworks prioritize human-centric values and rights-based approaches, whereas the US model balances innovation with targeted interventions in high-risk sectors. These differences reflect varying political systems, governance traditions, and strategic priorities regarding technological development and societal protection. Taken together, it will be challenging to design one-solution-fits-all AI applications. Rather, while the core – the AI model – might be the same deployments across regions will face varying safeguard and content censorship requirements.

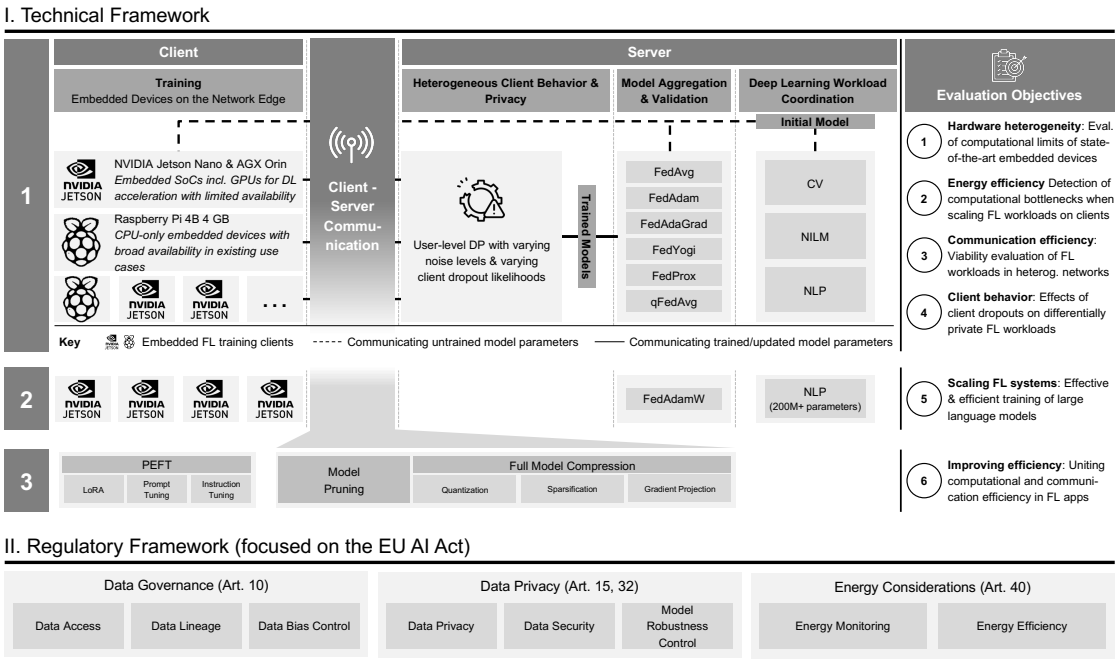


Figure 2.2: Overview of the research methodology applied in this thesis. We first establish a technical framework consisting of a benchmark for FL applications at the network edge, scaling of FL workloads with LLMs, and finally a systematic evaluation of computational and communication efficiency. Second, we design a pathway of using FL to design AI applications compliant with emerging AI regulation.

2.2 Federated Learning in Resource-Constrained Environments

Following our overarching research question, we aim to develop a better understanding of prerequisites for deploying highly capable deep learning models closer to end user devices. The methodology of this thesis follows a dual-framework structure, encompassing technical and regulatory components (Figure 2.2). The technical framework proceeds through three sequential phases: first, a comprehensive benchmark of FL client capabilities to establish baseline performance metrics; second, the scaling of FL workloads to enable the training of deep learning models with more than 200 million parameters; and third, an investigation into the joint optimization of computational and communication efficiency in the context of LLMs. In parallel, the regulatory framework develops a systematic approach for making high-risk AI applications under the EU AI Act regulatory compliant using FL.

Benchmark	Year	Primary Eval. Purpose	ML Domains CV	NLP	NILM	Analysis Dimensions Data Security	Dedicated Edge Deployments	Client Behavior	Client Capabilities
LEAF	2018	Aggregation	✓	✓					
FedML	2020	Aggregation	✓	✓					
Flower	2020	Aggregation	✓	✓		✓			
FederatedScope	2022	Personalization	✓	✓		✓			
FedScale	2022	Scalability	✓	✓		✓			✓
Green FL	2023	Environmental Factors	✓	✓					✓

Table 2.1: Comparison of existing FL benchmark studies with regard to covered domains and their study focus. We see limitations in exploring the characteristics of FL clients on the network edge.

2.2.1 Benchmarking Federated Learning Systems

In our methodology, we first assess the landscape of existing benchmarks for FL applications (Table 2.1). We categorize the benchmarking works into three generations, beginning with LEAF that evaluates the effects of non-IID data on FL algorithms such as FedAvg paired with small-model use cases. LEAF [31] represents the first systematic attempt to provide standardized FL evaluation datasets and metrics. The second generation is characterized by FedML [33], FederatedScope [40], and Flower [60] where scaling workloads has become the priority. During this phase, bigger models and more complex datasets have been used to evaluate the utility of FL algorithms, including adaptive optimization on client and server side. These frameworks address limitations in earlier benchmarks by supporting adaptive optimization, personalization, and more sophisticated communication protocols. The third generation of FL benchmarks, namely FedScale [41] and Green FL [112], is exploring the scalability characteristics of FL applications with models up to 12M parameters and under consideration of environmental aspects. This evolution reflects the growing maturity of FL research, with recent benchmarks continuing to incorporate real-world system constraints and sustainability considerations that were previously overlooked. Through this analysis, we identify a research gap that addresses hardware heterogeneity, energy and communication efficiency, as well as client behavioral patterns.

Hardware Heterogeneity. A hardware-centric evaluation methodology is essential, moving beyond simulations to actual device testing. This requires a diverse testbed with multiple generations of edge devices (both legacy and state-of-the-art), ranging from resource-constrained platforms to more capable edge AI accelerators. The main benefit originates from the immediate access to real performance characteristics rather than relying on theoretical models or simulated environments.

Client Behavior Analysis. Realistic client behavior modeling is necessary to understand how client unreliability impacts FL workloads. This requires implementing mechanisms to emulate varying dropout patterns and participation rates that mirror real-world deployment scenarios. The system should support configurable client reliability profiles based on target environments (industrial, mobile, remote sensing, etc.). Comprehensive privacy evaluation framework that adapts to client participation patterns is critical. The system must integrate user-level differential privacy that can dynamically adjust noise levels based on the number of actual client updates received. This requires measuring the trade-offs between privacy guarantees and model quality under varying client reliability conditions.

Computational & Communication Analysis. Network condition emulation capabilities are needed to evaluate communication efficiency across different connectivity profiles. The framework should support configurable network characteristics (bandwidth, latency, reliability) to test FL workloads under factory-grade connections, wireless networks, and limited connectivity scenarios typical of edge deployments. Energy measurement instrumentation for accurate power profiling across heterogeneous devices is vital. This requires hardware-level power monitoring for embedded devices and appropriate estimation techniques for virtualized environments, allowing energy efficiency to be quantified as throughput per watt across different model architectures and sizes.

To truly close the research gap, these requirements must be integrated into a cohesive benchmarking framework that enables systematic evaluation of FL workloads across the full spectrum of edge computing environments, from industrial settings with reliable connectivity to challenging remote deployments with intermittent participation and limited resources.

2.2.2 Scaling FL Workloads

To effectively scale FL workloads across heterogeneous edge environments, a comprehensive evaluation methodology is required that addresses the fundamental challenges of computational resources, energy constraints, and communication overhead. This methodology systematically evaluates FL scalability across model sizes, hardware platforms, and network configurations.

The scaling assessment begins with model complexity evaluation, utilizing the encoder-decoder FLAN-T5 transformer family (ranging from 80M to 3B parameters) to represent varying computational demands. This approach enables identification of precise scaling thresholds for edge deployment across lightweight to heavyweight model architectures. Computational scalability is measured through Model FLOP Utilization (MFU) analysis paired with fine-grained training step decomposition (forward pass, loss calculation, optimizer step, backward pass) to pinpoint scaling bottlenecks. Mini-batch scaling experiments systematically determine at what point resource-constrained devices reach computational saturation, revealing that embedded hardware reaches practical computational limits significantly earlier than theoretical analysis suggests.

To address energy scaling challenges, we require an in-depth understanding of energy efficiency, providing a hardware-agnostic metric suitable for heterogeneous FL deployments. This enables real-time scalability assessment without requiring detailed client hardware specifications, which is particularly valuable when scaling to numerous diverse edge devices. Communication scalability, often the primary limiting factor in large-scale FL deployments, is evaluated through Granularity analysis, measuring the computation-to-communication time ratio to determine if adding more clients would improve system throughput and accelerate training. This is complemented by a per-bit communication cost model that quantifies energy consumption across different network configurations, revealing scaling implications for both bandwidth-limited wireless and higher-capacity wired connections.

We evaluate the scaling limitation by incorporating state-of-the-art techniques for parameter-efficient training (e.g., Low-Rank Adaptation (LoRA)) that can reduce the number of trainable parameters to $\leq 1\%$ of total model size. Four federated optimization techniques (FedAvg, FedAvgM, FedAdam, and FedAdamW) are benchmarked to identify which approach delivers the most favorable convergence speed and the highest final model quality at scale.

All experiments are conducted in a physical testbed with holistic measurements of client, server, and communication metrics across modern data center hardware (e.g., NVIDIA A100) and state-of-the-art embedded devices (e.g., NVIDIA Jetson AGX Orin). This controlled experimental approach enables quantitative assessment of how each factor –

model size, hardware capabilities, optimization technique, and communication pattern – impacts FL scalability on the network edge.

2.2.3 Computational and Communication Efficiency in Embedded FL Systems

When scaling FL workloads on the network edge, the limited memory bandwidth and orders of magnitude more expensive communication costs become key limiting factors. To develop an understand of how computational and communication efficiency can become a joint optimization target, we conducted a comprehensive analysis of computational and communication approaches for foundation model deployment in FL applications through a structured methodological framework. Our investigation involved: (1) systematic literature analysis from premier machine learning, distributed systems, and security venues; (2) comparative analysis using standardized metrics (e.g., model parameter count, trainable parameter percentage, communication overhead reduction, and cross-domain applicability); (3) capabilities assessment of FL frameworks (e.g., security aggregation protocols, edge device compatibility); and (4) methodical gap analysis employing differential mapping between computational efficiency methods and communication optimization techniques. With our multi-dimensional approach, we outline future research to unite computational and communication efficiency in a joint optimization goal for training LLMs with FL.

2.3 Regulatory Considerations

The EU AI Act introduces comprehensive regulatory requirements for machine learning systems, particularly those deployed in high-risk applications. This paper examines how FL can serve as a technical framework to achieve regulatory compliance while preserving privacy and system performance. Through an interdisciplinary approach combining legal analysis and technical assessment, we evaluate the capacity of FL to address core regulatory challenges in data governance, privacy preservation, and model quality control, while also examining the distribution of responsibilities among stakeholders under this

emerging regulatory landscape.

2.3.1 Achieving Regulatory Compliance Under the EU AI Act Using FL

We employ a three-pronged methodological approach to evaluate the implications of the EU AI Act and how FL can help achieve full regulatory compliance. Our methodology combines legal analysis, quantitative experimentation, and qualitative assessment to provide a comprehensive understanding of the regulatory challenges and opportunities, especially for high-risk applications such as AI-enabled job application screening tools or financial analysis tools.

To systematically analyze FL under the AI Act, we establish evaluation criteria aligned with key regulatory requirements. For data governance, our analysis focuses on data bias reduction and enforcement of regulatory privacy. The legislation requires high-risk applications to implement robust data governance practices, including bias detection, documentation of training data, and adherence to GDPR principles. We examine the potential of FL to mitigate data bias by improving data availability and creating broader training datasets while maintaining privacy guarantees.

For privacy evaluation, we investigate technical capabilities of private and secure computation methods in FL applications, including Secure Multi-Party Computation, Homomorphic Encryption, and Differential Privacy. We identify potential gaps between these state-of-the-art privacy techniques and the regulatory requirements that necessitate examination of training data for biases while maintaining GDPR compliance.

Regarding energy efficiency, we develop a holistic methodology accounting for both computational and communication energy costs, addressing forthcoming energy efficiency requirements. The total energy consumption is modeled as the sum of computational energy and communication energy. Beyond energy considerations, we evaluate model quality control and robustness mechanisms that operate without direct data access. The AI Act mandates appropriate levels of accuracy, robustness, and consistent performance through quality management systems.

For empirical evaluation, we design experiments to quantify measurable impacts of implementing AI Act requirements in FL systems. We implement an FL system to fine-tune a 110M parameter BERT transformer model on the 20 News Group dataset, simulating a high-risk application (job application screening). The experimental setup comprises 100 non-IID client subsets created via Latent Dirichlet Allocation, reflecting realistic data heterogeneity scenarios in federated environments.

We evaluate different privacy-preserving techniques across various parameters to understand the trade-off between system scalability, model performance, and privacy protection. We quantify the costs of validation and model monitoring to ensure robustness and quality management, measuring idle time energy consumption during validation phases. The experiments use Federated Averaging (FedAvg) with 2,000 aggregation rounds and 10% client participation per round.

To complement the quantitative measurements, we conduct a qualitative analysis of FL characteristics that cannot be empirically quantified. We evaluate the potential of FL to access siloed data and generate more representative models compared to centralized approaches, addressing the EU AI Act requirements on representative, high-quality training data. We assess the inherent advantages of FL in data lineage tracking and privacy protection, which align with the "data protection by design" principles.

We examine the potential of FL potential to facilitate compliance with GDPR requirements within the EU AI Act framework, particularly regarding data subject rights and data security. Additionally, we identify the potential for FL to become the preferred privacy-preserving ML technique for high-risk applications under the EU AI Act, considering its alignment with fundamental EU values including privacy, data protection, and non-discrimination. This comprehensive methodology enables us to identify research priorities that would facilitate FL adoption in the emerging regulatory landscape, combining technical benchmarking with forward-looking policy analysis.

2.3.2 Defining Responsibility in FL Systems Under the EU AI Act

We employed an interdisciplinary methodology to examine responsibility allocation in FL systems under EU AI regulation. We conducted normative legal analysis, interpreting key provisions of the EU AI Act as it applies to FL. This provided the regulatory foundation for our technical investigation.

To understand the technical dimensions, we methodically deconstructed the FL pipeline into its constituent components: data acquisition, storage, preprocessing, and model aggregation. This systems decomposition approach enabled us to identify specific responsibility touchpoints throughout the FL lifecycle, highlighting how the server-client architecture inherently distributes control.

Our methodology incorporated a comparative analysis of cross-silo and cross-device FL architectures to demonstrate how architectural choices influence the practical distribution of legal responsibilities. This comparison revealed distinct patterns of responsibility allocation depending on implementation choices.

We utilized gap analysis to identify inconsistencies between current regulatory frameworks and FL implementations, pinpointing areas requiring further technical development and regulatory clarification. This analysis highlighted needs for improved auditability, verifiability, integrity, and privacy mechanisms.

The core methodological contribution of our work lies in its interdisciplinary synthesis. By integrating computer science and legal perspectives, we developed insights into both technical solutions and regulatory implementations that could clarify stakeholder roles in FL environments. This interdisciplinary approach was essential given the socio-technical nature of responsibility allocation in cooperative machine learning systems. Our methodology purposefully focuses on conceptual analysis rather than empirical validation, establishing a foundational framework for understanding responsibility allocation in FL under emerging AI regulation.

Chapter 3

Publication Summary

In this chapter, we summarize the three core contributions to this publication-based thesis. We also discuss two non-core publications that frame the regulatory context for deployment of FL applications within the EU.

3.1 FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems

REFERENCE Herbert Woisetschläger, Alexander Erben, Ruben Mayer, Shiqiang Wang, and Hans-Arno Jacobsen. "FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems." In: *Proceedings of the 25th International Middleware Conference*. Middleware '24. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 88–102. ISBN: 9798400706233. DOI: 10.1145/3652892.3700751. URL: https://doi.org/10.1145/3652892.3700751 (Core publication #1)

FULL-TEXT VERSION ENCLOSED Appendix A

SUMMARY We introduce FLEdge, a novel benchmarking framework designed to evaluate Federated Learning (FL) applications in edge computing environments. Our research addresses a significant gap in existing FL benchmarks by focusing specifically on client capabilities in heterogeneous hardware settings, with particular attention to computational bottlenecks, client behavior patterns, and data security implications. Through extensive experimentation using models ranging from 14K to 80M trainable parameters, we conducted tests on dedicated hardware with emulated network characteristics and client behavior. Our findings reveal that current state-of-the-art embedded hardware faces substantial memory limitations, leading to processing times up to 4 times longer than modern data center GPUs. That said, common performance optimization patters known to work well in data center environments like minibatch size scaling can lead to less performance in an FL system. Instead, it is better to scale the number of clients per training round in the context of required throughput and communication cost. Our study also finds that FL aggregation techniques can cope with moderate client unreliability ($\leq$ 20%) but suffer from notable performance loss in environments where clients or communication drop out frequently. This research advances the understanding of real-world implementation challenges in FL systems while providing a practical benchmark suite easily extensible with additional evaluation modules.

AUTHOR CONTRIBUTIONS Conceived and implemented the approach. Designed the experiments and conducted the analysis. Wrote the paper. **Sole first authorship** ($>$ 50% contribution).

3.2 Federated Fine-tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly

REFERENCE Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly." In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. Santiago, Chile: Association for Computing Machinery, 2024, pp. 39–50. ISBN: 9798400706110. DOI: 10.1145/3650203.3663331. URL: https://doi.org/10.1145/3650203.3663331 (Core publication #2)

FULL-TEXT VERSION ENCLOSED Appendix B

SUMMARY In our paper, we conduct an end-to-end evaluation of federated fine-tuning large language models (LLMs) on edge devices. We focus our analysis on fine-tuning the FLAN-T5 model family on edge networks, systematically studying energy efficiency across clients, communication, and servers. We demonstrate that edge devices like the NVIDIA Jetson AGX Orin face substantial memory bandwidth limitations compared to data center GPUs, which materially impact computational efficiency. We introduce energy efficiency as a novel real-time metric for assessing computational efficiency in federated learning systems, showing that it correlates strongly with traditional Model-FLOP Utilization metrics while being more practical to measure in real-world deployments. Through extensive experimentation, we establish that adaptive federated optimization techniques, particularly our proposed FedAdamW optimizer, can achieve up to 8x faster convergence compared to standard FedAvg with momentum. Our analysis reveals critical challenges in communication efficiency when deploying foundation models in federated settings, with communication costs exceeding computational costs by up to four orders of magnitude. These findings underscore the urgent need for more efficient communication protocols and optimization strategies in edge-based federated learning systems to make federated learning of foundation models practical at scale.

AUTHOR CONTRIBUTIONS Conceived and implemented the approach. Designed the experiments and conducted the analysis. Wrote the paper. **Sole first authorship** ($>$ 50% contribution).

3.3 A Survey on Efficient Federated Learning Methods for Foundation Model Training

REFERENCE Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "A survey on efficient federated learning methods for foundation model training." In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. IJCAI '24. Jeju, Korea, 2024, pp. 8317–8325. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/919. URL: https://doi.org/10.24963/ijcai.2024/919 (Core publication #3)

FULL-TEXT VERSION ENCLOSED Appendix C

SUMMARY In this survey, we introduce a novel taxonomy focused on computational and communication efficiency for training foundation models (FMs) using federated learning (FL). While FL has become established for privacy-preserving collaborative training, existing approaches often discuss small deep-learning models and full-model training. The reality for many applications is different - FMs are typically pre-trained across diverse tasks and can be fine-tuned for specific downstream tasks using significantly smaller datasets. We identify a significant gap between FL methods for computational and communication efficiency. While research exists on computational efficiency for FM training and fine-tuning in FL applications, communication efficiency methods predominantly target full-model training. Our taxonomy aims to identify synergies between these approaches. We provide a holistic evaluation of existing computational efficiency methods for FMs and communication efficiency techniques in FL settings. We examine how current techniques can drive FM adoption and assess the readiness of FL frameworks for large models. Additionally, we discuss important future research directions, highlighting areas where computational and communication efficiency domains are converging. We outline the technical challenges hindering FM operationalization in federated applications, particularly regarding generative tasks and privacy considerations.

AUTHOR CONTRIBUTIONS Conceived and implemented the approach. Designed the experiments and conducted the analysis. Wrote the paper. **Sole first authorship** ($>$ 50% contribution).

3.4 Federated Learning Priorities Under the European Union Artificial Intelligence Act

REFERENCE Herbert Woisetschläger, Alexander Erben, Bill Marino, Shiqiang Wang, Nicholas D. Lane, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning Priorities Under the European Union Artificial Intelligence Act." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2402.05968. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/48.pdf (Non-core publication #1)

FULL-TEXT VERSION ENCLOSED Appendix D

SUMMARY We present an analysis demonstrating that Federated Learning (FL) is key to achieving compliance with the European Union's AI Act, particularly for high-risk AI applications. Through careful examination of the FL architecture, we show how it inherently addresses critical regulatory requirements around data governance, privacy, and resource allocation. Our analysis focuses on three fundamental areas: data governance, where FL enables improved access to domain data while maintaining local control; privacy and security, where FL supports lawful data processing through local training; and energy efficiency, where FL allows for distributed resource utilization.

While establishing the strong alignment of FL with regulatory requirements, we also critically assess its current limitations. We identify significant challenges, including difficulties in implementing "right to be forgotten" requirements, managing energy-privacy trade-offs, and handling the computational overhead of privacy-preserving techniques. Our experimental analysis shows that adding private computing techniques to an FL application can render the traning process practically infeasible compared to centralized learning. We provide a comprehensive research agenda addressing these gaps and outline priorities for enhancing the regulatory compliance of FL.

AUTHOR CONTRIBUTIONS Co-developed and implemented the approach. Designed the experiments and conducted the analysis. Wrote the paper. **Sole first authorship** (> 50% contribution).

3.5 Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis

REFERENCE Herbert Woisetschläger*, Simon Mertel*, Christoph Krönke, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2407.08105. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/16.pdf (Non-core publication #2, * indicates shared first authorship)

FULL-TEXT VERSION ENCLOSED Appendix E

SUMMARY We present an interdisciplinary analysis of responsibility allocation in FL systems under the EU AI Act. Our research examines how FL, which enables distributed training of AI models while preserving data privacy through parameter sharing, intersects with new regulatory requirements for AI system providers. Our investigation reveals that while FL naturally distributes responsibility between clients and server operators, there are compelling technical and legal reasons to consolidate primary responsibility with the server operator as the designated "service provider" under the EU AI Act. We identify critical technical capabilities needed to enable this responsibility transition, including enhanced auditability, verifiability, data integrity verification, and privacy preservation mechanisms. Our analysis distinguishes between cross-device FL, involving numerous small-scale clients, and cross-silo FL, with fewer institutional participants, noting that the latter architecture provides greater flexibility in responsibility distribution. We propose specific technical and regulatory implementations to help server operators assume comprehensive service provider responsibilities while maintaining client compliance. This work contributes to the ongoing dialogue with the EU AI Office regarding EU AI Act implementation.

AUTHOR CONTRIBUTIONS Co-developed and implemented the approach. Conducted the analysis. Wrote half of the paper. **Shared first authorship** (50% contribution).

Chapter 4

Discussion

This thesis provides an interdisciplinary perspective on energy efficient FL systems for both small-scale and LLM-scale model training, especially on the network edge and under consideration of emerging AI regulation. In this chapter, we broadly discuss the main finding of our work in a broader context and highlight future research opportunities that can help make FL more suitable for edge computing use cases while improving regulatory compliance with globally emerging AI regulation.

Benchmarking FL at the network edge

Through the development of FLEdge, a novel benchmarking framework, the research provides valuable insights into the performance of FL clients on edge devices, revealing substantial memory limitations and processing constraints in current state-of-the-art embedded hardware. Our comprehensive benchmarking approach demonstrates that even the most advanced embedded platforms struggle with the computational demands of modern FL workloads. We identified significant memory bottlenecks that result in processing times up to 4× longer on edge devices compared to modern data center GPUs. This performance gap proved especially pronounced during backpropagation, suggesting that current embedded AI accelerators require alternative architectural solutions or algorithms that reduce the memory pressure during training, or both. Contrary to established practices in high-performance deep learning, we find that scaling batch sizes on embedded devices does not lead to greater computational efficiencies.

In fact, our micro-benchmarks revealed that doubling the minibatch size typically doubles runtime rather than providing the performance improvement observed in data center environments. This challenges conventional optimization approaches and necessitates edge-specific strategies. Our detailed profiling experiments with the FLAN-T5 Small model uncovered specific hardware limitations, including inefficient context switching between CPU and GPU on devices with unified memory. Despite theoretical advantages of shared memory architecture, these context switches consumed more processing time than direct memory transfers on discrete GPU systems.

By providing this hardware-centric perspective on FL, FLEdge closes the research gap that existed in understanding deployment challenges and develop more efficient solutions for edge computing environments where energy constraints, client reliability, and hardware diversity are critical considerations.

Training LLMs in FL systems

We further demonstrate that scaling FL systems to accommodate LLMs presents unique challenges, particularly regarding computational efficiency, energy consumption, and communication costs. Through extensive experimentation with the FLAN-T5 model family on NVIDIA Jetson AGX Orin devices, we establish that memory bandwidth limitations significantly impact computational efficiency even more than what we had discovered when initially using FLEdge.

We observe that while these edge devices possess significantly improved computational capabilities compared to previous generations, they still face severe memory bandwidth bottlenecks when handling foundation models like FLAN-T5 Large or even XL. This limitation manifests as linearly growing optimization step times on edge devices, compared to exponential efficiency gains on data center GPUs like the NVIDIA A100. We introduce an energy efficiency metric (η_e) defined as tokens per second throughput over average power draw, which proves valuable for real-time monitoring without requiring detailed hardware specifications. Our analysis reveals a strong correlation between η_e and MFU across the tested models, making it an excellent practical alternative in FL settings where client hardware details may be unknown or heterogeneous.

By quantifying the cost of LLM training in FL systems, we provide valuable insights into hardware, software, and communication bottlenecks. The proposed strategies for assessing client capabilities in an online fashion and holistically capturing the communication cost can be used a practical tools to improve the overall energy efficiency of FL systems for increased regulatory compliance.

Computational and communication efficiency as a joint optimization objective

Our systematic analysis of computational and communication efficiency in FL applications led to the development of a novel taxonomy that identifies critical gaps between these domains. Our work highlights that while computational efficiency methods for foundation models exist, communication efficiency techniques predominantly target full-model training, creating a significant disconnect that hampers practical deployment of foundation models in federated settings. PEFT has proven to be effective for reducing both the computational and communication intensity in FL applications but exhibits increased sensitivity toward data heterogeneity. Further, the compatibility with private computing techniques such as (ϵ, δ)-DP is unclear as the PEFT responds stronger to the training data than full model training [113].

The methodological misalignment between computational and communication efficiency is increasingly challenging as foundation models scale exponentially to billions of parameters. We identify this gap as a critical research opportunity for developing integrated methodologies that combine the computational advantages of parameter-efficient fine-tuning with communication-efficient protocols specifically optimized for distributed foundation model deployment.

FL under emerging AI regulation

While FL inherently addresses key regulatory requirements around data governance and privacy, our interdisciplinary analysis reveals significant research gaps that must be addressed for FL to become the standard for regulatory-compliant AI systems.

The EU AI Act establishes comprehensive requirements for high-risk AI applications. Our analysis positions FL as uniquely advantageous in this new regulatory landscape

due to its fundamental design principles. By processing data at its source rather than centralizing it, FL naturally aligns with the Act's emphasis on data protection by design and default. This architecture provides natural data lineage capabilities, simplifies GDPR compliance, and enables access to valuable siloed data that might otherwise remain inaccessible due to privacy concerns.

However, our research identifies several critical gaps that the FL community must address. The current performance trade-offs highlighted in the AI Act present challenges for FL systems, particularly regarding energy efficiency, computational costs of privacy, and lifecycle monitoring under privacy-preserving operations. The strict data governance requirements of the Act demand new approaches to data quality management and bias detection that can function without direct data access. Work towards ethics in FL systems provides a promising path to facilitating data governance in a comprehensive and practical way [114].

We advocate for a significant redirection of research priorities in the FL community. First, data quality requirements must be made amenable to FL through indirect quality assessment techniques that do not compromise privacy. Second, energy efficiency must become a primary concern, with CO_2-based optimization approaches developed to compete with centralized training. Third, we need clearer expressions of privacy within the regulatory context of the EU AI Act, including better alignment of technical privacy-preserving mechanisms with legal requirements.

Further, the allocation of responsibilities to model provider, FL server operator, and clients has been an open question. Our analysis demonstrates that the data locality of FL resolves challenges in monitoring data lineage and simplifies consent management, a key inquiry of the EU AI Act. However, the cooperative nature of FL training introduces complexity in determining service provider responsibilities. We argue that technical solutions focusing on auditability, verifiability, integrity, and privacy must be developed to enable server operators to assume full responsibility, particularly in cross-device settings. For cross-silo architectures, we propose balanced responsibility distribution through carefully structured terms of service agreements. Resolving these responsibility allocations is critical for unlocking the potential of FL to enhance data access while maintaining regulatory compliance under the AI Act.

Chapter 5

Conclusions

This thesis has examined FL systems on *embedded hardware*, focusing on *scalability*, *efficiency*, and *regulatory compliance*. Our research addresses fundamental challenges in bringing modern FL workloads to the network edge, including practical deployment considerations regarding computational and communication efficiency. We also examine the path towards regulatory compliant deep learning systems under the EU AI Act. With our systematic analysis we show not only that energy efficiency is a technical priority but has become a defacto requirement under emerging AI regulation. This opens a clear path for future research addressing legal priorities by means of improving the training efficiency on FL clients as well as the communication when large deep learning models are involved.

Our benchmarking framework, FLEdge, reveals key performance limitations when deploying FL applications on edge devices. Memory bottlenecks and inefficient hardware utilization result in substantially longer training times compared to data center environments. Contrary to established practices, batch size scaling on embedded hardware does not benefit the overall training efficiency. These findings demonstrate that successful edge deployment requires fundamentally different optimization approaches, with energy consumption and memory bandwidth as the primary constraints in mind.

The incorporation of LLMs in FL introduces significant challenges at the network edge.

Our systematic analysis of LLMs demonstrates that while computationally feasible on modern embedded hardware, training efficiency decreases notably with model scale. Memory bandwidth limitations create significant bottlenecks during backpropagation. This renders PEFT as a principally well-suited approach to tailoring pre-trained LLMs toward specific use cases. When full model training is required, communication costs become particularly challenging, exceeding computational costs by several orders of magnitude. Thus, the need for joint optimization of computational and communication cost grows with the number of model parameters.

The distributed architecture of FL positions it advantageously within emerging regulatory frameworks, particularly the EU AI Act. By maintaining data locality and minimizing raw data transfers, FL inherently satisfies requirements for data protection by design. However, our interdisciplinary analysis identifies significant gaps between current technical capabilities and regulatory expectations regarding energy efficiency, data quality assurance, and responsibility allocation. These findings suggest that future FL research must prioritize verifiable data governance without centralization, energy-aware optimization, and technical mechanisms that enable appropriate responsibility distribution among system participants to achieve full regulatory compliance.

Overall, we hope that this work spurs research towards more energy efficient and even more user-centric FL techniques that ultimately help build stronger and more helpful models for everyone.

Bibliography

[1] Herbert Woisetschläger, Alexander Erben, Ruben Mayer, Shiqiang Wang, and Hans-Arno Jacobsen. "FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems." In: *Proceedings of the 25th International Middleware Conference*. Middleware '24. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 88–102. ISBN: 9798400706233. DOI: 10.1145/3652892.3700751. URL: https://doi.org/10.1145/3652892.3700751.

[2] Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly." In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. Santiago, Chile: Association for Computing Machinery, 2024, pp. 39–50. ISBN: 9798400706110. DOI: 10.1145/3650203.3663331. URL: https://doi.org/10.1145/3650203.3663331.

[3] Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "A survey on efficient federated learning methods for foundation model training." In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. IJCAI '24. Jeju, Korea, 2024, pp. 8317–8325. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/919. URL: https://doi.org/10.24963/ijcai.2024/919.

[4] Herbert Woisetschläger, Alexander Erben, Bill Marino, Shiqiang Wang, Nicholas D. Lane, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning Priorities Under the European Union Artificial Intelligence Act." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2402.05968. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/48.pdf.

[5] Herbert Woisetschläger*, Simon Mertel*, Christoph Krönke, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2407.08105. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/16.pdf.

[6] Daouda Sow, Herbert Woisetschläger, Saikiran Bulusu, Shiqiang Wang, Hans-Arno Jacobsen, and Yingbin Liang. "Dynamic Loss-Based Sample Reweighting for Improved Large Language Model

Pretraining." In: *The Thirteenth International Conference on Learning Representations*. ICLR'25. 2025. URL: https://openreview.net/forum?id=gU4ZgQNsOC.

[7] Hajar Emami Gohari, Swanand Ravindra Kadhe, Syed Yousaf Shah, Constantin Adam, Abdulhamid Adebayo, Praneet Adusumilli, Farhan Ahmed, Nathalie Baracaldo Angel, Santosh Borse, Yuan-Chi Chang, Xuan-Hong Dang, Nirmit Desai, Ravital Eres, Ran Iwamoto, Alexei Karve, Yan Koyfman, Wei-Han Lee, Changchang Liu, Boris Lublinsky, Takuyo Ohko, Pablo Pesce, Maroun Touma, Shiqiang Wang, Shalisha Witherspoon, Herbert Woisetschläger, David Wood, Kun-Lung Wu, Issei Yoshida, Syed Zawad, Petros Zerfos, Yi Zhou, and Bishwaranjan Bhattacharjee. "GneissWeb: Preparing High Quality Data for LLMs at Scale." Feb. 2025. URL: https://huggingface.co/datasets/ibm-granite/GneissWeb.

[8] Ryan Zhang, Herbert Woisetschläger, Shiqiang Wang, and Hans Arno Jacobsen. "MESS+: Energy-Optimal Inferencing in Language Model Zoos with Service Level Guarantees." In: *Workshop on Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning in Conjunction with NeurIPS'24*. 2024. URL: https://openreview.net/forum?id=OoReeQpwmW.

[9] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschläger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. "A survey on dataset distillation: approaches, applications and future directions." In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI '23. Macao, P.R.China, 2023. ISBN: 978-1-956792-03-4. DOI: 10.24963/ijcai.2023/741. URL: https://doi.org/10.24963/ijcai.2023/741.

[10] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data." In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 20–22 Apr 2017, pp. 1273–1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[11] Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. "Federated learning review: Fundamentals, enabling technologies, and future applications." In: *Information Processing & Management* 59.6 (Nov. 2022), p. 103061. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2022.103061. URL: http://dx.doi.org/10.1016/j.ipm.2022.103061.

[12] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. "The future of digital health with federated learning." In: *npj Digital Medicine* 3.1 (Sept. 2020). ISSN: 2398-6352. DOI: 10.1038/s41746-020-00323-1. URL: http://dx.doi.org/10.1038/s41746-020-00323-1.

[13] Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. *Federated Learning for Mobile Keyboard Prediction*. 2018. URL: https://arxiv.org/abs/1811.03604.

[14] Council of the European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)Text with EEA relevance.* Document 32024R1689. July 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[15] Martin Kretschmer, Thomas Margoni, and Pinar Oruç. "Copyright Law and the Lifecycle of Machine Learning Models." In: *IIC - International Review of Intellectual Property and Competition Law* 55.1 (Jan. 2024), pp. 110–138. ISSN: 2195-0237. DOI: 10.1007/s40319-023-01419-3. URL: http://dx.doi.org/10.1007/s40319-023-01419-3.

[16] Giorgio Franceschelli and Mirco Musolesi. "Copyright in generative deep learning." In: *Data & Policy* 4 (2022). ISSN: 2632-3249. DOI: 10.1017/dap.2022.10. URL: http://dx.doi.org/10.1017/dap.2022.10.

[17] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. "A survey on federated learning." In: *Knowledge-Based Systems* 216 (Mar. 2021), p. 106775. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2021.106775. URL: http://dx.doi.org/10.1016/j.knosys.2021.106775.

[18] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. "A Principled Approach to Data Valuation for Federated Learning." In: *Federated Learning*. Springer International Publishing, 2020, pp. 153–167. ISBN: 9783030630768. DOI: 10.1007/978-3-030-63076-8_11. URL: http://dx.doi.org/10.1007/978-3-030-63076-8_11.

[19] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. "A Fairness-aware Incentive Scheme for Federated Learning." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. AIES '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 393–399. ISBN: 9781450371100. DOI: 10.1145/3375627.3375840. URL: https://doi.org/10.1145/3375627.3375840.

[20] Ollie Liu, Sami Jaghouar, Johannes Hagemann, Shangshang Wang, Jason Wiemels, Jeff Kaufman, and Willie Neiswanger. *METAGENE-1: Metagenomic Foundation Model for Pandemic Monitoring*. 2025. DOI: 10.48550/ARXIV.2501.02045. URL: https://arxiv.org/abs/2501.02045.

[21] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. "Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models." In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=Tlsdsb6l9n.

[22] Michael D. Ward, Maxwell I. Zimmerman, Artur Meller, Moses Chung, S. J. Swamidass, and Gregory R. Bowman. "Deep learning the structural determinants of protein biochemical properties by comparing structural ensembles with DiffNets." In: *Nature Communications* 12.1 (May 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-021-23246-1. URL: http://dx.doi.org/10.1038/s41467-021-23246-1.

[23] Michael Shirts and Vijay S. Pande. "Screen Savers of the World Unite!" In: *Science* 290.5498 (Dec. 2000), pp. 1903–1904. ISSN: 1095-9203. DOI: 10.1126/science.290.5498.1903. URL: http://dx.doi.org/10.1126/science.290.5498.1903.

[24] Council of the European Union. *European Health Data Space: Council adopts new regulation improving cross-border access to EU health data*. 2025. URL: https://www.consilium.europa.eu/en/press/press-releases/2025/01/21/european-health-data-space-council-adopts-new-regulation-improving-cross-border-access-to-eu-health-data/.

[25] Council of the European Union. *General data protection regulation (GDPR)*. Document 32016R0679. Apr. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679.

[26] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. "PAPAYA: Practical, Private, and Scalable Federated Learning." In: *Proceedings of Machine Learning and Systems*. Ed. by D. Marculescu, Y. Chi, and C. Wu. Vol. 4. 2022, pp. 814–832. URL: https://proceedings.mlsys.org/paper_files/paper/2022/file/a8bc4cb14a20f20d1f96188bd61eec87-Paper.pdf.

[27] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. "Federated Learning with Buffered Asynchronous Aggregation." In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 28–30 Mar 2022, pp. 3581–3607. URL: https://proceedings.mlr.press/v151/nguyen22b.html.

[28] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. "A Survey on Federated Learning for Resource-Constrained IoT Devices." In: *IEEE Internet of Things Journal* 9.1 (2022), pp. 1–24. DOI: 10.1109/JIOT.2021.3095077.

[29] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. "Adaptive Federated Learning in Resource Constrained Edge Computing Systems." In: *IEEE Journal on Selected Areas in Communications* 37.6 (2019), pp. 1205–1221. DOI: 10.1109/JSAC.2019.2904348.

[30] NVIDIA. *NVIDIA DGX Spark - A Grace Blackwell AI Supercomputer on your desk*. https://www.nvidia.com/en-us/products/workstations/dgx-spark/. [Accessed 07-04-2025]. 2025.

[31] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. *LEAF: A Benchmark for Federated Settings*. 2018. DOI: 10.48550/ARXIV.1812.01097. URL: https://arxiv.org/abs/1812.01097.

[32] Fan Lai, Yinwei Dai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. "FedScale: Benchmarking Model and System Performance of Federated Learning." In: *Proceedings of the First Workshop on Systems Challenges in Reliable and Secure Federated Learning*. ResilientFL '21. Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 1–3. ISBN: 9781450387088. DOI: 10.1145/3477114.3488760. URL: https://doi.org/10.1145/3477114.3488760.

[33] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. *FedML: A Research Library and Benchmark for Federated Machine Learning*. 2020. DOI: 10.48550/ARXIV.2007.13518. URL: https://arxiv.org/abs/2007.13518.

[34] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners." In: (2019).

[35] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. *Language Models are Few-Shot Learners*. 2020. DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165.

[36] Lorenzo Sani, Alex Iacob, Zeyu Cao, Royson Lee, Bill Marino, Yan Gao, Dongqi Cai, Zexi Li, Wanru Zhao, Xinchi Qiu, and Nicholas D. Lane. *Photon: Federated LLM Pre-Training*. 2024. DOI: 10.48550/ARXIV.2411.02908. URL: https://arxiv.org/abs/2411.02908.

[37] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Wang, and Yiran Chen. *Towards Building the Federated GPT: Federated Instruction Tuning*. 2023. arXiv: 2305.05644 [cs.CL].

[38] Yuanyishu Tian, Yao Wan, Lingjuan Lyu, Dezhong Yao, Hai Jin, and Lichao Sun. "FedBERT: When Federated Learning Meets Pre-training." In: *ACM Trans. Intell. Syst. Technol.* 13.4 (Aug. 2022). ISSN: 2157-6904. DOI: 10.1145/3510033. URL: https://doi.org/10.1145/3510033.

[39] Tobias Müller, Milena Zahn, and Florian Matthes. "Unlocking the Potential of Collaborative AI - On the Socio-Technical Challenges of Federated Machine Learning." In: *31st European Conference on Information Systems - Co-creating Sustainable Digital Futures, ECIS 2023, Kristiansan, Norway, June 11-16, 2023*. Ed. by Margunn Aanestad, Stefan Klein, Monideepa Tarafdar, Shengnan Han, Sven Laumer, and Isabel Ramos. 2023. URL: https://aisel.aisnet.org/ecis2023%5C_rp/245.

[40] Yuexiang Xie, Zhen Wang, Dawei Gao, Daoyuan Chen, Liuyi Yao, Weirui Kuang, Yaliang Li, Bolin Ding, and Jingren Zhou. "FederatedScope: A Flexible Federated Learning Platform for Heterogeneity." In: *Proceedings of the VLDB Endowment* 16.5 (2023), pp. 1059–1072.

[41] Fan Lai, Yinwei Dai, Sanjay S. Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. "FedScale: Benchmarking Model and System Performance of Federated Learning at Scale." In: *International Conference on Machine Learning (ICML)*. 2022.

[42] Wentai Wu, Ligang He, Weiwei Lin, and Rui Mao. "Accelerating Federated Learning over Reliability-Agnostic Clients in Mobile Edge Computing Systems." In: *IEEE Transactions on Parallel and Distributed Systems* (2020), pp. 1–1. ISSN: 2161-9883. DOI: 10.1109/tpds.2020.3040867. URL: http://dx.doi.org/10.1109/TPDS.2020.3040867.

[43] Yutong Dai, Zeyuan Chen, Junnan Li, Shelby Heinecke, Lichao Sun, and Ran Xu. "Tackling data heterogeneity in federated learning with class prototypes." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6. 2023, pp. 7314–7322.

[44] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 5972–5984.

[45] NVIDIA. *NVIDIA Jetson AGX Orin Series*. 2022. URL: https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf.

[46] NVIDIA. *NVIDIA H100 Tensor Core GPU*. 2024. URL: https://resources.nvidia.com/en-us-tensor-core/nvidia-tensor-core-gpu-datasheet.

[47] Mingzhe Chen, Nir Shlezinger, H. Vincent Poor, Yonina C. Eldar, and Shuguang Cui. "Communication-efficient federated learning." In: *Proceedings of the National Academy of Sciences* 118.17 (Apr. 2021). ISSN: 1091-6490. DOI: 10.1073/pnas.2024789118. URL: http://dx.doi.org/10.1073/pnas.2024789118.

[48] Hui-Po Wang, Sebastian Stich, Yang He, and Mario Fritz. "ProgFed: Effective, communication, and computation efficient federated learning by progressive training." In: *International Conference on Machine Learning*. PMLR. 2022, pp. 23034–23054.

[49] Jakub Konečnỳ. "Federated Learning: Strategies for Improving Communication Efficiency." In: *arXiv preprint arXiv:1610.05492* (2016).

[50] Aaron Grattafiori, Abhimanyu Dubey, et al. *The Llama 3 Herd of Models*. 2024. DOI: 10.48550/ARXIV.2407.21783. URL: https://arxiv.org/abs/2407.21783.

[51] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. DOI: 10.48550/ARXIV.2106.09685. URL: https://arxiv.org/abs/2106.09685.

[52] Jed Mills, Jia Hu, and Geyong Min. "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT." In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 5986–5994. DOI: 10.1109/JIOT.2019.2956615.

[53] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. "A survey of federated learning for edge computing: Research problems and solutions." In: *High-Confidence Computing* 1.1 (June 2021), p. 100008. ISSN: 2667-2952. DOI: 10.1016/j.hcc.2021.100008. URL: http://dx.doi.org/10.1016/j.hcc.2021.100008.

[54] House Of Commons of Canada. *An Act to enact the Consumer Privacy Protection Act, the Personal Information and Data Protection Tribunal Act and the Artificial Intelligence and Data Act and to make consequential and related amendments to other Acts*. June 2022. URL: https://www.parl.ca/DocumentViewer/en/44-1/bill/C-27/first-reading.

[55] Feb. 2022. DOI: 10.1787/cb6d9eca-en. URL: http://dx.doi.org/10.1787/cb6d9eca-en.

[56] Anka Reuel, Lisa Soder, Benjamin Bucknall, and Trond Arne Undheim. "Position: Technical Research and Talent is Needed for Effective AI Governance." In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, 21–27 Jul 2024, pp. 42543–42557. URL: https://proceedings.mlr.press/v235/reuel24a.html.

[57] Tobias Müller, Milena Zahn, and Florian Matthes. "A Process Model for the Practical Adoption of Federated Machine Learning." In: *29th Americas Conference on Information Systems, AMCIS 2023, Panama City, Panama, August 10-12, 2023*. Ed. by Paul A. Pavlou, Vishal Midha, Animesh Animesh, Traci A. Carte, Alexandre R. Graeml, and Alanah Mitchell. Association for Information Systems, 2023. URL: https://aisel.aisnet.org/amcis2023/sig%5C_aiaa/sig%5C_aiaa/1.

[58] Texas State Senate. *Texas Responsible Artificial Intelligence Governance Act*. 2024. URL: https://capitol.texas.gov/tlodocs/89R/billtext/pdf/HB01709I.pdf.

[59] New York State Senate. *New York Artificial Intelligence Consumer Protection Act*. 2023. URL: https://www.nysenate.gov/legislation/bills/2023/S8209.

[60] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. *Flower: A Friendly Federated Learning Research Framework*. 2020. DOI: 10.48550/ARXIV.2007.14390. URL: https://arxiv.org/abs/2007.14390.

[61] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "LoRA: Low-Rank Adaptation of Large Language Models." In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[62] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. "BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models." In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1–9. DOI: 10.18653/v1/2022.acl-short.1. URL: https://aclanthology.org/2022.acl-short.1/.

[63] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. "Parameter-Efficient Transfer Learning for NLP." In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html.

[64] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. *FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization*. 2019. DOI: 10.48550/ARXIV.1909.13014. URL: https://arxiv.org/abs/1909.13014.

[65] Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. "Federated Full-Parameter Tuning of Billion-Sized Language Models with Communication Cost under 18 Kilobytes." In: *Forty-first International Conference on Machine Learning*. 2024. URL: https://openreview.net/forum?id=cit0hg4sEz.

[66] Feb. 2022. DOI: 10.1787/cb6d9eca-en. URL: http://dx.doi.org/10.1787/cb6d9eca-en.

[67] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. "FedBN: Federated Learning on Non-IID Features via Local Batch Normalization." In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=6YEQUn0QICG.

[68] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. "The non-IID data quagmire of decentralized machine learning." In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020.

[69] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Ananda Theertha Suresh, Dave Bacon, and Peter Richtárik. *Federated Learning: Strategies for Improving Communication Efficiency*. 2018. URL: https://openreview.net/forum?id=B1EPYJ-C-.

[70] Yuang Jiang, Shiqiang Wang, Víctor Valls, Bong Jun Ko, Wei-Han Lee, Kin K. Leung, and Leandros Tassiulas. "Model Pruning Enables Efficient Federated Learning on Edge Devices." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.12 (2023), pp. 10374–10386. DOI: 10.1109/TNNLS.2022.3166101.

[71] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. "PAPAYA: Practical, Private, and Scalable Federated Learning." In: *MLSys*. 2022. URL: https://proceedings.mlsys.org/paper_files/paper/2022/hash/a8bc4cb14a20f20d1f96188bd61eec87-Abstract.html.

[72] Lumin Liu, Jun Zhang, S.H. Song, and Khaled B. Letaief. "Client-Edge-Cloud Hierarchical Federated Learning." In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148862.

[73] Chengxu Yang, Qipeng Wang, Mengwei Xu, Zhenpeng Chen, Kaigui Bian, Yunxin Liu, and Xuanzhe Liu. "Characterizing Impacts of Heterogeneity in Federated Learning upon Large-Scale Smartphone Data." In: *Proceedings of the Web Conference 2021*. WWW '21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 935–946. ISBN: 9781450383127. DOI: 10.1145/3442381.3449851. URL: https://doi.org/10.1145/3442381.3449851.

[74] Canh T. Dinh, Nguyen H. Tran, Minh N. H. Nguyen, Choong Seon Hong, Wei Bao, Albert Y. Zomaya, and Vincent Gramoli. "Federated Learning Over Wireless Networks: Convergence Analysis and Resource Allocation." In: *IEEE/ACM Trans. Netw.* 29.1 (Feb. 2021), pp. 398–409. ISSN: 1063-6692. DOI: 10.1109/TNET.2020.3035770. URL: https://doi.org/10.1109/TNET.2020.3035770.

[75] Michael Crawshaw and Mingrui Liu. "Federated Learning under Periodic Client Participation and Heterogeneous Data: A New Communication-Efficient Algorithm and Analysis." In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024. URL: https://openreview.net/forum?id=WftaVkL6G2.

[76] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. "Adaptive Federated Optimization." In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=LkFG3lB13U5.

[77] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. "On the importance of initialization and momentum in deep learning." In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147. URL: https://proceedings.mlr.press/v28/sutskever13.html.

[78] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[79] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. "Federated Learning: Challenges, Methods, and Future Directions." In: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60. DOI: 10.1109/MSP.2020.2975749.

[80] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. "Advances and Open Problems in Federated Learning." In: *Found. Trends Mach. Learn.* 14.1–2 (June 2021), pp. 1–210. ISSN: 1935-8237. DOI: 10.1561/2200000083. URL: https://doi.org/10.1561/2200000083.

[81] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. "Federated Machine Learning: Concept and Applications." In: *ACM Trans. Intell. Syst. Technol.* 10.2 (Jan. 2019). ISSN: 2157-6904. DOI: 10.1145/3298981. URL: https://doi.org/10.1145/3298981.

[82] Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. "Towards Federated Learning at Scale: System Design." In: *Proceedings of the Second Conference on Machine Learning and Systems, SysML 2019, Stanford, CA, USA, March 31 - April 2, 2019*. Ed. by Ameet Talwalkar, Virginia Smith, and Matei Zaharia. mlsys.org, 2019. URL: https://proceedings.mlsys.org/paper%5C_files/paper/2019/hash/7b770da633baf74895be22a8807f1a8f-Abstract.html.

[83] Arun Vishwanath, Fatemeh Jalali, Kerry Hinton, Tansu Alpcan, Robert W. A. Ayre, and Rodney S. Tucker. "Energy Consumption Comparison of Interactive Cloud-Based and Local Applications." In: *IEEE Journal on Selected Areas in Communications* 33.4 (Apr. 2015), pp. 616–626. ISSN: 0733-8716. DOI: 10.1109/jsac.2015.2393431. URL: http://dx.doi.org/10.1109/JSAC.2015.2393431.

[84] Pengchao Han, Shiqiang Wang, and Kin K. Leung. " Adaptive Gradient Sparsification for Efficient Federated Learning: An Online Learning Approach." In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2020, pp. 300–310. DOI: 10.1109/ICDCS47774.2020.00026. URL: https://doi.ieeecomputersociety.org/10.1109/ICDCS47774.2020.00026.

[85] Pedro Valdeira, João Xavier, Cláudia Soares, and Yuejie Chi. "Communication-efficient Vertical Federated Learning via Compressed Error Feedback." In: *2024 32nd European Signal Processing Conference (EUSIPCO)*. 2024, pp. 1037–1041. DOI: 10.23919/EUSIPCO63174.2024.10715377.

[86] Samuel Horváth and Peter Richtarik. "A Better Alternative to Error Feedback for Communication-Efficient Distributed Learning." In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=vYVI1CHPaQg.

[87] Farzad Samie, Lars Bauer, and Jörg Henkel. "IoT technologies for embedded computing: a survey." In: *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*. CODES '16. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2016. ISBN: 9781450344838. DOI: 10.1145/2968456.2974004. URL: https://doi.org/10.1145/2968456.2974004.

[88] Emna Baccour, Naram Mhaisen, Alaa Awad Abdellatif, Aiman Erbad, Amr Mohamed, Mounir Hamdi, and Mohsen Guizani. "Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence." In: *IEEE Communications Surveys & Tutorials* 24.4 (2022), pp. 2366–2418. DOI: 10.1109/COMST.2022.3200740.

[89] Fatemeh Jalali, Rob Ayre, Arun Vishwanath, Kerry Hinton, Tansu Alpcan, and Rod Tucker. "Energy Consumption of Content Distribution from Nano Data Centers versus Centralized Data Centers." In: *ACM SIGMETRICS Performance Evaluation Review* 42.3 (Dec. 2014), pp. 49–54. ISSN: 0163-5999. DOI: 10.1145/2695533.2695555. URL: http://dx.doi.org/10.1145/2695533.2695555.

[90] Haftay Gebreslasie Abreha, Mohammad Hayajneh, and Mohamed Adel Serhani. "Federated Learning in Edge Computing: A Systematic Survey." In: *Sensors* 22.2 (Jan. 2022), p. 450. ISSN: 1424-8220. DOI: 10.3390/s22020450. URL: http://dx.doi.org/10.3390/s22020450.

[91] Anup Das, Akash Kumar, and Bharadwaj Veeravalli. "Energy-aware task mapping and scheduling for reliable embedded computing systems." In: *ACM Trans. Embed. Comput. Syst.* 13.2s (Jan. 2014). ISSN: 1539-9087. DOI: 10.1145/2544375.2544392. URL: https://doi.org/10.1145/2544375.2544392.

[92] Qing Li and Caroline Yao. *Real-Time Concepts for Embedded Systems*. Jan. 2003. DOI: 10.1201/9781482280821. URL: http://dx.doi.org/10.1201/9781482280821.

[93] Francky Catthoor, Sven Wuytack, Eddy De Greef, Florin Balasa, Lode Nachtergaele, and Arnout Vandecappelle. *Custom Memory Management Methodology*. Springer US, 1998. ISBN: 9781475728491. DOI: 10.1007/978-1-4757-2849-1. URL: http://dx.doi.org/10.1007/978-1-4757-2849-1.

[94] Yujeong Choi and Minsoo Rhu. "PREMA: A Predictive Multi-Task Scheduling Algorithm For Preemptible Neural Processing Units." In: *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 2020, pp. 220–233. DOI: 10.1109/HPCA47549.2020.00027.

[95] Zixiao Wang, Biyao Che, Liang Guo, Yang Du, Ying Chen, Jizhuang Zhao, and Wei He. "PipeFL: Hardware/Software co-Design of an FPGA Accelerator for Federated Learning." In: *IEEE Access* 10 (2022), pp. 98649–98661. DOI: 10.1109/ACCESS.2022.3206785.

[96] NVIDIA. *NVIDIA H100 Tensor Core GPU Architecture Overview — resources.nvidia.com*. https://resources.nvidia.com/en-us-tensor-core?ncid=no-ncid. [Accessed 21-02-2025]. 2023.

[97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: an imperative style, high-performance deep learning library." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[98] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeed. "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications." In: *IEEE Access* 8 (2020), pp. 140699–140725. ISSN: 2169-3536. DOI: 10.1109/access.2020.3013541. URL: http://dx.doi.org/10.1109/access.2020.3013541.

[99] Zhan-Lun Chang, Seyyedali Hosseinalipour, Mung Chiang, and Christopher G. Brinton. "Asynchronous Multi-Model Dynamic Federated Learning Over Wireless Networks: Theory, Modeling, and Optimization." In: *IEEE Transactions on Cognitive Communications and Networking* 10.5 (2024), pp. 1989–2004. DOI: 10.1109/TCCN.2024.3391329.

[100] Raspberry Foundation. *Raspberry Pi 5*. https://www.raspberrypi.com/products/raspberry-pi-5/. [Accessed 21-02-2025]. 2024.

[101] Ollama. *GitHub - ollama/ollama: Get up and running with Llama 3.3, DeepSeek-R1, Phi-4, Gemma 2, and other large language models. — github.com*. https://github.com/ollama/ollama. [Accessed 21-02-2025]. 2024.

[102] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. "Efficient Memory Management for Large Language Model Serving with PagedAttention." In: *Proceedings of the 29th Symposium on Operating Systems Principles*. SOSP '23. Koblenz, Germany: Association for Computing Machinery, 2023, pp. 611–626. ISBN: 9798400702297. DOI: 10.1145/3600006.3613165. URL: https://doi.org/10.1145/3600006.3613165.

[103] Kiran Seshadri, Berkin Akin, James Laudon, Ravi Narayanaswami, and Amir Yazdanbakhsh. " An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks." In: *2022 IEEE International Symposium on Workload Characterization (IISWC)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2022, pp. 79–91. DOI: 10.1109/IISWC55918.2022.00017. URL: https://doi.ieeecomputersociety.org/10.1109/IISWC55918.2022.00017.

[104] Intel. *Intel® Neural Compute Stick 2 – Produktspezifikationen | Intel — intel.de*. https://www.intel.de/content/www/de/de/products/sku/140109/intel-neural-compute-stick-2/specifications.html. [Accessed 21-02-2025]. 2016.

[105] Nuria Oliver, Alexander Peukert, et al. *First Draft General-Purpose AI Code of Practice*. Nov. 2024. URL: https://ec.europa.eu/newsroom/dae/redirection/document/109946.

[106] United Kingdom Government. *A pro-innovation approach to AI regulation — gov.uk*. https://www.gov.uk/government/publications/ai-regulation-a-pro-innovation-approach/white-paper. [Accessed 21-02-2025]. 2023.

[107] The White House. *Initial Rescissions Of Harmful Executive Orders And Actions*. 2025. URL: https://www.whitehouse.gov/presidential-actions/2025/01/initial-rescissions-of-harmful-executive-orders-and-actions/.

[108] The White House. *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence*. Oct. 2023. URL: https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and%20-trustworthy-development-and-use-of%20-artificial-intelligence/.

[109] The White House. *Executive Order on Removing Barriers To Americal Leadership In Artificial Intelligence*. Jan. 2025. URL: https://www.whitehouse.gov/presidential-actions/2025/01/removing-barriers-to-american-leadership-in-artificial-intelligence/.

[110] Steve Holland. *Trump announces private-sector $500 billion investment in AI infrastructure*. 2025. URL: https://www.reuters.com/technology/artificial-intelligence/trump-announce-private-sector-ai-infrastructure-investment-cbs-reports-2025-01-21/.

[111] China Law Translate. *Interim Measures for the Management of Generative Artificial Intelligence Services*. July 2023.

[112] Ashkan Yousefpour, Shen Guo, Ashish Shenoy, Sayan Ghosh, Pierre Stock, Kiwan Maeng, Schalk-Willem Krüger, Michael Rabbat, Carole-Jean Wu, and Ilya Mironov. *Green Federated Learning*. 2023. DOI: 10.48550/ARXIV.2303.14604. URL: https://arxiv.org/abs/2303.14604.

[113] Galen Andrew, Om Thakkar, Hugh Brendan McMahan, and Swaroop Ramaswamy. "Differentially Private Learning with Adaptive Clipping." In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan. 2021. URL: https://openreview.net/forum?id=RUQ1zwZR8_.

[114] Liangqi Yuan, Ziran Wang, and Christopher G. Brinton. "Digital Ethics in Federated Learning." In: *IEEE Internet Computing* 28.5 (2024), pp. 66–74. DOI: 10.1109/MIC.2024.3370408.

Appendix A

FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems

Printed with the permission of

Herbert Woisetschläger, Alexander Erben, Ruben Mayer, Shiqiang Wang, and Hans-Arno Jacobsen. "FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems." In: *Proceedings of the 25th International Middleware Conference.* Middleware '24. Hong Kong, Hong Kong: Association for Computing Machinery, 2024, pp. 88–102. ISBN: 9798400706233. DOI: 10.1145/3652892.3700751. URL: https://doi.org/10.1145/3652892.3700751

FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems

Herbert Woisetschläger
h.woisetschlaeger@tum.de
Technical University of Munich
Germany

Alexander Erben
alex.erben@tum.de
Technical University of Munich
Germany

Ruben Mayer
ruben.mayer@uni-bayreuth.de
University of Bayreuth
Germany

Shiqiang Wang
wangshiq@us.ibm.com
IBM Research
United States

Hans-Arno Jacobsen
jacobsen@eecg.toronto.edu
University of Toronto
Canada

Abstract

Federated Learning (FL) has become a viable technique for realizing privacy-enhancing distributed deep learning on the network edge. Heterogeneous hardware, unreliable client devices, and energy constraints often characterize edge computing systems. In this paper, we propose FLEdge, which complements existing FL benchmarks by enabling a systematic evaluation of client capabilities. We focus on computational and communication bottlenecks, client behavior, and data security implications. Our experiments with models varying from 14K to 80M trainable parameters are carried out on dedicated hardware with emulated network characteristics and client behavior. We find that state-of-the-art embedded hardware has significant memory bottlenecks, leading to 4× longer processing times than on modern data center GPUs.

CCS Concepts: • **Computing methodologies** → **Distributed artificial intelligence**; • **General and reference** → **Performance**.

Keywords: Federated Learning, Performance Benchmark

ACM Reference Format:
Herbert Woisetschläger, Alexander Erben, Ruben Mayer, Shiqiang Wang, and Hans-Arno Jacobsen. 2024. FLEdge: Benchmarking Federated Learning Applications in Edge Computing Systems. In *24th International Middleware Conference (MIDDLEWARE '24), December 2–6, 2024, Hong Kong, Hong Kong.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3652892.3700751

MIDDLEWARE '24, December 2–6, 2024, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-0623-3/24/12
https://doi.org/10.1145/3652892.3700751

1 Introduction

Federated Learning (FL) has become an established middleware abstraction to facilitate distributed and privacy-preserving deep learning (DL) [6, 23, 60]. With increasing access to data by eliminating the need to transfer data to a central location, FL reduces network communication and fosters privacy. Privacy in FL applications is typically realized by introducing differential privacy [2, 42]. There is ample research on benchmarking FL workloads, which focuses on different fields of application, data heterogeneity [6, 23], FL aggregation strategies [23, 35], DL model personalization [9], and federated hyperparameter optimization [66]. However, these benchmarks typically target server-grade hardware and are simulation-based [6, 23]. As such, an often neglected factor is the type of system FL workloads are deployed to: edge computing systems.

Edge computing is a technology for managing a fleet of devices distributed across geographies [57], serving latency-critical or data-sensitive tasks [24, 50]. Systems at the edge are characterized by diverse hardware and low client reliability [13], varying network quality [22, 36], and energy constraints [30]. Additionally, FL workloads are prone to a high degree of data heterogeneity [18, 43, 54]. This creates a set of challenges when placing FL workloads at the edge since servers and clients are strongly intertwined. While clients facilitate the training on their local data, servers use FL strategies (e.g., FedAvg) to aggregate the trained client models into a global model that generalizes over all clients.

As clients in edge computing systems are often embedded devices, their computational capabilities and points of most energy-efficient operations vary. Additionally, devices can be more than 5 years old in current systems that require high functional safety [14, 53]. Older devices usually do not have DL accelerating hardware, while more recently introduced platforms are powerful system-on-a-chip (SoC) devices carrying a GPU [46, 48], leading to a major gap in computational capabilities. In practice, this requires careful deployment planning for FL workloads, especially when large language models (LLMs) are involved [15].

FLEdge extends existing FL benchmarking works by introducing systematic studies on client behavior, communication and energy efficiency, and hardware diversity. As these dimensions are critical to FL system design, we help researchers and practitioners develop new FL efficiency methods and drive practical adoption of FL, especially in edge computing systems.

Client behavior. State-of-the-art benchmarks for FL applications provide a variety of DL models, datasets, and FL strategies [6, 23, 69]. Yet, no work systematically studies the effects of client dropouts on the overall training performance in real systems. Client dropouts will likely happen in edge computing systems where clients and network connectivity can be unstable [58], which makes it an integral part to analyze, especially with regard to the interference effects with differential privacy (DP) that depend on the number of clients actually submitting model updates during a training round.

Communication efficiency. Equally important to the client behavior are network conditions, as it is an integral component for an efficient edge deployment. However, many contributions discuss communication efficiency in FL systems [32, 41, 51, 68] without providing metrics that help estimate the viability of a workload to be federated into an edge computing system with wireless communication, like 4G [1].

Energy efficiency. Existing benchmark works predominantly focus on computational speed [35, 69]. However, two key design variables in edge computing systems are computational and energy efficiency, which both contribute to the overall energy efficiency of an FL system. Energy is a scarce resource in edge computing systems and ultimately determines how fast we can process a workload depending on client hardware.

Hardware diversity. While simulated embedded devices can be used to evaluate the scalability of FL algorithms in depth, they neglect the underlying processor architecture and memory bandwidth. This results in significant performance differences when deploying on different types of hardware and limits the applicability to edge computing systems [5, 23, 57].

Overall, FLEdge aims to improve the understanding of FL application deployment in edge computing systems and specifically answers the following research question:

> *How well do state-of-the-art FL workloads respond to deployment in edge computing systems?*

Our contributions are as follows:

(1) **We introduce FLEdge – a hardware-centric benchmarking suite for FL workloads in system-heterogeneous environments**. FLEdge extends the widely used FL framework Flower [5] with a module to control client behavior, an adaptive user-level differential privacy adapter that accounts for client dropouts, a network emulator, and extensive monitoring capabilities to evaluate the computational efficiency of embedded devices. Our code base is publicly available.[1]
(2) **FLEdge provides a holistic evaluation pipeline for FL workloads w.r.t. client behavior and communication efficiency**. The client behavior module allows flexible modeling of client reliability based on environmental conditions and the intended deployment target. This enables practitioners to quickly evaluate the robustness of their FL workloads. We also add a freely controllable network adapter that allows for the emulation of realistic connectivity and the exploration of the viability of deploying FL workloads in edge computing systems.
(3) **We conduct extensive experiments on computational capabilities and energy efficiency of widely used embedded devices when running FL workloads**. Our experimental results show that the state-of-the-art embedded AI accelerator is challenged with backpropagation, which calls for alternative solutions that better use the hardware architecture. Our experiments on client behavior in conjunction with differentially private FL workloads show a high sensitivity of model quality w.r.t. client reliability.

Our work is structured as follows. In Section 2, we outline the requirements analysis for our benchmark. In Section 3, we introduce our methodology. In Section 4, we outline our experimental design, including datasets, DL models, FL strategies, and practical considerations. Section 5 discusses experimental results of our benchmark. Section 6 contains related work. Section 7 discusses lessons learned and in Section 8, we conclude our work.

2 Requirement Analysis

As edge computing applications are gaining popularity, especially in realistic environments without perfect control over system parameters, challenges in designing efficient systems arise. On a client, we care about energy efficiency and training reliability, as well as privacy, such that the data is processed in a timely and secure manner. At the same time, there are often (mobile) clients with heterogeneous computing resources involved, which may negatively affect the training speed. That said, we need a hardware-centric benchmarking framework to assess the feasibility of FL applications on the network edge. Furthermore, as FL aims to be a communication-efficient ML paradigm, cost and efficiency are key decision variables. The objective of our work is to evaluate the end-to-end FL pipeline, investigating system components that have come short in existing research:

[1] https://github.com/laminair/FLEdge.

Client behavior. Generally, client reliability is key for a distributed application. While for most edge computing applications, tasks run individually on clients, e.g., in the context of CV applications with MobileNet models [26], for FL, there are significant dependencies between clients and the server. FL typically involves an iterative training procedure where a set of N clients trains a shared model over multiple rounds $t \in T$. In each round, a subset $M \in N$ clients is selected for training. Training an FL model involves aggregating multiple client updates into a single model: $w^{t+1} = \frac{1}{|M|} \sum_{m \in M} w_m^t$. For this, we use FL strategies like FedAvg [41]. In edge computing, clients are unreliable compared to data center services and may show a high failure rate. Thus, M can vary within a training round, and consequently, fewer client updates will be aggregated. As such, it is a requirement to test the robustness of state-of-the-art FL strategies with varying realistic client behavior typically found in edge computing systems.

Privacy. Another key concern for FL systems is data privacy, as clients are unwilling to share their data. FL already increases the level of data privacy to a certain extent [41] but does not provide a *guarantee* for privacy to clients. Formal methods like (ϵ, δ)- Differential Privacy (DP) can be applied to obtain such a guarantee. DP can be applied on sample- or user-level, where the latter is of special interest for FL [2, 42]. As clients in FL systems only share their model updates, it is sufficient to apply DP to model updates and achieve the same guarantees as with sample-level DP [67]. The objective of user-level DP is to deny membership inference or gradient inversion attacks. This only works if the noise level is calculated appropriately based on the number of clients that actually submit a model update in an FL training round. As such, bringing together robustness and data privacy requires user-level DP to adapt to suddenly failing clients is a requirement.

Communication efficiency. Along with varying client behavior and privacy levels, communication is an integral part of ensuring a high service quality of FL workloads in edge computing systems. In many use cases, tasks are running independently on clients [58]. However, in FL systems, there is a strong coupling between clients and the server as the global model state is maintained on the server. Therefore, we are interested in the processing latency and every aspect of network communication for FL on the edge. Typically, there are three scenarios for deploying edge networks. The first and most reliable is a wired connection, often found in factories [7]. The second is a high-bandwidth wireless connection, such as 4G LTE [1], and the third is a low-bandwidth and high latency connection, often found in remote areas [29, 58]. The anticipation of network connectivity is key as to whether it makes sense to federate the training process for a given DL model. Interestingly, existing works measure network conditions in different ways. Some focus on available bandwidth [45, 70], and others focus on the amount of data transmitted [61, 64]. As such, there is a necessity for a comprehensive and unifying metric that quickly answers the question of whether it is viable to deploy an FL workload into an edge computing system with heterogeneous network conditions based on the amount of time we use on computation vs. communication.

Communication cost. FL applications in edge computing systems are often operated over wide-area networks [63], involving a multitude of network hops. While it is theoretically possible to measure and report the energy cost per hop, it is challenging to achieve in a real-world system since each networking component is often owned by a different entity, such as the client owner, the internet service provider, and the cloud operator. Thus, we require a solution to reliably estimate the total communication cost of an FL workload.

Energy efficiency. To evaluate energy efficiency, we need a reliable method to evaluate the characteristics of existing devices for FL workloads. In edge deployments, clients have previously been employed as a data collection platform only and are now required to run computationally intensive FL workloads. This substantially increases energy consumption, which is often a challenge due to limited power availability [11]. To mitigate these effects, hardware acceleration on embedded devices has been introduced [46]. For instance, GPUs generally provide a significant performance benefit over CPUs while being more energy efficient and cost-saving at the same time [55]. Therefore, it is a requirement to investigate the energy efficiency of different hardware w.r.t. their projected workloads. While DL workloads in the cloud are hard to measure for their energy efficiency due to virtualized hardware [19], the benefit of embedded devices is their SoC design that allows for measurement of the real-time energy consumption of the entire device and all individual device components (e.g., CPU, GPU).

Hardware diversity. Edge computing is designed around the pattern of offloading computational tasks to edge devices, usually embedded devices in the proximity of a data source. This provides better control over the trade-off between computation and communication [39]. Yet, a major challenge in these systems is hardware diversity, either due to long product life cycles in industrial systems or due to a lack of infrastructure control. For instance, widely used embedded devices for industrial edge require extensive reliability testing and therefore include hardware as old as five years [53]. As such, it is necessary to benchmark not only the most recently released devices but also older generations still being used in a wide variety of systems [65]. Also, embedded devices are becoming increasingly capable as modern platforms, such as the NVIDIA Jetson AGX Orin, often entail integrated GPUs. This makes benchmarking hardware more complex and is also likely to change the energy demand of devices for workloads. A hardware-centric benchmark must, therefore, account for a variable number of components on an SoC-based embedded device.

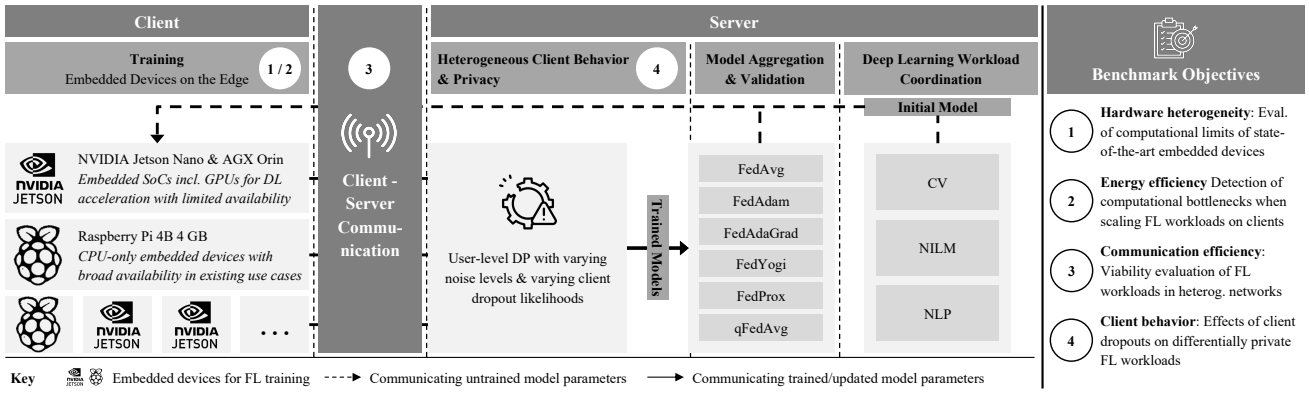


Figure 1. Federated Learning protocol for one training round from a system perspective with 1 - 4 indicating the focus areas for our work and the benchmark subjects for FLEdge. Our system uses Flower as the underlying FL framework and implements each component in a modular and extensible manner.

3 FLEdge: Benchmarking Framework

Client behavior. As robustness is important to all FL workloads, and edge computing systems suffer from heterogeneous client behavior, we configure our testbed to simulate realistic client failures. For FL workloads, it does not matter whether the failure origin is hardware or communication. We model the client dropout as an independent binomial distribution since we assume clients will reenter training for future rounds and clients do not interfere with one another,

$$p_m^d = P(m) = \mathrm{Bin}(p). \tag{1}$$

We vary the failure likelihood p from 0% to 50% in our experiments, with 0% being very reliable clients as they are often found in factories and 50% representing unreliable clients, e.g., mobile clients with wireless connetion. For example, the FL strategy FedAvg is extended with p_m^d as follows,

$$w^{t+1} = \frac{1}{|M|} \sum_{m \in M} (w_m^t \cdot p_m^d). \tag{2}$$

The behavioral pattern can be freely exchanged for other patterns that suit particular use cases.

Communication efficiency. To transfer the model updates from the clients to the server, we need to consider communication efficiency and find a method to express the viability of adding an embedded computing platform under given network conditions. To evaluate communication efficiency, we use granularity [27], which is an established metric for evaluating the training efficiency in a distributed system by comparing the computation time against the communication time,

$$G = \frac{T_{\text{computation}}}{T_{\text{communication}}}. \tag{3}$$

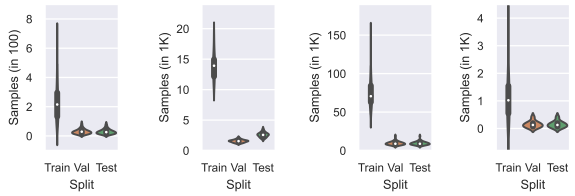
$G \gg 1$ is considered favorable for distributing workloads, while $G \simeq 1$ or $G < 1$ indicates little utility when distributing a workload to a given system or device. Our benchmark allows the emulation of different realistic connectivity profiles per client to explore the effect of G.

Communication cost. It is important to consider the communication costs of scaling an FL system since we operate in a data-parallel regime. While theoretically, we can scale the number of clients such that we fully saturate the server's network bandwidth, the effects of a large number of clients are limited [8]. However, to establish the scalability-cost trade-off, we employ the *per-bit communication cost model* that allows us to assert a client with constant communication costs per model update transmission [28]. The model allows us to calculate the energy we consume at every hop in a network,

$$\begin{aligned} P_{\text{t}} = E_t \cdot \mathcal{B} = (n_{\text{as}} \cdot E_{\text{as}} + n_{\text{lc}} \cdot E_{\text{lc}} + n_{\text{lb}} \cdot E_{\text{lb}} + E_{\text{bng}} \\ + n_e \cdot E_e + n_c \cdot E_c + n_d \cdot E_d) \cdot \mathcal{B}. \end{aligned} \tag{4}$$

E_{as}, E_{lc}, E_{lb}, E_{bng}, E_e, E_c, E_d are the per-bit energy consumption of n_{as} edge ethernet switches, zero or one n_{lc} LTE client modem, zero or one n_{lb} LTE base station, the broadband network gateway (BNG), one or more edge routers n_e, one or more core routers n_c, and one or more data center Ethernet switches n_d, respectively. $\mathcal{B}$ denotes the size of a model update in bits.

Computational efficiency. Aside from communication in edge computing systems, energy is a vital component of system efficiency as it may be a scarce resource in remote areas. Therefore, our testbed contains real-time measurement capabilities and control mechanisms to limit the power draw of each embedded device. This enables us to explore the deployment characteristics of FL workloads in a wide variety of edge computing systems. We measure the energy efficiency as the throughput in samples per second (denoted



(a) BLOND **(b)** FEMNIST **(c)** Shakesp. **(d)** Samsum

Figure 2. The non-IID subsets for our clients are sampled from a Dirichlet distribution ($\alpha = 1$).

as Q) divided by the average power draw (denoted as W) over the experiment time:

$$\eta_e = \frac{\mathrm{Q}}{W}. \tag{5}$$

Hardware diversity. To develop a detailed understanding of where potential inefficiencies could come from, we employ a micro-benchmark to study the effect of hardware diversity on the training performance of FL workloads. It focuses on the timing of the DL step times, namely the batch loading, forward, loss calculation, backward, and optimizer steps. With this, we get a detailed understanding of computational inefficiencies on embedded devices and uncover differences to data center hardware that potentially become bottlenecks for state-of-the-art FL workloads. Additionally, we use the PyTorch profiler with Kineto support to investigate performance bottlenecks in our FL clients. This provides us with the runtime of individual kernels, highlighting potential bottlenecks.

3.1 Protocol

Our setup follows the widely used client-server architecture for FL workloads [6, 23, 69] and is depicted in Figure 1. The four focus areas and their practical utility are discussed in the following.

Training. Training is facilitated entirely by the clients. They receive the hyperparameter configuration along with the model parameters from the server and train for exactly 1 local epoch before communicating the updates to the server. We do this for our client dropout and privacy experiments to isolate client failure effects on model quality and avoid any interference with data drift [31, 44].

Communication. We integrate a controllable network interface on each client as edge computing systems that involve embedded devices can be found in various environments (e.g., production lines or remote weather sensing stations). In addition to the steady 1 Gbit/s network link, which resembles industrial settings, our testbed allows us to emulate wireless communication (e.g., LTE, 3G) per client for remote setups.

Privacy. A core promise of FL is to preserve privacy. For user-level DP, the model aggregation method is extended by adding Gaussian noise to the model weights (cf. Equation (6)).

Table 1. The datasets in our benchmark vary in modality, size, and data heterogeneity to resemble a large variety of real-world use cases.

Pipeline	Dataset Name	Total Samples	Samples per Device	Format	Size
NILM	BLOND	13,164	234±132	HDF5	5.46 GB
CV	FEMNIST	814,255	13,958±2,106	PNG	3.34 GB
NLP	Shakespeare	4,226,054	75,131±18,281	TXT	0.38 GB
NLP	Samsum	11,780	491±723	TXT	0.012 GB

With the additional noise $\xi \sim \mathcal{N}(0, I\sigma_\Delta^2)$, we introduce a natural trade-off between model accuracy and the level of privacy. The variance σ_Δ^2 depends on the number of clients sampled in a training round and on their L2 update norm, as with more heterogeneous client updates, more noise has to be added to ensure (ϵ , δ) - DP ,

$$w^{t+1} = \frac{1}{|M|} \sum_{m \in M} ((w_m^t + z \cdot \xi) \cdot p_m^d) \tag{6}$$

The strength of the privacy guarantee is measured by the privacy budget ϵ where lower values are better and provide higher levels of privacy [16]. ϵ depends on δ, the likelihood of unintentional input data leakage. Usually, δ is set to a value equal to the inverse of the anticipated total number of samples in a dataset (e.g., for the BLOND dataset, it would amount to $\delta = \frac{1}{13{,}164}$, see Section 4.3). As such, our system entails server-side user-level DP. In this way, we can adjust the DP noise levels on the server based on the number of updates received rather than having to request a partial recalculation of DP model updates from each client.

3.2 Practical Assumptions & Configuration

As we are interested in the performance of FL workloads, we configure our testbed to serve under realistic conditions for industrial edge computing systems. We describe the configuration along Figure 1.

Training. Clients are dedicated to the FL workload only and assign all available resources to the task with the respective optimal hyperparameter configuration as depicted in Table 3. We choose a client participation rate of 20% for all experiments, but it can be freely configured for other scenarios.

Communication. The network is configured to either emulate a factory environment with a synchronous 1 Gbit/s link [53] or to run a client-specific 4G LTE wireless [1] network with higher latency and asynchronous 15 MBit/s upload and 40 MBit/s download bandwidths that is often found when using mobile or remotely embedded devices. The LTE bandwidths represent the global average connectivity [56]. For the per-bit energy consumption model, we adopt the measurements and network topology from Jalali et al. [28], Vishwanath et al. [59] and consider two distinct scenarios for our communication cost analysis P_t: (I) a wired 1 Gbit/s interconnect as it can be found in, e.g., factories, and (II) a

Table 2. FL strategy hyperparameters. Key: η, η_l = server-side learning rates, β_1, β_2 = Weight decay rates, τ = Server-side adaptivity level, q = Fairness parameter for qFedAvg, μ = proximal parameter for FedProx.

FL Strategy	Dataset	η	η_l	β_1	β_2	τ	q	μ	Further Details
FedAvg	All	-	-	-	-	-	-	-	FedAvg does not have any server hyperparameters
FedAdam	All	-1.5	-1	0.9	0.99	$1e^{-2}$	-	-	
FedAdaGrad	All	0	0	-	-	$1e^{-3}$	-	-	
FedYogi	All	-1.5	-1.5	0.9	0.99	$1e^{-5}$	-	-	
FedProx	Shakespeare	-	-	-	-	-	-	$1e^{-2}$	
	Others	-	-	-	-	-	-	1	
qFedAvg	Shakespeare	-	-	-	-	-	$1e^{-2}$	-	
	Others	-	-	-	-	-	1	-	

wireless setup where clients connect via an LTE gateway. For (I), we use a network topology with $n_{\text{as}} = 2$, $n_{\text{lc}} = 0$, $n_{\text{lb}} = 0$, $n_{\text{bng}} = 1$, $n_{\text{e}} = 3$, $n_{\text{c}} = 4$, $n_{\text{d}} = 2$. For (II), we set $n_{\text{as}} = 0$, $n_{\text{lc}} = 1$, $n_{\text{lb}} = 1$, $n_{\text{bng}} = 1$, $n_{\text{e}} = 4$, $n_{\text{c}} = 4$, $n_{\text{d}} = 2$.

Privacy. The server is set up to handle both workloads that require very light privacy guarantees, i.e., for workloads that deal with non-sensitive data and for workloads ($\epsilon > 5$) that require very tight guarantees ($\epsilon < 1$) [2, 42].

Client behavior. The client dropout module is designed to account for a wide variety of use cases that create different wear and tear on embedded devices and reduce their reliability. Typically, in industrial environments, service quality is essential. Thus, the availability times often range well above 95% of the operation time [3]. With our study, we not only account for these services but also look into the effects of clients with very low reliability as they are found in loose collaborative learning tasks [25]. To do so, we vary the client dropout likelihood from 0% to 50%.

Model aggregation & validation. The server runs state-of-the-art FL strategies and is configured according to related work. The detailed parameterization is available in Table 2.

DL workload coordination. The server is set up to coordinate state-of-the-art FL workloads that you would find in systems with both high and low reliability to provide a holistic picture of use cases.

4 Experimental Setup

Our hardware-centric benchmark is deployed to a dedicated edge computing testbed. We use it to explore FL workloads in resource-constrained environments and explore the effects of power and network limitations on FL performance. For this benchmark, we deploy widely used state-of-the-art datasets from the NILM, CV, and NLP domains for FL applications, including an LLM workload.

4.1 Testbed

We aim to explore FL applications in real systems by introducing two data center and three different embedded device types.

Table 3. Our hyperparameters are chosen based on related work and hyperparameter sweeps for the best possible performance.

Dataset	Model	Optim.	LR	W. Decay	Dropout	Hidden Dim	Params	Minibatch
BLOND	CNN	SGD	0.055	0.0			14,000	128
	LSTM	SGD	0.045	0.001	0	15	40,000	128
	ResNet	SGD	0.052	0.001			100,000	128
	DenseNet	SGD	0.075	0.001			252,000	128
FEMNIST	CNN	Adam	0.001	0.0			33,000	32
Shakespeare	LSTM	SGD	0.8	0.0	0	256	819,000	32
Samsum	FLAN-T5-Small	AdamW	0.0001	0.0	0	N/A	80,000,000	1 - 128

Data center GPU (GPU). We use a GPU-accelerated data center node with 64 CPU cores, 256 GB of memory, and an NVIDIA A6000 (GPU) as our baseline device. The NVIDIA A6000 has a memory bandwidth of 768 GB/s. The VM has 3 TB NVMe storage. The VM is running Ubuntu 20.04 LTS with Python 3.9 and PyTorch 1.10. We use CUDA 11.6 and cuDNN 8.6.

x86-CPU-base Clients (VM). As a proxy for x86-based embedded devices without DL acceleration, we use virtualized systems with 4 CPU cores and 4 GB of memory (VM). The VM has a memory bandwidth of 25.6 GB/s and a network interconnect of 1 GBit/s. We use Ubuntu 20.04 LTS with Python 3.9 and PyTorch 1.10. We use the estimates from SelfWatts [19] as the power utilization profile for each VM. Fieni et al. calculate approx. 50 Watts for a 4 CPU core VM on Intel Xeon E5 processors.

NVIDIA Jetson AGX Orin 64GB (Orin). Our Orins are running Jetpack 5.1. This includes Ubuntu 20.04 LTS with Python 3.8, PyTorch 1.13, and CUDA 11.8. As the libraries are compiled specifically for the platform, we cannot adjust the stack to older PyTorch versions. The Orins have a memory bandwidth of 204 GB/s, a disk size of 64 GB, and a 10 Gbit/s network connection. They come with 2048 CUDA cores and 64 Tensor Cores. We measure power via the internal hardware-based monitoring functionality.

NVIDIA Jetson Nano 2GB (Nano). The devices run Ubuntu 18.04 LTS with Python 3.6 and PyTorch 1.10.[2] The Nano has a memory bandwidth of 25.6 GB/s. The latest supported CUDA version is 10.2 with cuDNN 7.2. The Nanos carry 128 CUDA cores but no Tensor cores. The maximum power draw of a Nano is 15 Watts.

Raspberry Pi 4B 4GB (RPi). They run with Ubuntu 20.04 LTS, Python 3.9, and PyTorch 1.10. There is no hardware acceleration available. The RPi has a memory bandwidth of 25.6 GB/s. The RPis have a class 10 32 GB SD card each and a 1GBit/s network interface. The RPis have a peak power draw of 10 Watts.

4.2 Software Stack

FLEdge is implemented on top of widely used FL libraries. We use Flower [5] to run the FL routine and PyTorch Lightning

[2]Jetson Nano with Maxwell architecture [47] only supports CUDA 10.2 and PyTorch 1.10 with Python 3.6.

[17] to allow for easy extensibility and seamless integration of new FL workloads.

4.3 FL Workloads

We use FL workloads that have been explored in previous benchmarking works.

Datasets. For NILM, we use the BLOND dataset [33]. It is an office environment appliance load monitoring dataset. It contains 13,164 samples with 12 appliance classes and captures electrical appliances in building-level office environments, such as laptops, monitors, and printers. We adopt Schwermer et al.'s [52] approach to FL with BLOND. Each sample in the dataset consists of 25,600 power readings (current, voltage). Instead of generating the input features for the DL models in the data loader, we deviate from Schwermer et al. and create the Active Power, Apparent Power, Reactive Power, and MFCC (`n_mfcc = 64`) input features offline [4]. This results in a reduced sample size of 68 x 1. The main reason for reducing the dataset size is to fit the dataset onto our RPi devices.

Table 4. Compounding effects of client dropouts and differential privacy for FedAvg across varying z levels and client dropout rates.

		BLOND								FEMNIST		Shakespeare	
		CNN		DenseNet		LSTM		ResNet		CNN		LSTM	
z	p	ϵ	Acc.	ϵ	Acc.	ϵ	Acc.	ϵ	Acc.	ϵ	Acc.	ϵ	Acc.
Loc. baseline		N/A	0.96	N/A	0.89	N/A	0.95	N/A	0.91	N/A	0.75	N/A	0.59
0	0%	∞	0.75	∞	0.74	∞	0.77	∞	0.73	∞	0.70	∞	0.53
	10%	∞	0.70	∞	0.73	∞	0.69	∞	0.73	∞	0.69	∞	0.52
	20%	∞	0.32	∞	0.69	∞	0.74	∞	0.72	∞	0.68	∞	0.51
	50%	∞	0.46	∞	0.67	∞	0.66	∞	0.70	∞	0.66	∞	0.49
0.3	0%	8.0	0.73	8.0	0.70	8.0	0.74	8.0	0.63	6.1	0.67	6.6	0.53
	10%	8.1	0.73	8.1	0.70	8.1	0.70	8.1	0.63	6.6	0.66	6.6	0.52
	20%	8.1	0.40	8.1	0.69	8.1	0.70	8.1	0.62	6.6	0.65	6.6	0.49
	50%	8.1	0.44	8.1	0.67	8.1	0.64	8.1	0.60	6.7	0.61	6.6	0.44
0.5	0%	2.3	0.54	2.3	0.64	2.3	0.74	2.3	0.61	2.1	0.62	2.0	0.52
	10%	2.4	0.54	2.4	0.63	2.4	0.74	2.4	0.61	2.1	0.61	2.0	0.49
	20%	2.4	0.38	2.4	0.60	2.4	0.72	2.4	0.60	2.1	0.61	2.0	0.45
	50%	2.4	0.36	2.4	0.57	2.4	0.70	2.4	0.60	2.1	0.59	2.0	0.44
1	0%	0.4	0.51	0.4	0.56	0.4	0.64	0.4	0.60	0.4	0.44	0.4	0.52
	10%	0.4	0.51	0.4	0.56	0.4	0.61	0.4	0.60	0.4	0.43	0.4	0.44
	20%	0.4	0.36	0.4	0.56	0.4	0.60	0.4	0.59	0.4	0.44	0.4	0.44
	50%	0.4	0.34	0.4	0.54	0.4	0.57	0.4	0.57	0.4	0.39	0.5	0.39
1.3	0%	0.2	0.48	0.2	0.56	0.2	0.69	0.2	0.42	0.3	0.34	0.3	0.51
	10%	0.2	0.45	0.2	0.56	0.2	0.64	0.2	0.42	0.3	0.34	0.3	0.44
	20%	0.2	0.38	0.2	0.56	0.2	0.63	0.2	0.42	0.3	0.32	0.3	0.44
	50%	0.2	0.34	0.2	0.51	0.2	0.61	0.2	0.39	0.3	0.28	0.3	0.39
1.5	0%	0.2	0.34	0.2	0.30	0.2	0.37	0.2	0.16	0.3	0.28	0.3	0.48
	10%	0.2	0.34	0.2	0.30	0.2	0.37	0.2	0.15	0.3	0.28	0.3	0.44
	20%	0.2	0.37	0.2	0.29	0.2	0.35	0.2	0.15	0.3	0.20	0.3	0.43
	50%	0.2	0.19	0.2	0.28	0.2	0.31	0.2	0.15	0.3	0.17	0.3	0.36

We employ the FEMNIST dataset [6] for CV. FEMNIST consists of 814,000 samples of hand-written digits and letters. The dataset contains 62 classes of handwritten digits (26 lower case letters, 26 upper case letters, and 10 digits). We randomly crop the grayscale samples to 28x28 and perform a random flip before training. Character recognition is frequently used by mobile clients to convert images to editable text documents.

We use the Shakespeare dataset [23] for NLP. It consists of the complete works of William Shakespeare. It is preprocessed in the exact same way as introduced in the LEAF benchmark [6]. The dataset was preprocessed with a sliding window of 80 characters and a stride of 1 to prepare it for the next-character prediction task. The vocabulary was generated over the alphabet, including special characters, resulting in a total size of 80. Overall, the nature of the Shakespeare dataset resembles a task like a next-word prediction on smartphone keyboards.

We complement experiments in the NLP domain with the SAMSum dataset, which contains 16,000 pairs of chat-message-like dialogue and summary pairs that may be used for sequence-to-sequence modeling tasks [21]. With SAMSum, we introduce a realistic use case that can be used on mobile clients to provide a quick overview of their chat history, as it is frequently found in applications like Slack. Apart from splitting the dataset into 10 Dirichlet subsets ($\alpha = 1$), we do not apply additional preprocessing. Researching FL applications on physical devices usually requires fitting the data distribution to the number of devices on our testbed. We opt to sample client subsets to 45 based on a Dirichlet distribution ($\alpha = 1$) to fit the number of same-type clients in our testbed (Figure 2).

Models. We train a total of 7 DL models. We use four different architectures to train on the BLOND datasets to showcase the sensitivity of different model sizes and architectures to real-world environmental conditions: a CNN, an LSTM, a ResNet, and a DenseNet architecture [52]. To train on the FEMNIST dataset, we train a small and efficient CNN architecture, originally presented in the LEAF benchmark [6]. For the Shakespeare dataset, we use an LSTM model with 256 hidden dimensions and zero dropout that has been well explored to solve the next character prediction task [23]. Additionally, we use the SAMSum dataset to evaluate the hardware performance of the NVIDIA Jetson AGX Orin devices. To do so, we employ FLAN-T5-Small, an 80M parameter state-of-the-art transformer model [12]. When fine-tuned for a specific downstream task, the FLAN-T5 model family has proven to deliver on-par performance with significantly larger foundation models such as LLama2 [20].

For the models used on BLOND, FEMNIST, and Shakespeare, we train one local epoch on clients and then send updates for aggregation to the server to eliminate potential risks of client drift [41]. The exact model configuration and hyperparameters are available from Table 3.

FL strategies. We explore all models in conjunction with six state-of-the-art FL strategies. We include FedAvg, the first communication efficient FL strategy, aggregating client updates over an unweighted average [41]. We also include adaptive strategies introduced by Reddi et al. [49], namely Fed-Adam, FedYogi, and Fed-AdaGrad. The adaptive strategies introduce a server-side learning rate to better account for data heterogeneity. We further adopt two strategies that aim to increase fairness and robustness in an FL system. FedProx [37] introduces a method for weighting client updates in the aggregation process based on the amount of data a client has processed for an update. q-fair FedAvg (qFedAvg) [38] is a derivative of FedAvg introducing a factor q that determines the level of fairness. Fairness in this context is defined by how well a model generalizes across clients. Higher generalizability is achieved by overweighting those clients that have the highest loss. This is done to gear a model stronger towards the high-loss clients and reduce the accuracy variance across clients. We fix the number of FL rounds to 10 for all FL strategies. Further details on the datasets and DL models are available in Table 2.

Table 5. Global model validation accuracy across FL strategies with varying client dropout rates after 100 FL training rounds with a client selection rate of 20% per training round. **Bold** highlights the best-performing FL strategy.

		BLOND				FEMNIST	Shakespeare
p	Strategy	CNN	DenseNet	LSTM	ResNet	CNN	LSTM
Loc. Baseline (p = 0%)		0.96±0.02	0.89±0.01	0.95±0.01	0.91±0.01	0.75±0.02	0.59±0.01
0%	FedAdaGrad	0.76±0.0	0.17±0.07	0.71±0.03	0.70±0.0	0.56±0.01	0.52±0.01
	FedAdam	0.73±0.01	0.03±0.0	0.64±0.05	0.64±0.02	0.38±0.05	0.5±0.03
	FedAvg	0.75±0.02	0.74±0.0	0.77±0.01	0.73±0.01	**0.7±0.0**	**0.53±0.0**
	FedProx	0.75±0.01	0.74±0.0	0.74±0.01	0.73±0.01	**0.7±0.0**	0.52±0.0
	FedYogi	**0.8±0.01**	**0.79±0.0**	**0.81±0.01**	**0.78±0.0**	0.53±0.0	0.25±0.0
	qFedAvg	0.73±0.0	0.7±0.0	0.79±0.0	0.71±0.0	0.03±0.0	0.47±0.0
10%	FedAdaGrad	0.76±0.03	0.03±0.0	0.69±0.02	0.68±0.02	0.55±0.04	0.51±0.02
	FedAdam	0.73±0.01	0.03±0.0	0.54±0.1	0.41±0.22	0.35±0.01	0.48±0.01
	FedAvg	0.7±0.04	0.73±0.01	0.69±0.04	0.72±0.0	**0.69±0.01**	**0.52±0.01**
	FedProx	0.74±0.05	0.73±0.01	0.68±0.03	0.73±0.01	0.68±0.01	**0.52±0.01**
	FedYogi	**0.76±0.02**	**0.78±0.01**	**0.8±0.0**	**0.76±0.03**	0.53±0.01	0.23±0.01
	qFedAvg	0.73±0.0	0.69±0.0	0.77±0.0	0.71±0.0	0.03±0.0	0.47±0.0
20%	FedAdaGrad	0.64±0.1	0.04±0.0	0.48±0.3	0.69±0.03	0.49±0.06	0.49±0.02
	FedAdam	0.73±0.0	0.03±0.0	0.53±0.13	0.63±0.0	0.34±0.05	0.35±0.06
	FedAvg	0.32±0.27	0.69±0.05	0.74±0.03	0.72±0.01	**0.68±0.02**	**0.51±0.02**
	FedProx	0.37±0.3	0.69±0.05	0.68±0.04	0.72±0.01	0.67±0.01	**0.51±0.02**
	FedYogi	**0.76±0.02**	**0.76±0.03**	**0.76±0.0**	**0.75±0.03**	0.47±0.08	0.21±0.03
	qFedAvg	0.72±0.01	0.68±0.0	0.75±0.0	0.71±0.0	0.03±0.0	0.47±0.0
50%	FedAdaGrad	0.42±0.24	0.03±0.0	0.35±0.27	0.68±0.04	0.45±0.06	0.45±0.03
	FedAdam	0.25±0.25	0.03±0.0	0.18±0.13	0.62±0.01	0.2±0.09	0.35±0.02
	FedAvg	0.46±0.16	0.67±0.06	0.66±0.01	0.7±0.02	**0.66±0.03**	**0.49±0.03**
	FedProx	0.52±0.16	0.67±0.06	0.68±0.04	0.7±0.02	0.64±0.02	**0.49±0.03**
	FedYogi	**0.72±0.01**	**0.71±0.09**	**0.74±0.04**	**0.74±0.04**	0.43±0.12	0.2±0.04
	qFedAvg	0.68±0.01	0.67±0.0	0.72±0.01	0.71±0.0	0.03±0.0	0.47±0.0

Network. We modify our testbed communication to realistically reflect real-world scenarios with ERRANT [56]. We use a 1 GBit/s synchronous network link as well as the global average 4G LTE connection characteristics of 40 MBit/s download and 15 MBit/s upload [56].

5 Results

To show the practical utility of FLEdge, we run extensive experiments evaluating FL workloads on client behavior, differential privacy, energy efficiency, and hardware diversity.

5.1 Client behavior

Existing FL strategies work well with unreliable clients. Unreliability is a core challenge for edge computing systems. Especially for FL workloads, this bears the potential for significant information loss whenever model updates are not transmitted to the server, regardless of the root cause. As embedded devices do not have power redundancy, run in sub-optimal environments w.r.t. heat dispersion, and may suffer from outside damages, they are considered unreliable by nature. Our experiments show that existing state-of-the-art FL strategies work well in systems with unreliable clients, i.e., high dropout rates. However, for each dataset, we see that one FL strategy always consistently performs best. This suggests that the FL strategy choice overall depends on the dataset, not the model. Yet, we also identify FedAdaGrad and FedAdam as particularly sensitive to client dropout. Thus, careful strategy selection and federated hyperparameter optimization are critical for systems that involve unreliable clients.

5.2 Differential Privacy

Heterogeneous client behavior has significant negative effects on the model quality of differentially private workloads. Since DP is an integral component of FL workloads, the question arises: how does it change the model quality with varying client reliability levels? Interestingly, the model quality decreases significantly when introducing client dropout in a system, and the negative effects across

Table 6. Computation and communication cost overview for our seven FL workloads across varying network conditions and 100 FL communication rounds (20% selection rate, 9 training clients per round). Wireless communication drives communication costs by one order of magnitude.

			Communication				Computation						G					
			4G LTE		1 GBit/s fiber		RPi 4		Nano		Orin		4G LTE			1 Gbit/s fiber		
Dataset	Model	Communicable parameters	Time	Power (kWh)	Time	Power (kWh)	Time	Power (kWh)	Time	Power (kWh)	Time	Power (kWh)	RPi 4	Nano	Orin	RPi 4	Nano	Orin
FEMNIST	CNN	0.1 MB	20s	0.0006	0.8s	0.0001	56s	0.0451	31s	0.0352	11s	0.0059	791	438	579	35000	19375	25625
BLOND	CNN	0.1 MB	20s	0.0003	0.8s	0.0001	65s	0.0530	23s	0.0269	11s	0.0059	918	325	155	40625	14375	6875
BLOND	LSTM	0.2 MB	20s	0.0007	0.8s	0.0001	80s	0.0652	21s	0.0243	13s	0.0074	565	148	92	25000	6563	4063
BLOND	ResNet	0.4 MB	30s	0.0016	1.2s	0.0002	588s	0.5197	36s	0.0417	15s	0.0086	2076	127	53	91875	5625	2344
BLOND	DenseNet	1.0 MB	70s	0.0041	2.8s	0.0006	586s	0.5125	36s	0.0416	16s	0.0086	828	51	23	36625	2250	1000
Shakespeare	LSTM	3.2 MB	240s	0.0134	9.6s	0.0021	981s	0.9172	59s	0.0724	19s	0.0103	433	26	8	19160	1152	371
SAMSum	FLAN-T5 Small	308 MB	21800s	1.3124	872s	0.2008	OOM	OOM	OOM	OOM	324s	2.1492	OOM	OOM	1.5	OOM	OOM	65.7

strategies become significant at a low DP level. Since we use a server-side user-level DP algorithm, we adjust the noise level based on the model updates received per training round. This provides appropriate privacy guarantees. Yet, the level of noise we have to add to cover for failing clients reduces the model quality significantly (Table 4). For instance, if we want to provide a loose privacy budget of $\epsilon = 8$ on the BLOND dataset and suffer from a high client dropout rate ($p = 0.5$), the accuracy almost halves compared to a system without client dropouts ($p = 0$). As such, allowing re-training with another client in a training round or accounting for late arrivals could mitigate the effects DP has on the model quality. Also, the evaluation of DP in unreliable systems is a strong indicator of whether deploying an FL workload into an edge computing system w.r.t. to the estimated model quality in relation to the privacy requirements is worthwhile. However, training quality and efficiency are equally relevant for an effective deployment.

5.3 Communication Efficiency

Granularity helps quantify the practical utility in relation to the network when deploying FL workloads in edge computing systems. Before discussing the hardware characteristics of diverse embedded platforms, the question is whether it is even worthwhile deploying an FL workload to them under given network conditions.

With larger models, FL workloads become more communication intensive as we have to send more model weights. From a practical perspective, quantifying whether it's worthwhile to consider including embedded devices in a system that runs an FL workload is essential. For all of our state-of-the-art FL workloads, G is significantly above 1 (Table 6). This indicates the high suitability of such small models to be trained as an FL workload, regardless of the communication technology. This is particularly useful for highly specialized tasks like object detection. Yet, for more generalizing tasks like in the NLP space, we require larger models to run on embedded devices and be trained at the edge. For larger models like FLAN-T5 Small, we see $G = 1.5$. With G close to 1, the practical utility is limited as communication takes approximately as much time as computation. Beyond the computation/communication trade-off, G is also suitable for quantifying the net effect of communication optimization methods for an FL workload. Further, reliability is a major concern in edge computing systems with embedded clients. We want to spend as little time communicating as possible to get the model updates to the server and not risk failures during long communication times. However, computational performance and efficiency are key challenges as well.

5.4 Energy Efficiency

Energy efficiency is a well-suited indicator for computational bottlenecks on embedded devices and does not require extensive client monitoring that could potentially infringe privacy. As we will see in Section 5.5, the measurements of local step times on clients unveil inefficiencies and scalability limits. Yet, micro-benchmarks are often difficult to facilitate when having a variety of devices in a system. While micro-benchmarks are useful for an in-depth exploration of bottleneck root causes on a client, they require significant efforts and interfere with the DL training process. So, how can we identify bottlenecks without interfering with the client's training process?

We find energy efficiency to be a well-suited estimator for the suitability of a device for a given task and also for the identification of bottlenecks (Figure 3). The main benefit of energy efficiency is that it measures the throughput per Watt of power consumption, two available metrics without interference with the training process, i.e., they are convenient to measure. The device comparison shows the significant advancements in hardware for FL workloads on the edge (Figure 3a). It is also an indicator that we should look more at models with multi-million parameter models for FL workloads. The Orins have proven to become more energy efficient with increasing parameter size and outperform all other embedded device platforms as well as data center resources for state-of-the-art FL workloads. We identify the computational capabilities of the Orins by running the 80M parameter FLAN-T5 model for the SAMSum text summarization task (Figure 3b). By scaling the batch size, we

quickly discover that the energy efficiency stagnates at a batch size of 8 and does not increase any further. The reason is the significant memory bottleneck we uncover with the micro-benchmark (see Section 5.5). Yet, this bottleneck is immediately visible without interfering with the training process itself. When monitoring embedded devices in practice, it is a necessity to have easy-to-interpret metrics for quick responses to inefficiencies. As such, energy efficiency covers two integral aspects of deploying FL workloads in edge computing systems. First, we can quantify the suitability of an embedded device for the deployment of an FL workload. Second, computational limitations become instantaneously visible for a remote operator of an FL system. This is particularly important since an operator cannot access the devices.

5.5 Hardware Heterogeneity

Contrary to current trends in high-performance DL, scaling the batch size on embedded devices does not lead to greater computational efficiencies. For our benchmark, embedded hardware forms the very center of attention since it ultimately depends on the devices deployed in the field how well an FL workload will run (Figure 4a). Therefore, our micro-benchmark provides a comprehensive baseline for the suitability of device/workload combinations. In systems with legacy hardware, as it is often found in industrial systems with long product lifecycles, the benchmark reveals their limited utility for FL workloads. This is especially evident for workloads that entail models with several million parameters since scaling the minibatch size beyond 8 samples does not yield performance benefits. As we double the minibatch size, the runtime also doubles.

This originates from the limited memory bandwidth of embedded devices and first affects the optimizer step function, which requires a significant amount of interactions between processor and memory. Interestingly, for the models larger than 100K parameters, the forward step also grows exponentially. This operation does not require many interactions between computing and memory, only batch loading once per batch to send the data through the DL model. One could think this might be an I/O bottleneck originating from the disk. Yet, we run the same SD cards on the Nanos and RPis. The Nanos do not show this behavior in any use case.

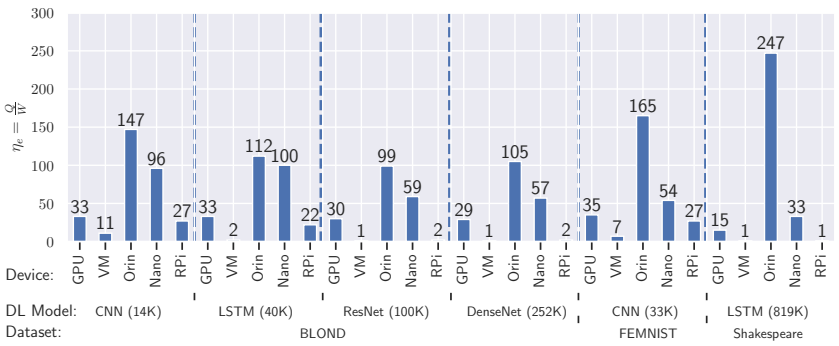
The Orin platform, as the most recent embedded device type, yields strong results throughout our experiments. Yet, they are made for significantly larger workloads involving DL models with several million parameters. Therefore, we study their behavior when fine-tuning the FLAN-T5 Small model with different batch sizes (Figure 4b). We see linearly growing step times, which is a strong sign of a bottleneck, as one would normally expect to see a logarithmically increasing step time with scaling batch size. The bottleneck originates from the limited memory bandwidth of the Orins (204 GB/s) vs. modern data center DL accelerators (e.g., the A6000 with 768 GB/s). Therefore, federated hyperparameter optimization is not only a matter of model quality but must also consider the hardware limitations.

Our profiling experiments with the FLAN-T5 Small model on the Orins and data center GPUs show two main bottlenecks. The profiling was done for an identical workload used across the different hardware with a minibatch size of 128 samples.

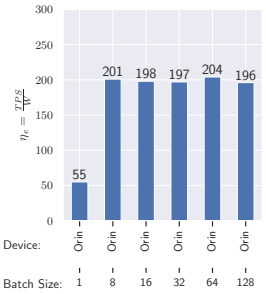
We start with CPU-related bottlenecks. The performance difference between the ARM-based Orins and the x86-based CPU in our GPU server becomes evident in the `opt.step()`, executed on each device's CPU. Overall, the `opt.step()` takes 4× longer on the Orins, which is primarily rooted in an `aten::_foreach_add` operation that performs an element-wise addition of tensors. Within this operation, there are noteworthy differences in how much time is spent on the actual computation and how much is used for data movement from and to the main memory. The Orins spend 83% of the operation time on data movement, while the GPU server only needs 77%. This originates from a different handling of the `aten::copy_` operation (to move data from GPU to CPU memory) on the two devices. Since the Orins have shared memory for the CPU and GPU, data is assigned to either device via a context switch, while on the GPU server, data is moved between the main memory and GPU memory. One would expect a context switch to be faster than copying data from one memory to another. However, we observe that the context switch takes between $5 - 8\mu s$, regardless of the data size, while the data movement on the GPU server depends on the amount of data transferred, taking a maximum of $1\mu s$. Overall, this requires the Orins to spend more time on data movement.

The second major bottleneck is located on the GPU during the forward step. At a high level, we find that the GPU on the Orins is permanently busy with operations, while the GPU server has notable idle times. We identify matrix multiplications on the GPU as the main driver for the longer processing times on the Orins. This is a sign of a computational bottleneck. Specifically, `ampere_sgemm_128x128_nn`, a compute-intensive operation, stands out since it takes up to 32× longer on the Orins than on the A6000 (11ms vs. $323\mu s$). This is rooted in the significantly higher number of tasks (warps) per GPU streaming multiprocessor (SM) on the Orins. We see 1200 warps per SM on the Orins, while there are only 37 on the A6000. As such, the Orins have to process 32× more tasks per SM than the A6000. At the same time, we also observe longer memory operation times. Immediately preceding the matrix multiplication, the `direct_copy_kernel_cuda` data movement operator from CPU to GPU memory takes up to 5.9× longer on the Orins vs. the A6000.

Overall, we observe a major limitation of the Orins when it comes to computational intensity and data movement between CPU and GPU, even though the device has a shared memory architecture.



(a) Device comparison



(b) Scaling batch size with the FLAN-T5 Small (80M) transformer

Figure 3. Energy efficiency measured across datasets and DL models (# parameters).[5] Results are measured over one epoch of training. The Orins train with a minibatch size of 256 across all models. Higher values are better.

As such, the micro-benchmark design paired with detailed profiling results aids not only in identifying the right device, workload, and hyperparameter combinations but also reveals hardware bottlenecks on the kernel level. Further, it helps detect defective parts and can be used as a predictive monitoring element to improve system stability, even though the clients may not be directly accessible.

6 Related Work

Edge computing. For Edge Computing systems, there are numerous benchmarking tools. Edge devices have long been regarded as a data pass-through gateway rather than data processing devices and, therefore, have been evaluated in the context of data transfer capacities and reliability [34]. In 2019, McChesney et al. [40] introduced DeFog, a benchmark including DL inference workloads on embedded devices with YOLO models. Yet, these benchmarks do not include system dependencies between clients and a server as we would find in FL systems, where training progress relies on client participation. Varghese et al. [58] provide a comprehensive overview of a wide range of edge computing benchmarks, but none is targeted to FL.

Table 7. Comparison of FLEdge with other FL benchmarks.

Benchmark	Year	Primary Eval. Purpose	ML Domains CV	NLP	NILM	Analysis Dimensions (1)	(2)	(3)	(4)
LEAF	2018	Aggregation	✓	✓					
FedML	2020	Aggregation	✓	✓					
Flower	2020	Aggregation	✓	✓		✓			
FederatedScope	2022	Personalization	✓	✓		✓			
FedScale	2022	Scalability	✓	✓		✓			
FLEdge	2024	FL Clients	✓	✓	✓	✓	✓	✓	✓

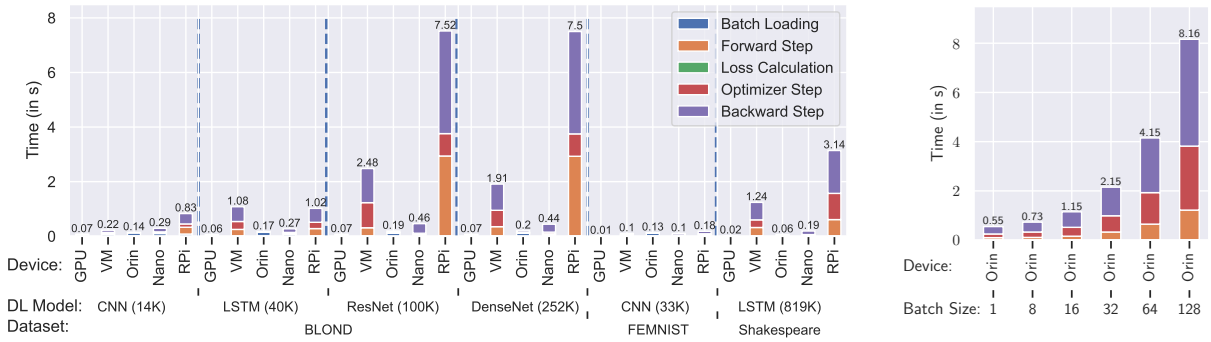
Analysis Dims.: (1) Data Security, (2) Dedicated Edge Deployment, (3) Client Behavior, (4) Client Capabilities

FL benchmarks. LEAF [6] and FedML [23] introduce two benchmarks that focus on the evaluation of FL optimization algorithms with data heterogeneity. Flower [5] and FederatedScope [69] extend FL benchmark capabilities with private computing by using (ϵ, δ)-DP as well as cryptographic aggregation. FederatedScope also enables benchmarking of personalized FL. FedScale [35] evaluates the scalability characteristics of FL systems. With FLEdge, we complement the benchmarking landscape for FL applications with client capability and behavior analysis tools that integrate with previous benchmarking works. Further, we are the first to design a benchmark that is entirely based on dedicated hardware. Our study emulates wide-area networking and client behavior to enable systematic analyses (Table 7).

Real-world FL workloads. It is particularly challenging to evaluate FL workloads in real-world edge computing systems as devices are often not accessible for research. FedScale [35] shed light on the high-level performance characteristics of two mobile device platforms for FL workloads. The same is true for FS-Real [10], which is based on FederatedScope. It presents a runtime optimized for mobile devices running Android. FLINT by Wang et al. [60] presents an approach to integrating FL workloads into existing ML landscapes and discusses the effects of scaling DL models beyond cloud resources onto mobile devices on the network edge. While Wang et al. discuss the utility of hardware analytics for task scheduling in FL systems, they outline the open necessity for in-depth hardware and energy analysis due to the wide variety of devices being available for FL workload deployment. As such, there is still a lack of real-time analysis of FL workloads on low-power devices and missing guidance on in-depth performance differences between platforms, especially since the landscape of DL-accelerated embedded devices is growing rapidly.

With FLEdge, we expand on hardware-centric research by conducting in-depth evaluations of compute performance

[5] We use estimates from SelfWatts [19] for approximating the power consumption of VMs.



(a) Device comparison

(b) Scaling batch size with the FLAN-T5 Small (80M) transformer

Figure 4. Training times for different device types and model sizes over one minibatch. We further scale the minibatch size for the FLAN-T5 transformer model to showcase edge-specific bottlenecks. Next to the model name, we report the model parameters. Lower is better.

across various embedded device generations and the role of energy in FL workloads. To address the distinctive characteristics of client behavior in edge computing systems, we extensively study the effects on DP FL workloads.

7 Lessons Learned

We summarize what we learned from studying FL from a hardware centric perspective. Findings relate to client behavior that we must account for in the future, improving FL applications based on the hardware specifications we find on the network edge and what workloads are beneficial to be deployed to embedded hardware.

Client participation patterns in FL are determinant for reliable training that entails (ϵ, δ)-DP. We find that client participation significantly negatively affects the training performance of FL workloads (Section 5.1). While we provide an approach to adjusting DP noise levels based on the number of clients that submit model updates, client dropouts lead to higher noise levels to provide the same privacy guarantees. This degrades the training performance and requires more time and resources to achieve the same performance as in FL systems that do not employ DP. Yet, to better understand how (ϵ, δ)-DP can be applied in practice, it is imperative to further study what ϵ levels can be considered acceptable. At the same time, it is a priority to understand realistic client participation statistics and build middleware that can account for unreliable clients [62].

Larger ML models benefit the scalability. With larger models, we spend more time on computation and less on communication (Section 5.3). Thus, the overall FL process becomes more efficient since more time is spent learning from client data. Thus, workload and hardware fit can help drive overall system efficiency, carefully balancing training speed and power draw.

Context switches for data assignment between CPU and GPU is a major limiting factor on embedded hardware. Despite the unified memory architecture on state-of-the-art embedded hardware, context switches between CPU and GPU memory are time-consuming compared to the physical data movement operation on data center GPUs (Section 5.5). This can be a major cost driver for FL applications in edge computing systems and should be considered when deciding on data movement frequency.

8 Conclusions

In this paper, we present FLEdge, a hardware-centric benchmark for deploying FL workloads on the edge. We study hardware diversity, efficiency, and robustness. Our experimental results show the significance of hardware diversity for the training performance of FL workloads, along with the important role of network conditions. We show the limits of current state-of-the-art embedded devices and the effects of DL hyperparameters on training efficiency. The minibatch size, which is a key factor in scaling DL workloads in data centers, shows very limited effects for FL workloads on the edge. Our experiments on energy efficiency introduce an early detector for computational bottlenecks that does not require interference with the training process. As such, it is suitable for further evaluating the suitability of energy efficiency as an optimization lever for FL workloads. While our experiments on heterogeneous client behavior show the robustness of state-of-the-art FL strategies, we identify a particular sensitivity of DP FL workloads to client failures, which is a critical factor when deploying FL workloads that involve sensitive data. Overall, FLEdge, our hardware-centric benchmark, contributes to the practical considerations required to deploy FL workloads to edge computing systems, and we hope to further spur research toward FL middleware that can control both on-client and global learning efficiency.

Acknowledgements

This work is partially funded by the German Federal Ministry of Economic Affairs and Climate Action (Grant: 16KN085729) and the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01). We would like to thank the PANDORA project (https://pandora-heu.eu/) for our fruitful discussions.

References

[1] 3GPP. 2008. 4G (LTE Advanced). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2585.

[2] Galen Andrew, Om Thakkar, H. Brendan McMahan, and Swaroop Ramaswamy. 2019. Differentially Private Learning with Adaptive Clipping. https://doi.org/10.48550/ARXIV.1905.03871

[3] Danilo Ardagna, Giuliano Casale, Michele Ciavotta, Juan F Pérez, and Weikun Wang. 2014. Quality-of-service in cloud computing: modeling techniques and their applications. *Journal of Internet Services and Applications* 5, 1 (Sept. 2014). https://doi.org/10.1186/s13174-014-0011-3

[4] Karim Said Barsim, Roman Streubel, and Bin Yang. 2014. Unsupervised adaptive event detection for building-level energy disaggregation. *Proceedings of Power and Energy Student Summit (PESS), Stuttgart, Germany* (2014).

[5] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. 2020. Flower: A Friendly Federated Learning Research Framework. https://doi.org/10.48550/ARXIV.2007.14390

[6] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. LEAF: A Benchmark for Federated Settings. https://doi.org/10.48550/ARXIV.1812.01097

[7] Maurizio Capra, Riccardo Peloso, Guido Masera, Massimo Ruo Roch, and Maurizio Martina. 2019. Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World. *Future Internet* 11, 4 (April 2019), 100. https://doi.org/10.3390/fi11040100

[8] Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. 2021. On Large-Cohort Training for Federated Learning. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). Virtual Event. https://openreview.net/forum?id=Kb26p7chwhf

[9] Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. 2022. pFL-Bench: A Comprehensive Benchmark for Personalized Federated Learning. https://doi.org/10.48550/ARXIV.2206.03655

[10] Daoyuan Chen, Dawei Gao, Yuexiang Xie, Xuchen Pan, Zitao Li, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. FS-Real: Towards Real-World Cross-Device Federated Learning. https://doi.org/10.48550/ARXIV.2303.13363

[11] Jiasi Chen and Xukan Ran. 2019. Deep Learning With Edge Computing: A Review. *Proc. IEEE* 107, 8 (Aug. 2019), 1655–1674. https://doi.org/10.1109/jproc.2019.2921977

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. arXiv:2210.11416 [cs.LG]

[13] Ryan A. Cooke and Suhaib A. Fahmy. 2020. A model for distributed in-network and near-edge computing with heterogeneous hardware. *Future Generation Computer Systems* 105 (April 2020), 395–409. https://doi.org/10.1016/j.future.2019.11.040

[14] Dell. 2023. Edge Networking | Dell USA. https://www.dell.com/en-us/shop/servers-storage-networking/sf/access-platforms

[15] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. 2023. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems* 38 (April 2023), 100857. https://doi.org/10.1016/j.suscom.2023.100857

[16] Cynthia Dwork and Aaron Roth. 2013. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3-4 (2013), 211–407. https://doi.org/10.1561/0400000042

[17] William Falcon et al. 2019. Pytorch lightning. *GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning* 3, 6 (2019).

[18] Xiuwen Fang and Mang Ye. 2022. Robust Federated Learning With Noisy and Heterogeneous Clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 10072–10081.

[19] Guillaume Fieni, Romain Rouvoy, and Lionel Seiturier. 2021. SelfWatts: On-the-fly Selection of Performance Events to Optimize Software-defined Power Meters. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, Virtual Event. https://doi.org/10.1109/ccgrid51090.2021.00042

[20] Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Bhushan TN. 2024. Tiny Titans: Can Smaller Large Language Models Punch Above Their Weight in the Real World for Meeting Summarization? https://doi.org/10.48550/ARXIV.2402.00841

[21] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, Hong Kong, China, 70–79. https://doi.org/10.18653/v1/D19-5409

[22] Mohammad Goudarzi, Marimuthu Palaniswami, and Rajkumar Buyya. 2022. Scheduling IoT Applications in Edge and Fog Computing Environments: A Taxonomy and Future Directions. *Comput. Surveys* 55, 7 (Dec. 2022), 1–41. https://doi.org/10.1145/3544836

[23] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. https://doi.org/10.48550/ARXIV.2007.13518

[24] Xiaofan He, Juan Liu, Richeng Jin, and Huaiyu Dai. 2017. Privacy-Aware Offloading in Mobile-Edge Computing. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. IEEE. https://doi.org/10.1109/glocom.2017.8253985

[25] Hua-Jun Hong. 2017. From Cloud Computing to Fog Computing: Unleash the Power of Edge and End Devices. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE. https://doi.org/10.1109/cloudcom.2017.53

[26] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://doi.org/10.48550/ARXIV.1704.04861

[27] Kai Hwang. 2010. *Advanced computer architecture*.

[28] Fatemeh Jalali, Rob Ayre, Arun Vishwanath, Kerry Hinton, Tansu Alpcan, and Rod Tucker. 2014. Energy Consumption of Content Distribution from Nano Data Centers versus Centralized Data Centers. *ACM SIGMETRICS Performance Evaluation Review* 42, 3 (Dec. 2014), 49–54. https://doi.org/10.1145/2695533.2695555

[29] Yaser Jararweh, Ahmad Doulat, Omar AlQudah, Ejaz Ahmed, Mahmoud Al-Ayyoub, and Elhadj Benkhelifa. 2016. The future of mobile cloud computing: Integrating cloudlets and Mobile Edge Computing.

In *2016 23rd International Conference on Telecommunications (ICT)*. IEEE. https://doi.org/10.1109/ict.2016.7500486

[30] Congfeng Jiang, Tiantian Fan, Honghao Gao, Weisong Shi, Liangkai Liu, Christophe Cérin, and Jian Wan. 2020. Energy aware edge computing: A survey. *Computer Communications* 151 (Feb. 2020), 556–580. https://doi.org/10.1016/j.comcom.2020.01.004

[31] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 5132–5143. https://proceedings.mlr.press/v119/karimireddy20a.html

[32] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. https://doi.org/10.48550/ARXIV.1610.05492

[33] Thomas Kriechbaumer, Anwar Ul Haq, Matthias Kahl, and Hans-Arno Jacobsen. 2017. MEDAL. In *Proceedings of the Eighth International Conference on Future Energy Systems*. ACM. https://doi.org/10.1145/3077839.3077844

[34] C. P. Kruger and G. P. Hancke. 2014. Benchmarking Internet of things devices. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE. https://doi.org/10.1109/indin.2014.6945583

[35] Fan Lai, Yinwei Dai, Sanjay S. Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. FedScale: Benchmarking Model and System Performance of Federated Learning at Scale. (2021). https://doi.org/10.48550/ARXIV.2105.11367

[36] Chao Li, Yushu Xue, Jing Wang, Weigong Zhang, and Tao Li. 2018. Edge-Oriented Computing Paradigms. *Comput. Surveys* 51, 2 (April 2018), 1–34. https://doi.org/10.1145/3154815

[37] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated Optimization in Heterogeneous Networks. https://doi.org/10.48550/ARXIV.1812.06127

[38] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2019. Fair Resource Allocation in Federated Learning. https://doi.org/10.48550/ARXIV.1905.10497

[39] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. 2017. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2322–2358. https://doi.org/10.1109/comst.2017.2745201

[40] Jonathan McChesney, Nan Wang, Ashish Tanwer, Eyal de Lara, and Blesson Varghese. 2019. DeFog. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. ACM. https://doi.org/10.1145/3318216.3363299

[41] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv* (2016). https://doi.org/10.48550/ARXIV.1602.05629

[42] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning Differentially Private Recurrent Language Models. https://doi.org/10.48550/ARXIV.1710.06963

[43] Aritra Mitra, Rayana Jaafar, George J. Pappas, and Hamed Hassani. 2021. Linear Convergence in Federated Learning: Tackling Client Heterogeneity and Sparse Gradients. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 14606–14619. https://proceedings.neurips.cc/paper_files/paper/2021/file/7a6bda9ad6ffdac035c752743b7e9d0e-Paper.pdf

[44] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic Federated Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 4615–4625. https://proceedings.mlr.press/v97/mohri19a.html

[45] Solmaz Niknam, Harpreet S. Dhillon, and Jeffery H. Reed. 2019. Federated Learning for Wireless Communications: Motivation, Opportunities and Challenges. https://doi.org/10.48550/ARXIV.1908.06847

[46] NVIDIA. 2020. Jetson Roadmap. https://developer.nvidia.com/embedded/develop/roadmap

[47] NVIDIA. 2022. https://developer.nvidia.com/blog/maxwell-most-advanced-cuda-gpu-ever-made/

[48] NVIDIA. 2022. NVIDIA Jetson AGX Orin Series. https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf

[49] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2020. Adaptive Federated Optimization. https://doi.org/10.48550/ARXIV.2003.00295

[50] Jinke Ren, Guanding Yu, Yinghui He, and Geoffrey Ye Li. 2019. Collaborative Cloud and Edge Computing for Latency Minimization. *IEEE Transactions on Vehicular Technology* 68, 5 (May 2019), 5031–5044. https://doi.org/10.1109/tvt.2019.2904244

[51] Felix Sattler, Simon Wiedemann, Klaus-Robert Muller, and Wojciech Samek. 2020. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems* 31, 9 (Sept. 2020), 3400–3413. https://doi.org/10.1109/tnnls.2019.2944481

[52] René Schwermer, Jonas Buchberger, Ruben Mayer, and Hans-Arno Jacobsen. 2022. Federated office plug-load identification for building management systems. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems* (Virtual Event). ACM, New York, NY, USA.

[53] Siemens. 2023. Industrial Edge Devices. https://www.siemens.com/global/en/products/automation/topic-areas/industrial-edge/industrial-edge-devices.html

[54] Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. 2021. Federated Reconstruction: Partially Local Federated Learning. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 11220–11232. https://proceedings.neurips.cc/paper_files/paper/2021/file/5d44a2b0d85aa1a4dd3f218be6422c66-Paper.pdf

[55] D. Steinkraus, I. Buck, and P.Y. Simard. 2005. Using GPUs for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE. https://doi.org/10.1109/icdar.2005.251

[56] Martino Trevisan, Ali Safari Khatouni, and Danilo Giordano. 2020. ERRANT: Realistic emulation of radio access networks. *Computer Networks* 176 (2020), 107289. https://doi.org/10.1016/j.comnet.2020.107289

[57] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S. Nikolopoulos. 2016. Challenges and Opportunities in Edge Computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE. https://doi.org/10.1109/smartcloud.2016.18

[58] Blesson Varghese, Nan Wang, David Bermbach, Cheol-Ho Hong, Eyal De Lara, Weisong Shi, and Christopher Stewart. 2022. A Survey on Edge Performance Benchmarking. *Comput. Surveys* 54, 3 (April 2022), 1–33. https://doi.org/10.1145/3444692

[59] Arun Vishwanath, Fatemeh Jalali, Kerry Hinton, Tansu Alpcan, Robert W. A. Ayre, and Rodney S. Tucker. 2015. Energy Consumption Comparison of Interactive Cloud-Based and Local Applications. *IEEE Journal on Selected Areas in Communications* 33, 4 (April 2015), 616–626. https://doi.org/10.1109/jsac.2015.2393431

[60] Ewen Wang, Boyi Chen, Mosharaf Chowdhury, Ajay Kannan, and Franco Liang. 2023. FLINT: A Platform for Federated Learning Integration. *Proceedings of Machine Learning and Systems* 5 (2023).

[61] Hui-Po Wang, Sebastian Stich, Yang He, and Mario Fritz. 2022. ProgFed: Effective, Communication, and Computation Efficient Federated Learning by Progressive Training. In *Proceedings of the 39th International*

Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162), Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, Baltimore, MD, 23034–23054. https://proceedings.mlr.press/v162/wang22y.html

[62] Shiqiang Wang and Mingyue Ji. 2024. A Lightweight Method for Tackling Unknown Participation Statistics in Federated Averaging. In *The Twelfth International Conference on Learning Representations*. ICLR, Vienna, Austria. https://openreview.net/forum?id=ZKEuFKfCKA

[63] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. 2019. In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning. *IEEE Network* 33, 5 (Sept. 2019), 156–165. https://doi.org/10.1109/mnet.2019.1800286

[64] Yujia Wang, Lu Lin, and Jinghui Chen. 2022. Communication-Efficient Adaptive Federated Learning. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, Baltimore, MD, 22802–22838. https://proceedings.mlr.press/v162/wang22o.html

[65] Yingchun Wang, Jingyi Wang, Weizhan Zhang, Yufeng Zhan, Song Guo, Qinghua Zheng, and Xuanyu Wang. 2022. A survey on deploying mobile deep learning applications: A systemic and technical perspective. *Digital Communications and Networks* 8, 1 (Feb. 2022), 1–17. https://doi.org/10.1016/j.dcan.2021.06.001

[66] Zhen Wang, Weirui Kuang, Ce Zhang, Bolin Ding, and Yaliang Li. 2022. FedHPO-B: A Benchmark Suite for Federated Hyperparameter Optimization. https://doi.org/10.48550/ARXIV.2206.03966

[67] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farokhi Farhad, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2019. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. https://doi.org/10.48550/ARXIV.1911.00222

[68] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature Communications* 13, 1 (April 2022). https://doi.org/10.1038/s41467-022-29763-x

[69] Yuexiang Xie, Zhen Wang, Dawei Gao, Daoyuan Chen, Liuyi Yao, Weirui Kuang, Yaliang Li, Bolin Ding, and Jingren Zhou. 2022. FederatedScope: A Flexible Federated Learning Platform for Heterogeneity. https://doi.org/10.48550/ARXIV.2204.05011

[70] Jie Xu and Heqiang Wang. 2021. Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective. *IEEE Transactions on Wireless Communications* 20, 2 (Feb. 2021), 1188–1200. https://doi.org/10.1109/twc.2020.3031503

CC BY-NC-SA 4.0

Attribution-NonCommercial-ShareAlike 4.0 International

Deed

Canonical URL : https://creativecommons.org/licenses/by-nc-sa/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes .

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Appendix B

Federated Fine-tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly

Printed with the permission of

Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly." In: *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning*. DEEM '24. Santiago, Chile: Association for Computing Machinery, 2024, pp. 39–50. ISBN: 9798400706110. DOI: 10.1145/3650203.3663331. URL: https://doi.org/10.1145/3650203.3663331

Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly

Herbert Woisetschläger
herbert.woisetschlaeger@tum.de
Technical University of Munich
Munich, Germany

Alexander Erben
alex.erben@tum.de
Technical University of Munich
Munich, Germany

Shiqiang Wang
wangshiq@us.ibm.com
IBM T.J. Watson Research Center
Yorktown Heights, United States

Ruben Mayer
ruben.mayer@uni-bayreuth.de
University of Bayreuth
Bayreuth, Germany

Hans-Arno Jacobsen
jacobsen@eecg.toronto.edu
University of Toronto
Toronto, Canada

ABSTRACT

With the emergence of AI regulations, such as the EU AI Act, requirements for simple data lineage, enforcement of low data bias, and energy efficiency have become a priority for everyone offering AI services. Being pre-trained on versatile and a vast amount of data, large language models and foundation models (FMs) offer a good basis for building high-quality deep learning pipelines. Fine-tuning can further improve model performance on a specific downstream task, which requires orders of magnitude less data than pre-training. Often, access to high-quality and low-bias data for model fine-tuning is limited due to technical or regulatory requirements. Federated learning (FL), as a distributed and privacy-preserving technique, offers a well-suited approach to significantly expanding data access for model fine-tuning. Yet, this data is often located on the network edge, where energy, computational, and communication resources are significantly more limited than in data centers.

In our paper, we conduct an end-to-end evaluation for fine-tuning the FLAN-T5 FM family on the network edge. We study energy efficiency potentials throughout FL systems - on clients, in communication, and on the server. Our analysis introduces energy efficiency as a real-time metric to assess the computational efficiency of an FL system. We show the stark need for further improvements in communication efficiency when working with FMs and demonstrate the importance of adaptive FL optimizers for FM training.

ACM Reference Format:
Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. 2024. Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly. In *Workshop on Data Management for End-to-End Machine Learning (DEEM 24), June 9, 2024, Santiago, AA, Chile.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3650203.3663331

DEEM 24, June 9, 2024, Santiago, AA, Chile

ACM ISBN 979-8-4007-0611-0/24/06
https://doi.org/10.1145/3650203.3663331

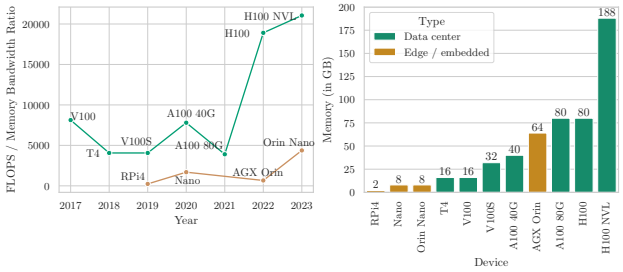


Figure 1: Development of computational power and resource availability of DL accelerators 2017 - 2023 for data centers and embedded systems. Key: RPi4 = Raspberry Pi 4, Nano = NVIDIA Jetson Nano, Orin Nano = NVIDIA Jetson Orin Nano, AGX Orin = NVIDIA Jetson AGX Orin 64 GB.

1 INTRODUCTION

Large Language Models (LLMs) and Foundation Models (FMs) are omnipresent in academia and practice and fuel new innovations [6]. These models have grown significantly with regard to parameter size, as more parameters improve the performance to a certain degree [18]. In line with the growing computational need for these models, deep learning (DL) hardware accelerators have become increasingly more capable. Recent developments indicate a generational leap in computational power for data center applications, with the NVIDIA H100 NVL delivering 7.8 TB/s memory bandwidth compared to the previous state-of-the-art A100 80GB GPU that only has 2 TB/s (Figure 1). Due to memory-bandwidth bottlenecked operations taking up to 40% of the training time [22], this improvement may lead to much faster training times for both small and large models. At the same time, computational capabilities on embedded devices for mobile edge computing are significantly growing, with the NVIDIA Jetson AGX Orin 64GB being the first-of-a-kind DL-accelerated embedded device that provides capabilities for training FMs [9]. This has never been possible before and enables us to build FL workloads with large transformer models, benefit from scattered data, and bring generative AI closer to users, all the while improving data privacy.

As this type of device is oftentimes scattered across geographies and entities, federated DL (FL) imposes itself as a well-suited technique for fine-tuning FMs in a distributed and private fashion. To

our knowledge, the largest models discussed in FL to this point entail FedBert and GPT2 [39, 48]. Both models were trained with FL methods on multi-GPU data center nodes. Only a few studies exist on field deployments [4]. As can be seen in Figure 1, the computing resources of state-of-the-art embedded hardware like the NVIDIA Jetson AGX Orin are orders of magnitude less than on a modern data center GPU like the H100. However, if we want to gain access to a broader data basis, we need to foster FL on the edge and bring FMs to embedded devices.

At the same time, new regulations like the European Union AI Act impose new limitations and requirements for FL [43] that need to be met such that applications become practical. This entails the need to prioritize energy efficiency. For FL, this is an underexplored area as most existing works either perform microbenchmarks for FL clients [27, 5] or focus on communication cost reduction [13].

The combination of resource limitations and increasing regulatory requirements presents us with a set of challenges:

(1) **Comparably low memory bandwidth on embedded devices limits the compute potential of FL applications on the edge**. We currently see a generation leap in data center DL accelerators regarding memory bandwidth, which has increased significantly (up to 7.8 TB/s). Even though the memory size on embedded devices has increased, the memory bandwidth remains comparatively low (up to 0.2 TB/s). This affects key memory-bandwidth bottlenecked operations for the training process, which could lead to severe training time penalties.

(2) **Energy efficiency has become a priority with the introduction of the EU AI Act.** The new regulation requires service providers that offer FL applications to focus on energy-efficient operations. To this point, energy efficiency can be measured by means of the Model-FLOP utilization [8], but it requires knowledge of what hardware is being used and how to optimally configure it. In FL systems, this can be impractical as we often do not know client details, and in many cases, clients are likely to participate only once in the training process [32].

(3) **FMs are large in size and are harder to train or fine-tune than small models**. The prevailing benefit of FMs is their ability to cater to many different tasks [6]. However, their performance on specific downstream tasks can be improved with fine-tuning [20]. Yet, the gradients during foundation model training or fine-tuning are at much higher risk of exploding or vanishing than in smaller tasks, as they are frequently discussed in FL [27, 17, 7].

(4) **Communication on the edge is significantly more expensive than in data centers**. While network bandwidth in data centers is available at 100 Gbit [2], mobile or remote communication over wide area networks is still a difficult challenge to achieve, especially when handling 100M+ parameter DL models. For distributed learning applications where high-bandwidth communication is available, we can use techniques such as ZeRo-offloading [37] and FSDP [50] that utilize a high-bandwidth interconnect for all-reduce communication between nodes to not materialize the full model, optimizer, and gradient state due to limited memory sizes. In FL, this is typically infeasible due to a limited network interconnect.

Based on the open challenges to creating efficient edge computing systems capable of training FMs, we formulate our research question: How can we efficiently realize FM training and fine-tuning at the network edge? Which levers can have the largest impact on improving FL system efficiency?

By exploring this research question, we make four major contributions to bridge the gap between federated foundation model training and energy-aware FL:

(1) **We systematically study the computational limitations of state-of-the-art embedded hardware for DL**. Nowadays, most papers in the FL space use data center hardware for their experiments [17, 7], while large amounts of data are scattered on the edge and must not be neglected as a field of application. We, therefore, conduct an in-depth micro-benchmark of various transformer models on the latest embedded and datacenter DL accelerators to identify computational bottlenecks.

(2) **We outline the limitations of theoretical metrics such as the Model-FLOP Utilization for FL applications**. As micro-benchmarks require extensive experimentation, practitioners often use metrics, such as the Model-FLOP Utilization (MFU), based on theoretical hardware performance limits to assess the computational efficiency of algorithms [11]. Calculating the MFU requires knowledge of hardware specifications on FL clients, which could be infeasible due to privacy considerations. As such, we identify energy efficiency as a readily available alternative to the MFU and outline that the computational limits of embedded AI accelerators appear significantly earlier than the MFU suggests.

(3) **We benchmark four state-of-the-art FL optimizers for FM fine-tuning**. A key to energy-efficient use of FL is the right optimizer choice. We systematically benchmark four state-of-the-art FL optimizers to quantify the energy savings with the right optimizer choice. We find adaptive optimization techniques to converge up to 8× faster than FedAvg with momentum, one of the most widely used FL optimizers [29].

(4) **We quantify the *total* cost of communication in FL applications with state-of-the-art FMs**. Our study identifies wide-area communication as the primary driver for energy consumption in FL systems, up to 4 orders of magnitude higher than the energy consumed by computing on clients.

This paper is structured as follows. Section 2 will outline relevant background. In Section 3, we present our methodology, and in Section 4, we present our benchmark design, including datasets, DL models, and FL strategies. Section 5 contains experimental evaluations of our benchmark. In Section 6, we present related work. In Section 7, we discuss our results. In Section 8, we conclude our work.

2 BACKGROUND

2.1 Performance Objectives in Data Center Environments

One of the most important issues when training in a data center is to maximize throughput by trying to use the hardware to its limit without being blocked by communication. Communication concerns both local communication, i.e., memory movement, and communication between GPUs and nodes, typically with a high bandwidth interconnect such as NVLink (7.8 TB/s) and Ethernet (100 Gbit) [2].

Measuring the effectiveness of each GPU in training is possible via Model FLOP Utilization (MFU) [8], which is the ratio of throughput achieved compared to the theoretical throughput of a model and a set of hardware. Common values for the MFU are between 5 – 20% (Figure 4a) because DL models are not defined as a single matrix multiplication that can be perfectly parallelized between tensor cores but as many operations with memory bandwidth bottlenecks such as softmax, residual additions, and activations [22]. These operations result in a FLOP usage significantly lower than the theoretical hardware capability, and each model architecture has its own set of operations that slow down throughput. However, the MFU can be used as a benchmark for how well a model is suited to work on a particular piece of hardware, as it fits the trade-offs between memory bandwidth, memory capacity, and FLOP. This way, we can compare the MFU for the same model on different hardware and contrast their results.

2.2 Performance Objectives on the Edge

In edge computing systems that involve embedded devices, performance considerations differ from those in data center environments as use cases typically vary [41]. Yet, to run FL workloads on embedded devices on the edge, we need to unite performance characteristics from data centers and edge computing.

Running FL workloads on the edge is all about minimizing the time we use a client's hardware and maximizing the throughput. Yet, the hardware is often located in remote areas with limited access to power or even on mobile devices with very restrictive battery management [34]. Also, in remote and mobile environments, network bandwidth utilization and total network traffic are critical. Both have a significant impact on communication latency, i.e., how fast we can move model weights between clients and a server. For foundation models, both can become a hurdle, as this kind of model tends to grow beyond several hundreds of millions of parameters in size or, in other words, beyond 1 GB in parameters to transfer over the network. Putting that into perspective with the average available wireless network bandwidth of 50 Mbit/s on mobile devices [40] yields communication times substantially longer than the actual computation time on clients [5].

2.3 Regulatory Requirements with Regard to Energy Efficiency

As we see regulatory frameworks on Artificial Intelligence emerge and be passed as laws, the first legislation to come into effect by 2024 is the EU AI Act [10]. Other countries have declared the EU AI Act as a lighthouse framework; they aim to align their individual frameworks with it [19]. A priority in the EU AI Act is energy efficiency [43]. The objective is to promote sustainable computing practices by holding service providers liable for monitoring energy consumption and subsequently fostering the energy efficiency of an FL system. As such, it is vital to understand where and how efficiency potentials can be lifted such that the practical applicability of FL is improved.

Algorithm 1: Federated Adam with decoupled weight decay (FedAdamW)

Given: set of clients $K \in \mathbb{N}^+$, server rounds $s \in \mathbb{N}^+$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\tau = 10^{-6}$, $\lambda \in \mathbb{R}$

Initialize: server round $t \leftarrow 0$, initial parameters $x_{t=0} \in \mathbb{R}$, first momentum vector $m_{t=0} \leftarrow 0$, second momentum vector $v_{t=0} \leftarrow 0$, server learning rate $\eta_s \in \mathbb{R}$, client learning rate $\eta_c \in \mathbb{R}$

for t **to** s **do**

`// server-side`
Sample subset $k \in K$
$x_i^t = x_t$
Distribute model weights to clients

`// client-side (is identical with FedAvg)`
for $i \in k$ ***in parallel*** **do**
$g_i^t \leftarrow \nabla F_i(x_i^t)$ `// compute local gradient`
$x_i^{t+1} = x_i^t - \eta_c \cdot g_i^t$ `// update client model`
Send model update x_i^{t+1} *to server*
end

`// server-side optimization`
$x_{t+1} \leftarrow \frac{1}{|k|} \sum_i^{|k|} x_i^{t+1}$ `// FedAvg`
$g_t \leftarrow x_t - x_{t+1}$ `// pseudo gradient`
$m_{t+1} \leftarrow \beta_1 \cdot m_t + (1 - \beta_1) \cdot g_t$ `// momentum`
$v_{t+1} \leftarrow \beta_2 \cdot v_t + (1 - \beta_2) \cdot g_t^2$ `// velocity`

`// FedAdamW`
$x_{t+1} \leftarrow x_t - \eta_s \cdot \lambda x_t$ `// decoupled weight decay`
$\hat{m}_{t+1} \leftarrow \frac{m_{t+1}}{1-\beta_1^t}$ `// regularized momentum`
$\hat{v}_{t+1} \leftarrow \frac{v_{t+1}}{1-\beta_2^t}$ `// regularized velocity`
$x_{t+1} \leftarrow x_{t+1} - \frac{\eta_s \cdot \hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \tau}$
$t \leftarrow t + 1$ `// update FL round`

end

Result: optimized parameters x_t

3 METHODOLOGY

Generally, when transferring LLM fine-tuning to the edge, we also transfer the challenges we currently have in data center environments into systems that suffer from more severe resource limitations. While energy efficiency is a specific challenge to edge computing systems, network bandwidth and computational efficiency are frequently discussed topics for DL applications in data centers.

With our hardware-centric study, we aim to provide a comprehensive perspective on energy efficiency levers in FL systems on the edge to foster sustainable computing and, subsequently, legal compliance with the EU AI Act. To do so, we organize our methodology along the following four pillars to cover the end-to-end training pipeline.

3.1 Computational Efficiency

By studying the behavior of state-of-the-art FL clients when it comes to scaling on-device training with varying model sizes and minibatch sizes, we aim to understand how the training steps (forward, loss calculation, `opt.step()`, and backward) differ between data center resources and clients deployed on the network edge. Maximization of resource utilization is the superior objective for DL and FL applications in data center environments, as this is usually equivalent to a cost-optimal solution [14]. In the HPC domain, MFU is used to calculate the hardware resource utilization based on the number of theoretical hardware FLOP/s. By varying the minibatch size, the MFU can also be used to identify computational bottlenecks, i.e., whether we are computationally bound or memory bandwidth limited. In our experiments, the theoretical capacity of the NVIDIA A100 is 312 TFLOP (at FP32), while the Jetson AGX Orin 64 GB provides 42.5 TFLOP or 13% of the A100. As such, the MFU is well suited for in-depth analysis but requires full knowledge of client hardware specifications and the availability of performance metrics. However, in FL systems, clients are often heterogeneous regarding their hardware and considered ephemeral, i.e., they are likely to participate in training only once [32].

3.2 Energy Efficiency

With the imminent legal requirements to focus on energy efficiency, practical FL system design must include energy monitoring, regardless of whether FL clients are deployed in a data center or on the network edge. Yet, in edge computing, energy efficiency has been a priority for a long time [38, 47, 51]. A major benefit of clients on the network edge, such as NVIDIA Jetson AGX Orin, is their hardware design since they are often created as a system-on-a-chip (SoC) and contain hardware-based power measurement units for each component (e.g., CPU, GPU). In contrast to MFU, which requires detailed hardware knowledge, energy metrics are likely to be readily available and easy to measure across all clients. We define energy efficiency as the tokens per second (TPS) throughput over the average power draw (W) for a workload,

$$\eta_e = \frac{\text{TPS}}{W}. \tag{1}$$

3.3 Communication Efficiency

Communication is equally important for federated LLM fine-tuning as computational efficiency. Typically, full models or partial model weights are communicated between client and server [33]. Yet, communication on data center settings is built on top of high-performance networking infrastructure that enables bandwidths of 100 Gbit and more [2]. On the edge, we often find significantly slower network links with 1 Gbit and below. For instance, the global average for communication over 4G LTE wireless is 40 Mbit download and 15 Mbit upload [40].

We need a reliable metric to quantify communication efficiency that, at the same time, tells us whether it is useful to further scale a FL workload over more clients or not. Borrowing from the HPC domain, Granularity (G) measures the ratio between the time it takes to compute a DL task (T_{comp}) and to communicate the model gradients or weights (T_{comm}) [21]. It is defined as

$$G = \frac{T_{\text{comp}}}{T_{\text{comm}}}. \tag{2}$$

In our FL scenario, the computation time is the maximum fine-tuning time on a client in each round, and the communication time is the time spent sending the model state, waiting, and receiving the aggregated model state from the server. In general, $G \gg 1$ indicates that adding one more client to a system has a positive effect on the total processing speed (higher throughput). $G \ll 1$ is an indicator for communication times significantly outweighing computation times and, therefore, no positive effect on system throughput. As such, we use G as the evaluation metric to evaluate the practical utility of federating an FL application.

In addition to scalability considerations, we evaluate communication costs when deploying FL applications to the network edge. To do so, we consider two scenarios. First, we look at a mobile edge computing scenario where clients are connected via an LTE wireless connection [1], which exhibits download and upload speeds of 40 and 10 Mbit, respectively [40]. Second, we consider a scenario where FL clients are operating at the network edge with a wired 1 Gbit connection that is often found in factory settings [26]. We use the *per-bit communication model* to estimate the total communication cost of our FL pipelines [43, 46, 23]. It is important to note that once wireless communication is involved, the energy consumption for communication increases by two orders of magnitude [23]. A detailed explanation and the exact parameterization of the per-bit communication model are available in Section A.

3.4 Model Performance

We use four widely used federated optimizers: (I) Federated Averaging (FedAvg) [33], (II) FedAvg with Momentum (FedAvgM) [29], (III) Federated Adam (FedAdam) [36], and (IV) we introduce FedAdam with decoupled weight decay (FedAdamW). The objective of each optimizer is to minimize the loss of a given neural network, typically done by stochastic gradient descent (SGD). All four optimizers share SGD as the optimization basis.

FedAvg is used to control the communication efficiency of an FL application as it allows training over multiple minibatches on a client before communicating a model update. With federated SGD, we would have to communicate after each minibatch [33]. However, as soon as we encounter high change rates in gradients, as is often the case when working with foundation models, we require adaptive control over the model learning rate [25]. FedAvgM introduces (first-order) momentum regularization to reduce the impact of early gradient and stabilize training. Yet, often, this is not enough, as reducing the momentum too much slows down model convergence towards the end of the training; subsequently increasing training costs [25]. For this, Reddi et al. [36] introduced FedAdam among other federated optimizers. Additionally to momentum, FedAdam uses velocity (second-order momentum) to further regularize gradients. Even though FedAdam may provide faster model convergence,

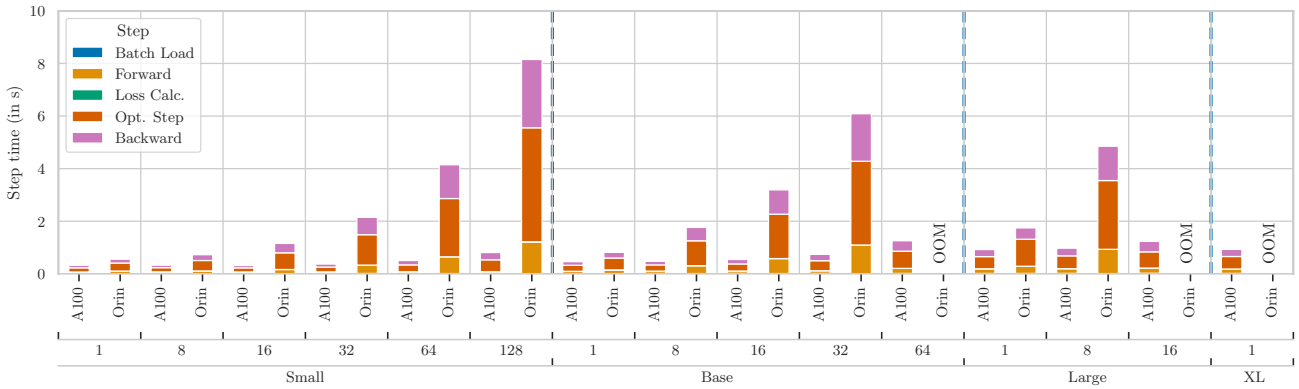


Figure 2: DL training step times across FLAN-T5 transformer models with varying minibatch sizes on the Samsum dataset running on the NVIDIA A100 and Jetson AGX Orin platform. Detailed metrics are available in Appendix Section C.

similar to its centralized counterpart Adam, it is challenged by a worse generalization over new data than SGD.

To tackle this challenge, we implement federated Adam with decoupled weight decay (FedAdamW) based on Loshchilov and Hutter [30] (Algorithm 1). A theoretical analysis for decoupling weight decay in adaptive optimization algorithms is provided by Jin et al. [24]. The objective is to use weight decay as an additional optimization technique to penalize large or vanishing gradients. To evaluate the effectiveness of each optimization technique, we study the validation loss and the Rouge-1 score. In natural language processing research, the Rouge-1 score evaluates unigram overlaps between a model-generated output and a reference text. Generally, a Rouge-1 score of 50 is considered a strong result. It is a derivative of the F1-Score known from classification tasks (as often encountered in computer vision research) [28].

4 EXPERIMENTAL SETUP

Our hardware-centric study for FL on the edge focuses on evaluating state-of-the-art DL workloads on embedded devices. As such, we focus on a state-of-the-art FM family.

Evaluation hardware. In our hardware-centered study, we focus on state-of-the-art deep learning accelerators for embedded and data center computing. We employ a cloud VM with a single NVIDIA A100 80 GB (SXM4) as a data center node (A100) to perform our local baseline experiments. Further, we use a dedicated cluster consisting of ten NVIDIA Jetson AGX Orin 64 GB nodes (Orin) as the only state-of-the-art embedded computing platform that provides enough computational resources for training FMs. The Orins are connected with a 1 Gbit synchronous network link and are monitored with 2 Hz for their power metrics (Figure 3). For our FL experiments, we use a GPU-accelerated VM co-located with the Orins to handle the model aggregation and testing of the global model. For all of our experiments, we do not limit hardware capabilities.

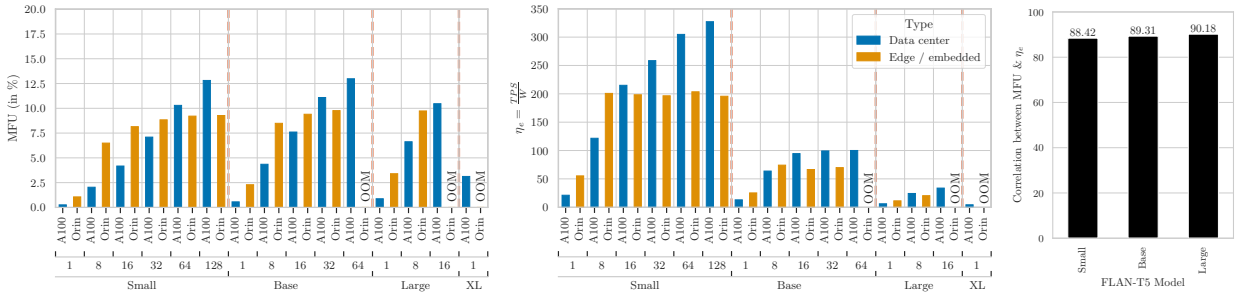
DL models. For our experiments, we adopt the FLAN-T5 transformer model family [9] for conditional text generation. Even though the FLAN-T5 models' parameter sizes are small compared to other state-of-the-art FMs, they often provide the best-in-class

Figure 3: Our NVIDIA Jetson AGX Orin 64GB Testbed. 10 devices with freely configurable network interconnect up to 10 Gbit. Active external cooling is a must at the given energy density of $10 \cdot 60W$ max. power draw.

performance [15]. We evaluate the computational training performance of the FLAN-T5-Small model with 80M parameters or 308 MB in size, the FLAN-T5-Base model with 250M parameters (990 MB), the FLAN-T5-Large model with 783M parameters (3.1 GB), and the FLAN-T5-XL model with 3B parameters (11.4 GB). For all models, we use their corresponding pre-trained tokenizers. For each model, we apply parameter-efficient fine-tuning (PEFT) in the form of Low-Rank Adaptation (LoRA), which is used to reduce the number of trainable parameters to $< 1\%$ of all model parameters [20]. We parameterize LoRA for the FLAN-T5 model family as follows: $r = 16$, $\alpha_l = 32$, dropout $= 0.05$. We do not fine-tune the LoRA bias.

Dataset. All the FLAN-T5 models are fine-tuned on the Samsum dataset with the objective of summarizing texts with a maximum token length of 512 elements [16]. The maximum model output length is 95 tokens, which can be translated into the summaries of the respective inputs. For our FL experiments, we choose to sample to the number of samples per client subset from a Dirichlet distribution as it is frequently used in related work [27, 17, 7].

FL setup. We use a Dirichlet $\alpha_d = 1$ to randomly split the Samsum dataset into 100 subsets that we distribute on the Orin compute cluster. We train all FLAN-T5 models until they overfit the Samsum



(a) MFU in %

(b) η_e in $\frac{\text{TPS}}{W}$

(c) Pearson Correlation

Figure 4: We study the model FLOP utilization (MFU) and the energy efficiency (η_e) of the FLAN-T5 transformer model family and find a strong correlation between the MFU and η_e, which is useful to evaluate root causes for poor training speeds in real-time.

dataset or have seen 60,000 samples. In each FL round, we let 10 physical clients participate, i.e., we have a participation rate of 10%. For each round, we perform 2 local training steps on each client before communicating with the server. The client-side optimizer is SGD with a learning rate of 1.0 and no momentum in all experiments. On the server side, we employ FedAvg, FedAvgM, FedAdam, and FedAdamW. The exact hyperparameters for the server-side optimizer can be drawn from Table 3 in Section B.

5 RESULTS

Our results are organized so that we first understand the computational limitations of what foundation model sizes can be deployed at the edge. We analyze to what point scaling primitives, as we know them from data center environments, hold true at the network edge. We then study computational bottlenecks and show the limitations of theoretical analysis. Next, we investigate the energy efficiency of training with varying minibatch sizes and point out how energy efficiency can be used to estimate the most energy-efficient on-device training configuration. We round off our systems analysis with a communication cost estimation. Lastly, we study the importance of decoupled weight decay when training state-of-the-art foundation models and quantify the cost savings.

5.1 Computational & Energy Efficiency

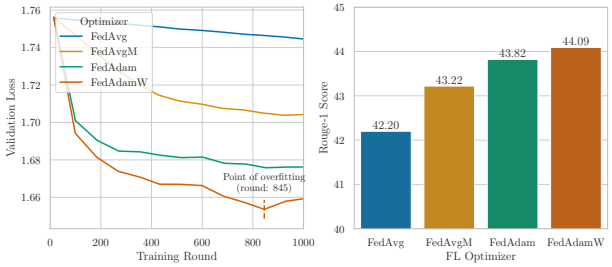
When designing FL systems, we have to anticipate what type of clients will be participating and how we can optimally use their training performance to receive a trained model in a timely manner. This is especially relevant for edge computing environments where client hardware is significantly distinct from data center hardware typically used to pre-train and prepare foundation models before using them in a federated setting [49, 3].

Increasing the minibatch size on embedded devices does not scale well. To understand local training performance in detail, we measure the timing using a microbenchmark (Figure 2). We find linearly growing `opt.step()` times for all models as we scale the minibatch size on the Orins, while the step times on the A100 platform scale logarithmically with increasing batch size (Figure 2). The same is true for the backward step.

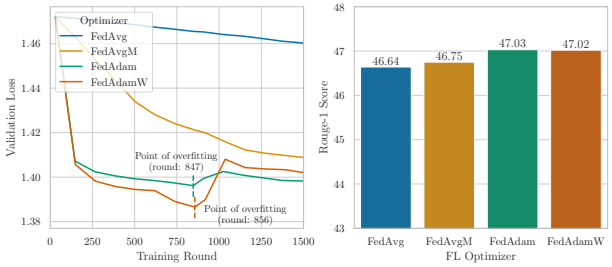
The Orin platform is severely bottlenecked by memory bandwidth compared to the A100. To develop an understanding of what needs to be done to move from linear to logarithmically growing training times, we look at the MFU (Figure 4a). The MFU can explain whether training exhibits a computational or memory bottleneck on an FL client. Throughout all experiments, we find a stagnating MFU for the Orins as we scale the minibatch size, while on the A100 the MFU steadily grows. As such, on the embedded platform, we reach the maximum theoretical computational efficiency with a minibatch size of 64 for FLAN-T5 Small, 32 for FLAN-T5 Base, and 8 for FLAN-T5 Large. Overall, a stagnating MFU as minibatch sizes increase means that increased parallel computation potential does not result in additional used FLOPs. This can only happen if we encounter a memory bandwidth bottleneck. We see from Figure 2 that the Orin `opt.step()` function updating model weights and biases is taking up a significant amount of time in comparison to A100, which suggests that its performance is highly dependent on memory bandwidth.

With our proposed energy efficiency metric η_e, we enable real-time monitoring of computational efficiency on the client-level. We study η_e and MFU of the NVIDIA A100 and Jetson AGX Orin across the FLAN-T5 transformer family. For the FLAN-T5-Small model, as we scale the batch size, we notice an increasing η_e until a minibatch size of 8 (Figure 4b). Afterwards, η_e remains constant, i.e., scaling the minibatch size further does not yield any performance benefits. The A100, for the same set of experiments, consistently scales with increasing minibatch size. The evaluation of the MFU on the same set of experiments as for η_e unveils an identical trend (Figure 4a). The correlation between the MFU and η_e originates from both metrics being tied to power draw via FLOPs and Tokens per Second (TPS), respectively.

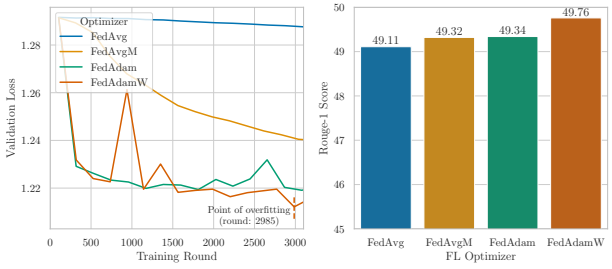
We reach the computational limits of state-of-the-art deep learning accelerators much earlier than theoretical analysis indicates. While the MFU suggests we should scale the minibatch size on the Orins up to 128 samples, we find that the actual memory bottlenecks appear at much smaller minibatch sizes already. When evaluating the energy efficiency during training, we found that we had already reached the highest efficiency on the Orins with a smaller minibatch size of 8 for all models compared to the A100. As such, we identified the practical computational limits of state-of-the-art embedded devices for DL.



(a) FLAN-T5 Small



(b) FLAN-T5 Base



(c) FLAN-T5 Large

Figure 5: We show the effectiveness of Federated AdamW by training the FLAN-T5 Model family in a federated setup with 100 clients (10 clients per round). We report the validation loss (left) and the Rouge-1 score (right) as performance indicators. Note: The loss spikes for FLAN-T5 Large originate from an increased sensitivity of LoRA adapters with a large parameter count to non-IID data [3].

5.2 Model Performance

Equally important to the hardware performance side is evaluating state-of-the-art FL optimizers on the algorithmic side, as this affects the energy efficiency of the entire FL system as well. We find adaptive optimization to be vital for FMs in FL applications.

Federating AdamW accelerates model convergence and saves cost and time compared to other widely-used FL optimizers. We compare four commonly used FL optimizers to find out about their training efficiency and convergence speed with state-of-the-art foundation models (Figure 5). While the FLAN-T5 model family exhibits a slow convergence speed with FedAvg that is approximately three orders of magnitude slower than FedAdam or FedAdamW, we find notable training progress with the adaptive optimizers after 135, 150, and 340 rounds for FLAN-T5 Small, Base, and Large, respectively. Also, for FLAN-T5 Base and Large, the achievable loss with FedAdamW is lower than with FedAdam before the model starts to overfit the Samsum dataset. As such, applying the state of the art for optimization in an FL application yields not only time and cost benefits but also improves model quality at the same time.

Table 1: Communication cost analysis when training the FLAN-T5 model family in an FL system with FedAdamW until the minimal loss is achieved. $G \gg 1$ suggests that a model is well-suited for FL at scale. kWh denotes the power consumption incurred during communication per FL round based on the per-bit communication model.

			FLAN-T5 Small (845 FL rounds)		FLAN-T5 Base (856 FL rounds)		FLAN-T5 Large (2985 FL rounds)	
Training	**Comm.**	**Device**	G	kWh	G	kWh	G	kWh
Full Model	LTE	A100	0.01	0.81	0.00	2.60	0.0	8.22
		Orin	0.03		0.01		0.0	
	1 Gbit	A100	14.90	0.13	4.64	0.40	1.47	1.27
		Orin	32.90		10.23		3.24	
PEFT	LTE	A100	1.70	0.01	0.65	0.02	0.25	0.05
		Orin	3.70		1.44		0.54	
	1 Gbit	A100	1690.90	< 0.01	664.29	< 0.01	244.74	0.01
		Orin	3727.30		1464.29		539.47	

5.3 Communication Efficiency

As we have shown, the computational optimization potential on state-of-the-art embedded hardware is limited. As such, it is key to consider the cost of communication during FL training and study how well a model can scale under limited communication.

PEFT significantly improves the scalability of FL systems, regardless of whether bandwidth-limited wireless communication is involved. During our experiments, we find PEFT improves G by up to 110× as compared to full model training (Table 1). This originates from a compounding effect that is beneficial in FL setups. Not only does PEFT reduce the demand for computational resources (esp. GPU memory), but by reducing the number of trainable parameters to $< 1\%$ of the total parameter count, it also reduces communication by $> 99\%$. Due to the relatively higher timeshare of computation compared to communication, this significantly increases G, indicating better scalability of an FL application regardless of the communication technology. At the same time and as expected, full model fine-tuning is only viable in environments that benefit from a high network bandwidth, often absent in FL.

6 RELATED WORK

We divide our related work section into two major streams of work. One is foundation model training with FL, and the other is energy-aware or energy-efficient FL.

Foundation model training with FL. With FATE-LLM, Fan et al. [12] present an extension of the FATE FL framework to train foundation models, specifically large language models, in a federated setting. They introduce a broad range of foundation models and parameter-efficient training techniques with a brief evaluation of the communication benefits of parameter-efficient fine-tuning techniques by means of trainable parameters. Similarly, FedML [17] supports the training of foundation models in FL systems by providing

a wide range of models ready to use. At the same time, we see a wide variety of parameter-efficient FL methods emerge that tackle data heterogeneity and address resource limitations. SLoRA [3] presents a method to tackle the challenge of non-IID data by calibrating the LoRA parameterization in a warm-up phase over multiple rounds, achieving stronger model performance than LoRA without calibration. FwdLLM [44] enables backpropagation-free fine-tuning of foundation models with a special focus on reducing the memory footprint, enabling the training of models on resource-constrained clients.

Energy-aware and energy-efficient FL. Energy-aware FL system design has been discussed extensively, especially in the space of edge computing applications [38, 51, 47, 45, 42]. The objective is to reduce the communication cost to a minimum while not compromising model quality. Yet, most works neglect the full cost of communication as it was introduced by the per-bit communication model [23]. Yousefpour et al. [46] provide a holistic viewpoint on energy consumption of a wide range of Fl system configurations. Especially, asynchronous FL, which accelerates the training process based on increased training parallelism, incurs a significantly higher energy footprint than round-based FL.

To the best of our knowledge, there is no overlap between federated foundation model training and energy-aware FL yet. Our paper creates this link by evaluating state-of-the-art hardware for its capabilities to serve FL workloads involving foundation models and discusses what can be done to improve the overall energy efficiency of an FL system. Our evaluation underpins the importance of developing parameter-efficient training techniques for FL, not only to mitigate data heterogeneity effects but also to reduce energy consumption and improve computational efficiency.

7 DISCUSSION

With formal regulations for AI applications on the horizon, energy monitoring and building energy-efficient FL systems will soon become a necessity to comply with standards for modern AI systems and subsequently build trust with end users. Therefore, it is key to understand the benefits of FL when working with foundation models (the good), the open challenges (the bad), and what indirect effects FL has (the ugly).

The Good. By design, FL enables data parallel training of a shared model across geo-distributed clients. A major benefit is the access to a much broader range of data as compared to centralized learning where training data is challenging to acquire. At the same time, the privacy-preserving design of FL also supports building the trust of end users in FL applications since clients must not share their raw data. Overall, this helps improve the quality of foundation models on downstream tasks and lower the data bias as we continuously train over an evolving client basis.

The Bad. We do find state-of-the-art embedded devices for deep learning applications to be bottlenecked, which limits the applicability of optimization techniques that we know from deep learning in high-performance computing environments, especially as in data centers, the memory bandwidth of GPUs has increased significantly (e.g., with the NVIDIA H100). However, as we show in the introduction, the trend of increasing memory bandwidth has also started for embedded devices. Nonetheless, we can develop targeted optimizations for FL workloads on embedded devices such as the Orins by profiling what GPU kernels are responsible for the on-client memory bottleneck. Also, promising techniques such as 1.58-bit training of foundation models are capable of reducing the need for high memory bandwidth significantly [31]. Furthermore, recent research has shown that LoRA is more sensitive towards a non-IID data distribution, but adaptive methods for configuring LoRA are a promising direction to mitigate this challenge [3].

The Ugly. We show in our study that even though we apply PEFT for all FLAN-T5 models, the energy consumption incurred during training and attributable to communication is still significant. To put the energy consumption into perspective: Fine-tuning FLAN-T5 Large over the Samsum dataset is possible on a single GPU or even on a single Orin, neglecting the benefit of broader data access, which FL provides. With our configuration (Table 3), training on an A100 takes approximately 3.33 hours or 1.3 kWh of power, and on an Orin, it takes approximately 8.33 hours or 0.5 kWh. As such, fine-tuning FLAN-T5 Large consumes more energy for communicating model updates than for the computations. This points out the need for future research on even more communication-efficient FL methods than we currently have available. A promising step is gradient projection based on probability-differentiated seeds [35].

8 CONCLUSIONS

In our work, we conduct an end-to-end study for FL workloads focusing on energy consumption involving three foundation models. We point out the hardware limits of state-of-the-art embedded hardware for deep learning and put the performance into perspective with modern data center hardware to discover distinct performance characteristics. We further introduce η_e, the real-time metric to evaluate computational bottlenecks, as a drop-in replacement for MFU and show significant potential for on-client optimizations. Additionally, we show the effectiveness of η_e as a proxy for MFU.

To understand the impact FL optimizers have on overall energy consumption, we study three widely used FL optimizers and compare them with FedAdamW, which not only improves model convergence speed but also achieves higher model quality. Based on the FedAdamW experiments, we quantified the trade-off between communication and computation in FL systems with granularity.

This underpins the relevance of parameter-efficient training techniques to improve communication efficiency in FL systems and render foundation model training practical. In the course of our communication analysis, we also quantify the end-to-end energy consumption for communication in our FL experiments, showing that communication is orders of magnitude more energy-intensive than computation for an FL application. Putting the current state of FL research into context with emerging AI regulation, we find significant benefits of FL over centralized learning when it comes to data lineage and the potential for data bias mitigation but we have to pick up on research for energy-efficient FL system designs.

To conclude, we demonstrate the feasibility of fine-tuning foundation models in FL systems but we also hope to raise awareness of the substantial challenges that need to be overcome to enable foundation model training on a broad basis for systems suffering from limited computational and network resources.

ACKNOWLEDGEMENTS

This work is partially funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01), the German Federal Ministry for Economic Affairs and Climate Action (Grant: 16KN085729), and the German Research Foundation (DFG, Grant: 392214008).

REFERENCES

[1] 3GPP. 4g (lte advanced). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2585, 2008.

[2] Amazon Web Services (AWS). Amazon AWS p3 instance types. https://aws.amazon.com/ec2/instance-types/p3/, 2023. Accessed: 2023-09-27.

[3] Sara Babakniya, Ahmed Elkordy, et al. SLoRA: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=06quMTmtRV.

[4] Sebastian Baunsgaard, Matthias Boehm, and et al. Exdra: Exploratory data science on federated raw data. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD/PODS '21. ACM, June 2021. doi: 10.1145/3448016.3457549. URL http://dx.doi.org/10.1145/3448016.3457549.

[5] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. Flower: A friendly federated learning research framework, 2020. URL https://arxiv.org/abs/2007.14390.

[6] Rishi Bommasani, Drew A. Hudson, et al. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

[7] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018. URL http://arxiv.org/abs/1812.01097.

[8] Aakanksha Chowdhery, Sharan Narang, et al. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1), mar 2024. ISSN 1532-4435.

[9] Hyung Won Chung, Le Hou, et al. Scaling instruction-finetuned language models. 2022. doi: 10.48550/ARXIV.2210.11416. URL https://arxiv.org/abs/2210.11416.

[10] Council of the European Union. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL - LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, apr 2021. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206. Document 52021PC0206.

[11] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857, 2023. ISSN 2210-5379. doi: https://doi.org/10.1016/j.suscom.2023.100857. URL https://www.sciencedirect.com/science/article/pii/S2210537923000124.

[12] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models, 2023. URL https://arxiv.org/abs/2310.10049.

[13] Jie Feng, Lei Liu, Qingqi Pei, and Keqin Li. Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2687–2700, 2021.

[14] Nathan C. Frey, Baolin Li, Joseph McDonald, Dan Zhao, Michael Jones, David Bestor, Devesh Tiwari, Vijay Gadepally, and Siddharth Samsi. Benchmarking resource usage for efficient distributed deep learning. In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–8, 2022. doi: 10.1109/HPEC55821.2022.9926375.

[15] Xue-Yong Fu, Md Tahmid Rahman Laskar, et al. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization?, 2024. URL https://arxiv.org/abs/2402.00841.

[16] Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu, editors, *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409.

[17] Chaoyang He, Songze Li, et al. Fedml: A research library and benchmark for federated machine learning, 2020. URL https://arxiv.org/abs/2007.13518.

[18] Jordan Hoffmann, Sebastian Borgeaud, et al. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

[19] House Of Commons of Canada. An Act to enact the Consumer Privacy Protection Act, the Personal Information and Data Protection Tribunal Act and the Artificial Intelligence and Data Act and to make consequential and related amendments to other Acts, 6 2022. URL https://www.parl.ca/DocumentViewer/en/44-1/bill/C-27/first-reading.

[20] Edward J Hu, Yelong Shen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[21] Kai Hwang. *Advanced Computer Architecture: Parallelism,Scalability,Programmability*. McGraw-Hill Higher Education, 1st edition, 1992. ISBN 0070316228.

[22] Andrei Ivanov, Nikoli Dryden, Tal Ben-Nun, Shigang Li, and Torsten Hoefler. Data movement is all you need: A case study on optimizing transformers. *Proceedings of Machine Learning and Systems*, 3:711–732, 2021.

[23] Fatemeh Jalali, Rob Ayre, Arun Vishwanath, Kerry Hinton, Tansu Alpcan, and Rod Tucker. Energy consumption of content distribution from nano data centers versus centralized data centers. *ACM SIGMETRICS Performance Evaluation Review*, 42(3):49–54, December 2014. ISSN 0163-5999. doi: 10.1145/2695533.2695555. URL http://dx.doi.org/10.1145/2695533.2695555.

[24] Jiayin Jin, Jiaxiang Ren, Yang Zhou, Lingjuan Lyu, Ji Liu, and Dejing Dou. Accelerated federated learning with decoupled adaptive optimization. 2022. doi: 10.48550/ARXIV.2207.07223. URL https://arxiv.org/abs/2207.07223.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[26] Kacper Kubiak, Grzegorz Dec, and Dorota Stadnicka. Possible applications of edge computing in the manufacturing industry—systematic literature review. *Sensors*, 22(7):2445, March 2022. ISSN 1424-8220. doi: 10.3390/s22072445. URL http://dx.doi.org/10.3390/s22072445.

[27] Fan Lai, Yinwei Dai, Sanjay S. Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. FedScale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning (ICML)*, 2022.

[28] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

[29] Wei Liu, Li Chen, et al. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8):1754–1766, 2020. doi: 10.1109/TPDS.2020.2975189.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[31] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024. URL https://arxiv.org/abs/2402.17764.

[32] Grigory Malinovsky, Samuel Horváth, Konstantin Pavlovich Burlachenko, and Peter Richtárik. Federated learning with regularized client participation. In *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. URL https://openreview.net/forum?id=6CDBpf7kNG.

[33] Brendan McMahan, Eider Moore, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[34] Samuel S. Ogden and Tian Guo. MODI: Mobile deep inference made efficient by edge computing. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, July 2018. USENIX Association. URL https://www.usenix.org/conference/hotedge18/presentation/ogden.

[35] Zhen Qin, Daoyuan Chen, et al. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes, 2023. URL https://arxiv.org/abs/2312.06353.

[36] Sashank J. Reddi, Zachary Charles, et al. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

[37] Jie Ren, Samyam Rajbhandari, et al. Zero-offload: Democratizing billion-scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564, 2021.

[38] Swapnil Sadashiv Shinde, Arash Bozorgchenani, Daniele Tarchi, and Qiang Ni. On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems. *IEEE Transactions on Vehicular Technology*, 71(2): 2041–2057, 2021.

[39] Yuanyishu Tian, Yao Wan, Lingjuan Lyu, Dezhong Yao, Hai Jin, and Lichao Sun. FedBERT: When federated learning meets pre-training. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–26, August 2022. doi: 10.1145/3510033. URL https://doi.org/10.1145/3510033.

[40] Martino Trevisan, Ali Safari Khatouni, and Danilo Giordano. Errant: Realistic emulation of radio access networks. *Computer Networks*, 176: 107289, July 2020. ISSN 1389-1286. doi: 10.1016/j.comnet.2020.107289. URL http://dx.doi.org/10.1016/j.comnet.2020.107289.

[41] Blesson Varghese, Nan Wang, David Bermbach, Cheol-Ho Hong, Eyal De Lara, Weisong Shi, and Christopher Stewart. A survey on edge performance benchmarking. *ACM Comput. Surv.*, 54(3), apr 2021. ISSN 0360-0300. doi: 10.1145/3444692. URL https://doi.org/10.1145/3444692.

[42] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *Ieee Network*, 33 (5):156–165, 2019.

[43] Herbert Woisetschläger, Alexander Erben, Bill Marino, Shiqiang Wang, Nicholas D. Lane, Ruben Mayer, and Hans-Arno Jacobsen. Federated learning priorities under the european union artificial intelligence act, 2024. URL https://arxiv.org/abs/2402.05968.

[44] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. Fwdllm: Efficient fedllm using forward gradient, 2023. URL https://arxiv.org/abs/2308.13894.

[45] Yunfan Ye, Shen Li, Fang Liu, Yonghao Tang, and Wanting Hu. Edgefed: Optimized federated learning based on edge computing. *IEEE Access*, 8:209191–209198, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.3038287. URL http://dx.doi.org/10.1109/ACCESS.2020.3038287.

[46] Ashkan Yousefpour, Shen Guo, Ashish Shenoy, Sayan Ghosh, Pierre Stock, Kiwan Maeng, Schalk-Willem Krüger, Michael Rabbat, Carole-Jean Wu, and Ilya Mironov. Green federated learning, 2023. URL https://arxiv.org/abs/2303.14604.

[47] Rong Yu and Peichun Li. Toward resource-efficient federated learning in mobile edge computing. *IEEE Network*, 35(1):148–155, 2021.

[48] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedGPT: Federated instruction tuning. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=TaDiklyVps.

[49] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9963–9977, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.632. URL https://aclanthology.org/2023.findings-acl.632.

[50] Yanli Zhao, Andrew Gu, et al. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16(12):3848–3860, aug 2023. ISSN 2150-8097. doi: 10.14778/3611540.3611569. URL https://doi.org/10.14778/3611540.3611569.

[51] Jingjing Zheng, Kai Li, Eduardo Tovar, and Mohsen Guizani. Federated learning for energy-balanced client selection in mobile edge computing. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1942–1947. IEEE, 2021.

APPENDIX

Our appendix is organized along the main paper structure. We provide additional details for our methodology, experimental setup, and experiment results.

Appendix A METHODOLOGY

Communication efficiency. We use to per-bit communication model proposed by Jalali et al. [23] to estimate the total communication cost of our FL experiments. In the following, we provide a detailed explanation of calculating the cost for an FL experiment with the per-bit communication cost model.

$$P_{\mathrm{t}} = E_t \cdot \mathcal{B} = (n_{\mathrm{as}} \cdot E_{\mathrm{as}} + n_{\mathrm{LTEE}} \cdot E_{\mathrm{LTEE}} + n_{\mathrm{LTEB}} \cdot E_{\mathrm{LTEB}} + E_{\mathrm{bng}} + n_e \cdot E_e + n_c \cdot E_c + n_d \cdot E_d) \cdot \mathcal{B}. \tag{3}$$

P_{t} is the total power draw for a single transmission. E_{as}, E_{LTEE}, E_{LTEB}, E_{bng}, E_e, E_c, E_d denote the per-bit energy consumption of edge one or more ethernet switches n_{as}, one or 0 client-side LTE endpoint n_{LTEE}, one or 0 LTE base stations n_{LTEB}, the broadband network gateway (BNG), one or more edge routers n_e, one or more core routers n_c, and one or more data center Ethernet switches n_d, respectively. We adopt the energy consumption of networking devices as specified in Jalali et al. [23] and assume $n_{\mathrm{as}} = 1$, $n_{\mathrm{bng}} = 1$, $n_{\mathrm{e}} = 3$. $n_{\mathrm{c}} = 4$, and $n_{\mathrm{d}} = 2$. For calculation involving wireless communication, we assume $n_{\mathrm{LTEE}} = 1$ and $n_{\mathrm{LTEB}} = 1$, 0 otherwise. $\mathbb{B}$ is the number of trainable model parameters multiplied by the parameter precision (32 bits in our case).

Table 2: Energy efficiency is measured in $\frac{\mathrm{TPS}}{W}$.

FLAN-T5 Model	Minib. Size	Device	Avg. Power Draw (W)	η_e	TPS
Small	1	A100	75.8	21.16	1603.62
		AGX Orin	16.7	55.26	923.04
	8	A100	103.17	121.63	12548.15
		AGX Orin	28.1	200.75	5641.15
	16	A100	120.96	215.03	26010.34
		AGX Orin	35.8	198.32	7099.35
	32	A100	171.15	258.52	44246.31
		AGX Orin	38.84	196.53	7633.56
	64	A100	212.53	304.82	64782.11
		AGX Orin	38.82	203.57	7903.25
	128	A100	247.72	327.2	81055.16
		AGX Orin	41.08	195.73	8040.93
Base	1	A100	85.63	13.0	1113.37
		AGX Orin	24.86	25.23	627.45
	8	A100	135.57	63.77	8645.35
		AGX Orin	31.3	74.21	2322.81
	16	A100	159.09	94.55	15041.23
		AGX Orin	38.59	66.51	2566.63
	32	A100	223.55	99.28	22194.39
		AGX Orin	38.54	69.84	2691.45
	64	A100	260.68	100.08	26088.77
		AGX Orin	Out of memory		
Large	1	A100	91.61	6.06	554.97
		AGX Orin	26.1	11.24	293.38
	8	A100	173.43	24.24	4204.11
		AGX Orin	41.46	20.36	844.18
	16	A100	196.9	33.76	6647.37
		AGX Orin	Out of memory		
XL	1	A100	128.21	4.31	552.0
		AGX Orin	Out of memory		

Appendix B EXPERIMENTAL SETUP

Evaluation hardware. We feel it is important to provide an estimate for establishing a research cluster with NVIDIA Jetson Orin 64 GB devices. We purchased the devices in mid-2023 at a unit price tag of roughly EUR 2,400, totaling EUR 24,000 just in compute. Additionally, we equipped each Orin with a Samsung 980 Pro 1 TB NVMe SSD, which cost us a total of EUR 700. The necessary networking infrastructure (FS S5860-48XMG-U + cables) with a 10 Gbit/s uplink for each device had a price tag of EUR 4,900. The enclosure is custom-made from sheet metal and aluminum to fit a standard 19-inch rack. The material for the case was around EUR 150 + 5 hours for assembly. In total, our embedded computing cluster cost us just shy of EUR 30,000. We are happy to share CAD designs and a full part list with anyone interested.

DL models. We use the pre-trained FLAN-T5 models provided by Google via the HuggingFace hub. For each model, we ran a hyperparameter search in a centralized experiment on a single node. In our experiments, we chose the optimal hyperparameter configuration as depicted in table 3. Our search space is as follows. The learning rate was selected from a set of values [0.1, 0.001, 0.0006, 0.0005, 0.0003, 0.0001, 0.00001]. Similarly, we selected the weight decay from the following set [0.1, 0.003, 0.001, 0.0009, 0.0001]. The momentum and β_1 were selected from the set [0.85, 0.9, 0.95]. β_2 was selected from the set [0.99, 0.995, 0.999, 0.9999].

Dataset. We randomly sample the Samsum dataset by using a Dirichlet distribution ($\alpha = 1.0$) for 100 clients in our experiments. The number of samples per client is depicted in Figure 6.

FL setup In Table 3, we provide the full set of hyperparameters used in our experiments. For our main paper, we limit the number of samples each model sees to 60, 000. In addition, we performed additional experiments to identify the point of overfitting for each FL optimizer that would allow a model to converge. We chose to validate the global model every 200 FL rounds.

Appendix C RESULTS

This appendix section contains additional results on the energy efficiency measurements and micro-benchmark timings.

C.1 Energy efficiency

Energy efficiency is derived from the average power draw of each device during our experiments. The experiments are fixed to 100 steps per epoch for each experiment. Table 2 contains details on our energy efficiency calculations.

C.2 Model FLOP Utilization

MFU helps to identify computational or memory bottlenecks. Table 5 depicts all details required to calculate the MFU.

C.3 Micro-benchmark

Table 4 describes the step timings in detail and provides a perspective on the speed differences between the data center and embedded hardware.

Model	Small				Base				Large			
Optimizer	FedAvg	FedAvgM	FedAdam	FedAdamW	FedAvg	FedAvgM	FedAdam	FedAdamW	FedAvg	FedAvgM	FedAdam	FedAdamW
Mini-Batch Size	30				20				10			
Learning Rate	0.01	0.1	0.0005	0.0005	0.001	0.1	0.0005	0.0005	0.01	0.1	0.0005	0.0005
Weight Decay	0.001	0.001	–	0.001	0.001	0.001	–	0.001	0.001	0.001	–	0.001
Momentum	0.0	0.9	–	–	0.0	0.9	–	–	0.0	0.9	–	–
β_1	–	–	0.9	0.9	–	–	0.9	0.9	–	–	0.9	0.9
β_2	–	–	0.999	0.999	–	–	0.999	0.999	–	–	0.999	0.999
Training rounds	1000				1500				1500			
Clients p. Round	10											

Table 3: Hyperparameter settings for all FLAN-T5 models and the corresponding optimizers.

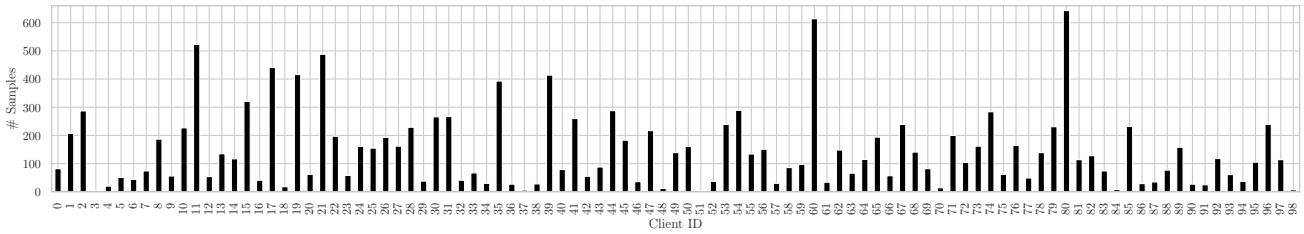


Figure 6: Dataset samples per client

Table 5: Details on MFU calculation for the NVIDIA A100 and Jetson AGX Orin platforms.

FLAN-T5 Model	Minib. Size	Device	TPS	Params	# Layers	d_{model}	d_{ff}	$n_{attheads}$	Seq. Len.	MFU
Small	1	A100	1657.0	0.0	8.0	512.0	1024.0	8.0	512.0	0.3
		Orin AGX	927.0	0.0	8.0	512.0	1024.0	8.0	512.0	1.1
	8	A100	13051.0	0.0	8.0	512.0	1024.0	8.0	512.0	2.0
		Orin AGX	5665.0	0.0	8.0	512.0	1024.0	8.0	512.0	6.5
	16	A100	26741.0	0.0	8.0	512.0	1024.0	8.0	512.0	4.2
		Orin AGX	7112.0	0.0	8.0	512.0	1024.0	8.0	512.0	8.1
	32	A100	45428.0	0.0	8.0	512.0	1024.0	8.0	512.0	7.1
		Orin AGX	7713.0	0.0	8.0	512.0	1024.0	8.0	512.0	8.8
	64	A100	65944.0	0.0	8.0	512.0	1024.0	8.0	512.0	10.3
		Orin AGX	8040.0	0.0	8.0	512.0	1024.0	8.0	512.0	9.2
	128	A100	82045.0	0.0	8.0	512.0	1024.0	8.0	512.0	12.8
		Orin AGX	8094.0	0.0	8.0	512.0	1024.0	8.0	512.0	9.3
Base	1	A100	1134.0	0.0	12.0	768.0	2048.0	12.0	512.0	0.6
		Orin AGX	631.0	0.0	12.0	768.0	2048.0	12.0	512.0	2.3
	8	A100	8805.0	0.0	12.0	768.0	2048.0	12.0	512.0	4.4
		Orin AGX	2339.0	0.0	12.0	768.0	2048.0	12.0	512.0	8.5
	16	A100	15380.0	0.0	12.0	768.0	2048.0	12.0	512.0	7.6
		Orin AGX	2591.0	0.0	12.0	768.0	2048.0	12.0	512.0	9.4
	32	A100	22427.0	0.0	12.0	768.0	2048.0	12.0	512.0	11.1
		Orin AGX	2692.0	0.0	12.0	768.0	2048.0	12.0	512.0	9.8
	64	A100	26250.0	0.0	12.0	768.0	2048.0	12.0	512.0	13.0
		AGX Orin	Out of memory							
Large	1	A100	562.0	0.0	24.0	1024.0	2816.0	16.0	512.0	0.9
		Orin AGX	298.0	0.0	24.0	1024.0	2816.0	16.0	512.0	3.4
	8	A100	4260.0	0.0	24.0	1024.0	2816.0	16.0	512.0	6.6
		Orin AGX	853.0	0.0	24.0	1024.0	2816.0	16.0	512.0	9.7
	16	A100	6728.0	0.0	24.0	1024.0	2816.0	16.0	512.0	10.5
		AGX Orin	Out of memory							
XL	1	A100	560.0	0.0	24.0	2048.0	5120.0	32.0	512.0	3.1
		AGX Orin	Out of memory							

Table 4: Results of the micro-benchmark of the FLAN-T5 transformer model family on the NVIDIA A100 and Jetson AGX Orin platforms. The sequence length per batch item is 512.

FLAN-T5 Model	Batch Size	Device	Backward	Opt. Step	Loss Calc.	Forward	Batch Loading	TPS	Total Time
Small	1	A100	0.09	0.16	0.0	0.06	0.01	1657.87	0.32
		AGX Orin	0.15	0.3	0.0	0.09	0.01	927.51	0.55
	8	A100	0.09	0.16	0.0	0.06	0.01	13051.3	0.33
		AGX Orin	0.22	0.4	0.0	0.1	0.01	5665.54	0.73
	16	A100	0.09	0.16	0.0	0.06	0.01	26741.79	0.31
		AGX Orin	0.36	0.63	0.0	0.15	0.01	7112.45	1.15
	32	A100	0.12	0.18	0.0	0.06	0.01	45428.27	0.37
		AGX Orin	0.66	1.16	0.0	0.32	0.01	7713.02	2.15
	64	A100	0.17	0.26	0.0	0.06	0.01	65944.32	0.51
		AGX Orin	1.28	2.22	0.0	0.64	0.01	7992.08	4.15
	128	A100	0.28	0.46	0.0	0.06	0.02	82045.0	0.81
		AGX Orin	2.6	4.34	0.0	1.21	0.01	8046.96	8.15
Base	1	A100	0.14	0.23	0.0	0.08	0.01	1134.51	0.46
		AGX Orin	0.22	0.45	0.0	0.14	0.01	631.08	0.82
	8	A100	0.14	0.23	0.0	0.09	0.01	8805.38	0.47
		AGX Orin	0.51	0.95	0.0	0.3	0.01	2339.49	1.76
	16	A100	0.17	0.27	0.0	0.09	0.02	15380.06	0.54
		AGX Orin	0.93	1.69	0.0	0.57	0.01	2590.44	3.19
	32	A100	0.24	0.38	0.0	0.1	0.01	22427.21	0.74
		AGX Orin	1.81	3.19	0.0	1.09	0.01	2692.26	6.09
	64	A100	0.4	0.65	0.0	0.19	0.01	26250.25	1.26
		AGX Orin	Out of memory						
Large	1	A100	0.27	0.46	0.0	0.18	0.02	562.2	0.92
		AGX Orin	0.43	1.03	0.0	0.27	0.01	298.91	1.75
	8	A100	0.29	0.49	0.0	0.18	0.02	4260.38	0.97
		AGX Orin	1.31	2.62	0.0	0.92	0.01	850.43	4.85
	16	A100	0.4	0.62	0.0	0.2	0.02	6728.79	1.23
		AGX Orin	Out of memory						
XL	1	A100	0.27	0.48	0.0	0.17	0.02	560.07	0.93
		AGX Orin	Out of memory						

ACM Publishing License and Audio/Video Release

Title of the Work: Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly
Submission ID:15600_10

Author/Presenter(s): Herbert Woisetschläger·Technical University of Munich;Alexander Erben·Technical University of Munich;Shiqiang Wang·IBM Research;Ruben Mayer·University of Bayreuth;Hans-Arno Jacobsen·University of Toronto

Type of material:full paper

Publication and/or Conference Name: DEEM 24: Workshop on Data Management for End-to-End Machine Learning Proceedings

1. Glossary

2. Grant of Rights

(a) Owner hereby grants to ACM an exclusive, worldwide, royalty-free, perpetual, irrevocable, transferable and sublicenseable license to publish, reproduce and distribute all or any part of the Work in any and all forms of media, now or hereafter known, including in the above publication and in the ACM Digital Library, and to authorize third parties to do the same.

(b) In connection with software and "Artistic Images and "Auxiliary Materials, Owner grants ACM non-exclusive permission to publish, reproduce and distribute in any and all forms of media, now or hereafter known, including in the above publication and in the ACM Digital Library.

(c) In connection with any "Minor Revision", that is, a derivative work containing less than twenty-five percent (25%) of new substantive material, Owner hereby grants to ACM all rights in the Minor Revision that Owner grants to ACM with respect to the Work, and all terms of this Agreement shall apply to the Minor Revision.
(d) If your paper is withdrawn before it is published in the ACM Digital Library, the rights revert back to the author(s).

☑ A. Grant of Rights. I grant the rights and agree to the terms described above.

☐ B. Declaration for Government Work. I am an employee of the national government of my country/region and my Government claims rights to this work, or it is not copyrightable (Government work is classified as Public Domain in U.S. only)

Are you a contractor of your National Government? ○ Yes ◉ No

Are any of the co-authors, employees or contractors of a National Government? ○ Yes ◉ No

---

3. Reserved Rights and Permitted Uses.

(a) All rights and permissions the author has not granted to ACM in Paragraph 2 are reserved to the Owner, including without limitation the ownership of the copyright of the Work and all other proprietary rights such as patent or trademark rights.

(b) Furthermore, notwithstanding the exclusive rights the Owner has granted to ACM in Paragraph 2(a), Owner shall have the right to do the following:

(i) Reuse any portion of the Work, without fee, in any future works written or edited by the Author, including books, lectures and presentations in any and all media.

(ii) Create a "Major Revision" which is wholly owned by the author

(iii) Post the Accepted Version of the Work on (1) the Author's home page, (2) the Owner's institutional repository, (3) any repository legally mandated by an agency funding the research on which the Work is based, and (4) any non-commercial repository or aggregation that does not duplicate ACM tables of contents, i.e., whose patterns of links do not substantially duplicate an ACM-copyrighted volume or issue. Non-commercial repositories are here understood as repositories owned by non-profit organizations that do not charge a fee for accessing deposited articles and that do not sell advertising or otherwise profit from serving articles.

(iv) Post an "Author-Izer" link enabling free downloads of the Version of Record in the ACM Digital Library on (1) the Author's home page or (2) the Owner's institutional repository;

(v) Prior to commencement of the ACM peer review process, post the version of the Work as submitted to ACM ("Submitted Version" or any earlier versions) to non-peer reviewed servers;

(vi) Make free distributions of the final published Version of Record internally to the Owner's employees, if applicable;

(vii) Make free distributions of the published Version of Record for Classroom and Personal Use;

(viii) Bundle the Work in any of Owner's software distributions; and

(ix) Use any Auxiliary Material independent from the Work.

When preparing your paper for submission using the ACM TeX templates, the rights and permissions information and the bibliographic strip must appear on the lower left hand portion of the first page.

The new ACM Consolidated TeX template Version 1.3 and above automatically creates and positions these text blocks for you based on the code snippet which is system-generated based on your rights management choice and this particular conference. When creating your document, please make sure that you are only using TAPS accepted packages. (If you would like to use a package not on the list, please send suggestions to acmtexsupport@aptaracorp.com RE: TAPS LaTeX Package

evaluation.)

NOTE: For authors using the ACM Microsoft Word Master Article Template and Publication Workflow, The ACM Publishing System (TAPS) will add the rights statement to your papers for you. Please check with your conference contact for information regarding submitting your source file(s) for processing.

Please put the following LaTeX commands in the preamble of your document - i.e., before \begin{document}:

\copyrightyear{2024}
\acmYear{2024}
\setcopyright{acmlicensed}\acmConference[DEEM 24]{Workshop on Data Management for End-to-End Machine Learning}{June 9, 2024}{Santiago, AA, Chile}
\acmBooktitle{Workshop on Data Management for End-to-End Machine Learning (DEEM 24), June 9, 2024, Santiago, AA, Chile}
\acmDOI{10.1145/3650203.3663331}
\acmISBN{979-8-4007-0611-0/24/06}

NOTE: For authors using the ACM Microsoft Word Master Article Template and Publication Workflow, The ACM Publishing System (TAPS) will add the rights statement to your papers for you. Please check with your conference contact for information regarding submitting your source file(s) for processing.

If you are using the ACM Interim Microsoft Word template, or still using or older versions of the ACM SIGCHI template, you must copy and paste the following text block into your document as per the instructions provided with the templates you are using:

DEEM 24, June 9, 2024, Santiago, AA, Chile

ACM 979-8-4007-0611-0/24/06...
https://doi.org/10.1145/3650203.3663331

NOTE: Make sure to include your article's DOI as part of the bibstrip data; DOIs will be registered and become active shortly after publication in the ACM Digital Library.

Once you have your camera ready copy ready, please send your source files and PDF to your event contact for processing.

---

4. ACM Citation and Digital Object Identifier.

(a) In connection with any use by the Owner of the Definitive Version, Owner shall include the ACM citation and ACM Digital Object Identifier (DOI).
(b) In connection with any use by the Owner of the Submitted Version (if accepted) or the Accepted Version or a Minor Revision, Owner shall use best efforts to display the ACM citation, along with a statement substantially similar to the following:

> "© [Owner] [Year]. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive version was published in {Source Publication}, https://doi.org/10.1145/{number}."

5. Livestreaming and Distribution

You are giving a presentation at the annual conference. This section of the rights form gives you the opportunity to grant or deny ACM the ability to make this presentation more widely seen, through (a) livestreaming of the presentation during the conference and/or (b) distributing the presentation after the conference in the ACM Digital Library, the "Conference Presentations" USB, and media outlets such as Vimeo and YouTube. It also provides you the opportunity to grant or deny our use of the presentation in promotional and marketing efforts after the conference.

Not all conference presentations are livestreamed; you will be notified in advance of the possibility of your presentation being livestreamed.

The permissions granted and/or denied here apply to all presentations of this material at the conference, including (but not limited to) the primary presentation and any program-specific "fast forward" presentations.

ACM's policy on the use of third-party material applies to your presentation as well as the documentation of your work; if you are using others' material in your presentation, including audio, you must identify that material on the ACM rights form and in the presentation where it is used, and secure permission to use the material where necessary.

Livestreaming.
I grant permission to ACM to livestream my presentation during the conference (a "livestream" is a synchronous distribution of the presentation to the public, separate from the presentation distributed to conference registrants).
(●) Yes
( ) No

Post-Conference Distribution.
I grant permission to ACM to distribute the recording of my presentation after the

conference as listed above.
(●) Yes
( ) No

6. Auxiliary Material

Do you have any Auxiliary Materials? ( ) Yes (●) No

7. Third Party Materials

In the event that any materials used in my presentation or Auxiliary Materials contain the work of third-party individuals or organizations (including copyrighted music or movie excerpts or anything not owned by me), I understand that it is my responsibility to secure any necessary permissions and/or licenses for print and/or digital publication, and cite or attach them below.

(●) We/I have not used third-party material.
( ) We/I have used third-party materials and have necessary permissions.

8. Artistic Images

If your paper includes images that were created for any purpose other than this paper and to which you or your employer claim copyright, you must complete Part IV and be sure to include a notice of copyright with each such image in the paper.
(●) We/I do not have any artistic images.
( ) We/I have any artistic images.

---

9. Representations, Warranties and Covenants

The undersigned hereby represents, warrants and covenants as follows:

(a) Owner is the sole owner or authorized agent of Owner(s) of the Work;

(b) The undersigned is authorized to enter into this Agreement and grant the rights included in this license to ACM;

(c) The Work is original and does not infringe the rights of any third party; all permissions for use of third-party materials consistent in scope and duration with the rights granted to ACM have been obtained, copies of such permissions have been provided to ACM, and the Work as submitted to ACM clearly and accurately indicates the credit to the proprietors of any such third-party materials (including any applicable copyright notice), or will be revised to indicate such credit;

(d) The Work has not been published except for informal postings on non-peer reviewed servers, and Owner covenants to use best efforts to place ACM DOI pointers on any such prior postings;

(e) The Auxiliary Materials, if any, contain no malicious code, virus, trojan horse or other software routines or hardware components designed to permit unauthorized access or to disable, erase or otherwise harm any computer systems or software; and

(f) The Artistic Images, if any, are clearly and accurately noted as such (including any applicable copyright notice) in the Submitted Version.

☑ I agree to the Representations, Warranties and Covenants.

---

10. Enforcement.

At ACM's expense, ACM shall have the right (but not the obligation) to defend and enforce the rights granted to ACM hereunder, including in connection with any instances of plagiarism brought to the attention of ACM. Owner shall notify ACM in writing as promptly as practicable upon becoming aware that any third party is infringing upon the rights granted to ACM, and shall reasonably cooperate with ACM in its defense or enforcement.

---

11. Governing Law

This Agreement shall be governed by, and construed in accordance with, the laws of the state of New York applicable to contracts entered into and to be fully performed therein.

Funding Agents

1. Bavarian Ministry of Economic Affairs, Regional Development and Energy award number(s):DIK0446/01

2. German Federal Ministry for Economic Affairs and Climate Action award number(s):16KN085729

3. German Research Foundation award number(s):392214008

DATE: **05/02/2024** sent to herbert.woisetschlaeger@tum.de at **04:05:06**

Appendix C

A Survey on Efficient Federated Learning Methods for Foundation Model Training

Printed with the permission of

Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. "A survey on efficient federated learning methods for foundation model training." In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. IJCAI '24. Jeju, Korea, 2024, pp. 8317–8325. ISBN: 978-1-956792-04-1. DOI: 10.24963/ijcai.2024/919. URL: https://doi.org/10.24963/ijcai.2024/919

Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Survey Track

A Survey on Efficient Federated Learning Methods for Foundation Model Training

Herbert Woisetschläger[1] , **Alexander Erben**[1] , **Shiqiang Wang**[2] ,
Ruben Mayer[3] and **Hans-Arno Jacobsen**[4]
[1]Technical University of Munich
[2]IBM Research
[3]University of Bayreuth
[4]University of Toronto
{herbert.woisetschlaeger, alex.erben}@tum.de, wangshiq@us.ibm.com, ruben.mayer@uni-bayreuth.de,
jacobsen@eecg.toronto.edu

Abstract

Federated Learning (FL) has become an established technique to facilitate privacy-preserving collaborative training across a multitude of clients. However, new approaches to FL often discuss their contributions involving small deep-learning models only and focus on training full models on clients. In the wake of Foundation Models (FM), the reality is different for many deep learning applications. Typically, FMs have already been pre-trained across a wide variety of tasks and can be fine-tuned to specific downstream tasks over significantly smaller datasets than required for full model training. However, access to such datasets is often challenging. By its design, FL can help to open data silos. With this survey, we introduce a novel taxonomy focused on computational and communication efficiency, the vital elements to make use of FMs in FL systems. We discuss the benefits and drawbacks of parameter-efficient fine-tuning (PEFT) for FL applications, elaborate on the readiness of FL frameworks to work with FMs and provide future research opportunities on how to evaluate generative models in FL as well as the interplay of privacy and PEFT.

1 Introduction

Foundation Models (FMs) [Bommasani *et al.*, 2021] have conquered the deep learning world with unprecedented speed, enabling generative artificial intelligence for a broad audience. As FMs have been pre-trained on an extensive data basis and can be used in multi-modal applications, they perform well over a wide range of tasks. To specialize these models on a downstream task, we use fine-tuning that can either be done over the full model or with parameter-efficient fine-tuning techniques (PEFT) [Hu *et al.*, 2021; Lester *et al.*, 2021; Zaken *et al.*, 2021]. A major advantage is that fine-tuning requires orders of magnitude smaller datasets than pre-training but benefits from access to a variety of samples pertaining to a task.

Access to a breadth of data has always been challenging in deep learning, as data owners are typically reluctant to share their data with service providers. To tackle the data access challenge, McMahan *et al.* [2017] introduced *Federated Learning* (FL). FL enables privacy-preserving machine learning over decentralized data without the necessity of sharing input data and does not require high bandwidth client connections. Rather, a set of clients collectively train a model and send their local model updates to a server that subsequently aggregates the updates to a global model. In FL applications that involve small models with less than 1 million parameters, we may spend as much time on communication as we spend on computation [Yousefpour *et al.*, 2023]. However, it is desirable to design systems in such a way that computation takes up the majority of time. Luckily, the larger the models become, the time spent on training is larger than on communication [Ryabinin *et al.*, 2023; Woisetschläger *et al.*, 2023; Beutel *et al.*, 2020]. As such, scaling the model size in FL systems can be desirable, and this introduces beneficial properties that can aid us in training large models with several 100 million parameters and beyond. These properties render FL the perfect choice for fine-tuning FMs for downstream tasks. FL can provide access to a significantly larger and more diverse data basis while benefiting from the increased time spent on the computation of FMs over small models. Yet, the costs of transmitting model updates remain significant even with the increased computational load of FMs [Yousefpour *et al.*, 2023], making it a priority to jointly optimize the training and communication efficiency.

Our survey is the first to study advances in computational and communication-efficient methods for *FM training* in FL applications. Our work contains three distinct contributions:

- **A novel taxonomy on FL methods for FM training focused on the core challenges in computation and communication**. We discover a gap between FL methods to increase computational efficiency and techniques to improve communication efficiency. While we see research on computational efficiency for FM training and fine-tuning in FL applications, communication efficiency methods are predominantly tailored to full-model training. Our taxonomy aims to identify synergies be-

tween FL methods for FMs and efficient communication methods.

- **Holistic evaluation of existing FL computational efficiency methods for FMs and communication efficiency techniques**. We study how existing techniques can help drive the adoption of FMs in FL applications and what needs to be done to render FL frameworks ready for large models.
- **We thoroughly discuss future research directions**. We highlight future directions for research on computational and communication efficiency as these domains grow closer together. Also, we show what is necessary to make FMs in FL applications a reality, especially with regard to generative tasks and privacy considerations.

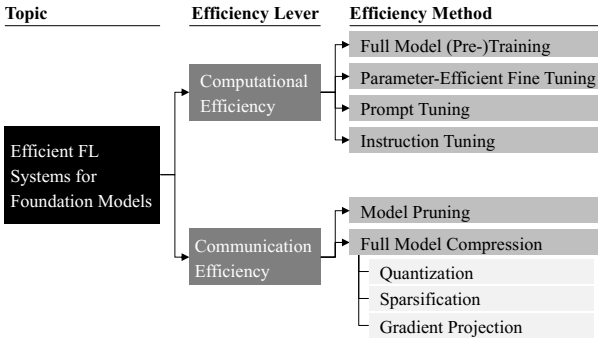


Figure 1: Our Taxonomy. Foundation Models, in conjunction with Federated Learning, require efficient computational and communication methods.

2 Taxonomy

Our taxonomy introduces a novel perspective focused on the current developments in the field of efficient computational and communication methods for FL applications intended for training and fine-tuning FMs. While communication efficiency has been studied extensively in FL, computational advancements in conjunction with large models are currently emerging. Our taxonomy is visualized in Figure 1.

2.1 Basics of Federated Learning

To understand the relevancy of our taxonomy, it is key to briefly introduce the fundamentals of FL and the notations of this paper.

Usually coordinated by a server, the overall goal of FL is to collaboratively and iteratively train a DL model across a set of clients N and minimize the global loss, where f denotes an objective function with parameters w,

$$\min_{w} f(w) := \frac{1}{|N|} \sum_{n=1}^{|N|} f_n(w). \tag{1}$$

With this, we create a model that generalizes across all clients $n \in N$. Specifically, at the beginning, we commonly initialize the model weights across clients. Then, clients train the model on their local data and return the updated model parameters to the server. At the same time, each client updates its local parameters with fixed minibatch size m over one or more epochs by applying gradient descent steps $\nabla l(w_t^n; m)$ to the model,

$$w_{t+1}^n = w_t - \eta \nabla l(w_t^n; m), \forall n \in N. \tag{2}$$

Subsequently, only the model parameters w_{t+1}^n are communicated back to the server.

2.2 Taxonomy Explanation

Our taxonomy is centered on two major challenges in FL: the computational and communication efficiency levers. While these challenges have been studied in FL extensively in the past [Zhang *et al.*, 2021], existing approaches predominantly focus on small models with $< 10\text{M}$ parameters. With the emergence of FMs as the backbone for multi-modal applications, we need to combine computational and communication efficiency in FL as these FMs typically come with > 1 billion parameters [Zhang *et al.*, 2023a]; a growth factor of $100\times$ This introduces additional computational load for FL clients, while their resources are often scarce already [Beutel *et al.*, 2020]. At the same time, communication loads are also growing since many parameters need to be transmitted.

Computational Efficiency. We discuss computational efficiency levers along four major categories. Full model training is used to train large transformer models from the very beginning (Section 3.1). Parameter-efficient fine-tuning techniques can be utilized to improve a pre-trained FM for a specific downstream task (Section 3.2). Prompt tuning enables performance improvements of an FM without training the model itself but by designing textual prompts that we prepend to an input (Section 3.3). Instruction tuning enables fine-grained control over the model training process and allows for a high degree of model specialization for certain downstream tasks (Section 3.4).

Communication Efficiency. While computationally efficient methods for FM training in FL applications can already reduce the number of parameters to communicate, there is still a significant amount of data to be communicated between clients and servers. For instance, parameter-efficient fine-tuning methods only require 1–2% of parameters to be trainable. This still amounts to up to 14M parameters when working with Alpaca-7B [Zhang *et al.*, 2023a], larger than the majority of models (with $< 1\text{M}$ trainable parameters) currently being discussed in FL research [He *et al.*, 2020; Beutel *et al.*, 2020].

We discuss communication efficiency methods along two major categories. (I) *Model pruning* is a method to communicate parts of a model between clients and the server, which resemble the most important parameters for a client (Section 4.1). (II) *Full model compression* is divided into three sub-categories: First, quantization is a method to decrease the numeric precision of model parameters. Second, sparsification is used as a way to zero out less important model parameters. Third, gradient projection transforms high dimensional parameter matrices in client updates into scalar vectors for communication (Section 4.2).

Paper	Training Regime	Underlying FL Aggregation Strategy	Max. Model Params	Trainable Parameters	Comms Savings compared to FMT	Domain CV	Domain NLP	Domain Audio
Centralized Learning	FMT, PEFT, PT, IT	–	$\geq$ 7B	–	–	✓	✓	✓
FedBERT [Tian *et al.*, 2022]	FMT	Weighted Average	117M	110M	0%		✓	
FedCLIP [Lu *et al.*, 2023]	PEFT	Weighted Average	85M	530K	$> 99\%$	✓		
FedPEFT [Sun *et al.*, 2023]	PEFT	Weighted Average	85M	230K	$> 99\%$	✓		
FedPETuning [Zhang *et al.*, 2023b]	PEFT	Weighted Average	125M	1.25M	99%		✓	
SLoRA [Babakniya *et al.*, 2023]	PEFT	Weighted Average	67M	140K	$> 99\%$		✓	
FedDPA [Yang *et al.*, 2024]	PEFT	Weighted Average	7B	N/A	$> 99\%$		✓	
FedPrompt [Zhao *et al.*, 2022]	PT	Weighted Average	223M	–	$> 99\%$		✓	
FedIT [Zhang *et al.*, 2023a]	IT	Weighted Average	7B	–	$> 99\%$		✓	

Table 1: Computational Efficiency Methods for FL Systems and FM. FMs are generally multi-modal and provide a strong performance across a variety of domains that are well explored in centralized learning but not in FL [Dosovitskiy *et al.*, 2020; OpenAI, 2023; Yang *et al.*, 2023].

3 Computational Efficiency

This section discusses recent methods to train FMs with FL methods. We distinguish between full model training (FMT), as it has been studied frequently in the FL domain, PEFT, prompt tuning (PT), and instruction tuning (IT). Table 1 summarizes existing computational efficiency methods for FM training.

3.1 Full Model Training

Generally, full model training is referred to when training all parameters of a neural network. In this approach, we locally train a model on all clients with the objective of minimizing the loss l.

The BERT model is one of the first models to use the transformer layer architecture – the building blocks of FMs – to achieve state-of-the-art performance at the time of its release. Tian *et al.* [2022] discuss federated pre-training of BERT-family models with up to 117M parameters by applying masked language modeling (MLM). In MLM, the loss is defined over the sum of probabilities P of predicted tokens $\hat{x}$ over the representation function of a masked sentence $g\left(\frac{x}{M(x)}\right)$,

$$l(w_n^t, m) = -\sum_{\hat{x} \in \mathcal{M}(\mathrm{x})} \log P\left(\hat{x} \middle| g\left(\frac{x}{M(x)}\right)\right), \forall n \in N. \tag{3}$$

In FL applications, supervised learning can be challenging as we cannot ensure a proper data labeling process for supervised learning because we usually cannot access client data. Here, MLM can be beneficial since it is a self-supervised learning technique that masks parts of a sentence. Those masked parts will be used as the prediction target to automatically create an input and target sample. Federated pre-training provides a net benefit over training on local datasets [Ding *et al.*, 2023; Babakniya *et al.*, 2023; He *et al.*, 2020]. However, the perplexity, an indicator for quality in large models, yields four orders of magnitude worse results for large models (e.g., GPT-2) trained in a federated fashion than centralized training, which is a stark indicator of low model quality [Tian *et al.*, 2022; Radford *et al.*, 2019]. Nonetheless, for use cases involving sensitive data and strict privacy regulation, full model pre-training allows the creation of foundation models based on federated data.

While FL pre-training has shown some promise, it is brittle and has shown to be worse at model sizes above 100M parameters [Tian *et al.*, 2022]. the applicability of model pre-training is currently limited due to the significantly lower model quality than in centralized training.

3.2 Parameter Efficient Fine-Tuning

Generally, PEFT is used to improve the performance of large models already trained on a large data corpus further and provide good performance across various tasks. This is especially effective since the data required for fine-tuning is orders of magnitude smaller than for pre-training. When applying PEFT, additional fully connected layers are inserted into the pre-trained model between the transformer blocks. While the original model weights are frozen, only the newly added layers are trained, typically resulting in $\geq$ 98% less communication [Hu *et al.*, 2021]. This renders PEFT techniques well-suited for FL applications since they address computation and communication alike. However, Babakniya *et al.* [2023] show in their study that PEFT is more sensitive towards non-IID data than FMT, but this sensitivity can be mitigated. PEFT can be applied in FL applications as follows.

Sparse fine-tuning of pre-trained model parameters. As communication is a key concern in FL, reducing the number of parameters to communicate between client and server has become a priority. One of the most used approaches to achieve this is BitFit [Zaken *et al.*, 2021]. The technique freezes almost all model parameters w_t and only trains the bias term b and the final classification layer w_t^{final} over the input features a_t, where the next-layer input features $a_{t+1} = a_t \cdot w_t + b$. With this technique, it is only required to communicate b and w_t^{final}. The communication load is reduced by $\geq$ 99%, i.e., instead of communicating 100 million parameters, only 100 thousand parameters are sent over the network.

Sun *et al.* [2023] propose FedPEFT, a framework for federated transformer fine-tuning that freezes the model weights to retain upstream knowledge within the model and adjust the systematic error for the downstream task. Their experimental results on vision transformers (ViT-B with 85M parameters) show on-par performance compared to full model fine-tuning on non-IID data, all the while reducing communication by 99.8%.

Adapter-based fine-tuning of additionally added parameters. When aiming to maintain a pre-trained model while introducing task-specific knowledge, adapter-based fine-tuning techniques provide strong performance and on-par efficiency compared to sparse fine-tuning techniques [Houlsby

et al., 2019]. Here are two adapter layers introduced in each transformer block of a foundation model. The adapter layer a^a_{t+1} is calculated based on a downstream projection w^{down}_t of input feature a^a_t into a lower-dimensional space $w^{\text{down}}_t \in \mathcal{R}^{d\times r}$ followed by an upstream projection $w^{\text{up}}_t \in \mathcal{R}^{r\times u}$, resulting in

$$a^a_{t+1} = w^{\text{up}}_t \cdot h(w^{\text{down}}_t \cdot a^a_t). \tag{4}$$

With this, we can fine-tune an FM over much fewer dimensions than when fully fine-tuning a model. This saves both computational resources and communication costs in the same range as FedPEFT.

An improvement with regard to computational and communication efficiency over additional adapters is low-rank adapters (LoRA) [Hu *et al.*, 2021]. The technique uses a lower dimensional representation $\mathbf{A} \in \mathcal{R}^{r\times u}$, where u is the dimension of the next layer after the LoRA adapter and $\mathbf{B} \in \mathcal{R}^{d\times r}$, where d is the dimension of the previous LoRA adapter. The weight updates are calculated with $w_{t+1} = w_t + \Delta w = w_t + \mathbf{BA}$. $r \ll \min(d, u)$ casts the weight update matrices into a much lower dimensionality than in the original transformer module without the necessity of adding additional adapters, i.e., LoRA builds an adapter for existing parameters. However, as $\mathbf{A}$ is initialized randomly to a Gaussian distribution and $\mathbf{B}$ as a zero matrix, this works well for centralized settings with IID data [Hu *et al.*, 2021]. For FL settings, this initialization method bears the risk of slowing down the fine-tuning process over non-IID data.

FedCLIP [Lu *et al.*, 2023] introduces a PEFT method with an adjusted FedAvg-based adapter aggregation technique. Their approach yields significant performance improvements over vanilla FedAvg and FedProx.

Zhang *et al.* [2023b] provide a systematic benchmark study on adapter-based fine-tuning methods in privacy-preserving FL systems. Their results show that fine-tuning with additional adapters and LoRA both yield the same benchmark results regarding model accuracy. However, LoRA requires 66% less communication than additional adapters.

However, both FedCLIP and FedPETuning yield a worse accuracy than full fine-tuning. SLoRA [Babakniya *et al.*, 2023] and FedDPA [Yang *et al.*, 2024] are LoRA-based techniques to fine-tune models in non-IID FL settings. Their approach parameterizes the weight update based on r,

$$w_{t+1} = w_t + \frac{\beta}{r}\mathbf{BA}. \tag{5}$$

As β depends on r, scaling the ratio helps control the weight update impact of a single client. Subsequently, this can be used to control inconsistent training updates caused by non-IID data. To practically achieve this, Babakniya *et al.* [2023] make use of a two-stage process: First, they used singular vector decomposition on $\mathbf{A}$ and $\mathbf{B}$ to obtain a common initialization point for LoRA across all clients in an FL system. Second, the training is facilitated with the commonly initialized low-rank representations. Their approach achieves on-par performance with full model fine-tuning. However, they require a warmup time of approximately 100 FL rounds for stage 1, which can be very expensive in FL settings as clients are often unavailable consecutively for such a large number of FL rounds [McMahan *et al.*, 2017].

3.3 Prompt Tuning

Prompt tuning is another efficient method for tuning pre-trained models to a downstream task. Here, we use binary sentiments subsequent to masked-language-modeling to achieve high-quality results [Lester *et al.*, 2021]. As such, the model remains entirely frozen, and we only tweak the prompts (a very small number of tokens) that are being prepended to each embedded input query to improve the output quality. In contrast to fine-tuning, this method does not interfere with the model architecture or parameters. Lester *et al.* [2021] show that the effects of prompt tuning on model performance in centralized settings become better with larger models, i.e., for large FMs, prompt tuning bears significant potential.

Specifically, in FL settings, for each client n the likelihood P for a desired output $\hat{x}$ is calculated over prepending trainable embedded prompts x_p to each embedded input x. In the interactive training process, the prompt is optimized in such a way that it optimally resembles the local objective of a client,

$$\max(P_n(\hat{x}|[x_p; x])), \forall n \in N. \tag{6}$$

Zhao *et al.* [2022] introduce FedPrompt, a method to efficiently communicate federally generated prompts only and aggregate them such that the global model performance of a pre-trained model improves for a downstream task. Their experimental evaluation shows a general sensitivity of prompt tuning towards data heterogeneity as the model performance degrades by 5 – 10% for the 100M parameter BERT model compared to a centrally trained baseline. However, with RoBERTa Base (124M parameters), the sensitivity diminishes, and the FL results are on par with centralized training. The larger T5 Base model (223M parameters) follows this trend, showing that prompt tuning becomes more effective with larger model sizes [Lester *et al.*, 2021].

3.4 Instruction Tuning

Some applications work with highly sensitive and protected data or require a very high model performance. The previously mentioned fine-tuning techniques may not yield sufficient results in these cases.

This is where instruction tuning comes into play as a technique that uses high-quality data. For instance, GPT-2 [Radford *et al.*, 2019] uses Reinforcement Learning with Human Feedback (RLHF). RLHF is a multi-stage process where an FM is initially trained on supervised data. In the second step - reward model training - the FM generates outputs over a given prompt, which a user then ranks based on their preference. With this, the model learns human preferences. In the third step - proximal policy optimization - the model trains self-supervised for a maximum reward [Zheng *et al.*, 2023].

With FedIT, Zhang *et al.* [2023a] study instruction tuning on LLaMA-7B in an FL application over heterogeneous client tasks, e.g., learning brainstorming and text summarization on different clients in a single system at the same time. Their results show that the additional context on a downstream task generated with federated instruction tuning provides net benefits over central training only, even in heterogeneous settings. However, these results were only produced over a single dataset, Dollybricks-15k.

3.5 Discussion

As more open-source FMs become available for unrestricted use (e.g., Alpaca [Taori *et al.*, 2023], Falcon [Penedo *et al.*, 2023]), it is unlikely that full model training will be a common use case since training an FM from scratch is very challenging, even in a centralized setting.

Therefore, we see a priority in improving upon PEFT for downstream tasks, for instance, by introducing new and enhancing existing algorithms to remove the currently required warm-up times to improve the performance of LoRA in non-IID data environments. Computer Vision (CV) applications may benefit from exploring prompt tuning for vision transformers [Jia *et al.*, 2022]. Also, we find significant challenges for instruction tuning as data quality is a general issue in FL [Longpre *et al.*, 2023], and access to human preferences, as it is required for RLHF, is hardly available in a real-world federated setting without incentive schemes.

Furthermore, PEFT also positively affects communication efficiency by significantly reducing the number of trainable (and thus communicable) parameters. As such, we now see computational and communication efficiency growing closer for FL and FMs, but not to a sufficient degree. As such, there is still a stark need to develop new approaches to use PEFT to improve computational and communication efficiency.

4 Communication Efficiency

With FMs, models have exponentially grown from a few million to several billion parameters to be able to serve multimodal tasks [Bommasani *et al.*, 2021]. For FL, this specifically means that the communication load between clients and servers has grown significantly, even though with PEFT, there is not necessarily the need to communicate an entire model. However, when fine-tuning a billion-parameter FM with adapter-based methods, we still need to facilitate communication for millions of parameters. For cross-device scenarios involving more than 1,000 clients per training round, the data traffic can quickly overstrain a server's network capacity and potentially incur significant communication costs for data transfer from the edge to a cloud [Xu *et al.*, 2023; Erben *et al.*, 2023]. Therefore, efficient communication and training design is vital for future FL systems. We distinguish two major efficiency methods: model pruning and full model compression. A detailed overview of studies on efficient communication in FL is provided in Table 2.

4.1 Model Pruning

The objective of model pruning (MP) is to retain and communicate only parts of a DL model that are relevant to a certain task. The reduction of parameters with this technique reduces the communication effort [Zhu and Gupta, 2017]. However, the success of pruning highly depends on the underlying data. Pruning client models without coordination may deny convergence with heterogeneous non-IID data in FL systems. Jiang *et al.* [2022] introduce PruneFL, a two-stage procedure to realize model pruning in FL systems. The first stage is carried out on a powerful client to find a common initialization for the model and generate the importance-based pruning mask. This mask is then iteratively refined over multiple FL training rounds under the consideration of all clients. The experimental results with PruneFL show two remarkable results: (I) The models have a shorter time to accuracy over the same task with PruneFL than with FedAvg, which is attributable to the higher degree of model specialization. (II) Since the model size is reduced, one would expect additional effects on faster computation, but there is limited hardware support for sparse matrix multiplications in training as they are required in PruneFL. Therefore, PruneFL has no computational benefits to this point. However, this may change with new hardware, such as sparse Tensor cores that support PruneFL's dynamic pruning approach [Zhu *et al.*, 2019]. With FedTiny, Huang *et al.* [2022] present an approach that works identically to PruneFL, except for them swapping the common initialization procedure with using batch normalization values of clients to choose a common initialization. Furthermore, FjORD [Horváth *et al.*, 2021] and HeteroFL [Diao *et al.*, 2020] provide similar approaches to pruning.

Isik *et al.* [2022] choose a similar approach for pruning models based on the lottery ticket hypothesis, first introduced to FL by Li *et al.* [2020]. Instead of commonly initializing a pruned model for an FL system, they initialize a random binary mask based on a shared seed on each client. This reduces computational efforts in the ramp-up phase. After an FL training round, each client communicates their binary mask to the server, which creates a global model based on the weighted average of those binary masks. With this, an approximate weight estimate replaces the parameters on the global model. From client to server, FedPM achieves significant communication efficiencies. However, the full model still has to be communicated from the server to the clients, lowering the net benefit.

Model pruning has also been discussed extensively for FM fine-tuning outside of FL [Lagunas *et al.*, 2021; Sanh *et al.*, 2020]. As existing pruning approaches have shown strong benefits to delivering on-par model performance compared to fine-tuning the full model, this is a promising direction to combine federated PEFT with highly efficient pruning techniques to further enhance communication. Along with pruning, sparse tensor hardware can lower computational loads.

4.2 Full Model Compression

Model pruning is prone to omit segments of a DL model that may become relevant at a later stage. This originates from domain shifts [Peng *et al.*, 2020] and might require preserving the full model with all its parameters. For this, three frequently discussed techniques for full model compression in FL systems are Quantization, Sparsification, and Gradient Projection.

Quantization (Q). The first work towards dynamic quantization is FedPAQ [Reisizadeh *et al.*, 2020], which combines FedAvg with strong quantization guarantees, where Q represents the quantization term for a local model update,

$$w_{t+1} = w_t + \frac{1}{|N|} \sum_{n=1}^{|N|} Q(w_{t+1}^n - w_t). \tag{7}$$

Algorithm	Enhancement Method(s)	Underlying FL Aggregation Strategy	Max. Model Parameters	Communication Savings vs. FedAvg	Domain CV	Domain NLP	Domain Audio
Centralized Learning	MP, Q, S, GP	–	$\geq$ 7B	–	✓	✓	✓
FedKSeed [Qin *et al.*, 2023]	GP	Weighted Average	3B	$\geq 99\%$		✓	
FedOBD [Chen *et al.*, 2023]	Q	Weighted Average	17M	89%	✓	✓	
PruneFL [Jiang *et al.*, 2022]	MP	Weighted Average	132M	80%	✓		
FedPM [Isik *et al.*, 2022]	MP	Weighted Average	12M	98%	✓		
FedTiny [Huang *et al.*, 2022]	MP	Weighted Average	132M	97%	✓		
SoteriaFL [Li *et al.*, 2022]	S	FedSGD	0.05M	N/A	✓		
FjORD [Horváth *et al.*, 2021]	MP	Weighted Average	11M	98%	✓	✓	
FedPAQ [Reisizadeh *et al.*, 2020]	Q	FedSGD	0.2 M	90%	✓		
HeteroFL [Diao *et al.*, 2020]	MP	Weighted Average	11M	98%	✓		
LotteryFL [Li *et al.*, 2020]	MP	Weighted Average	138M	50%	✓		

Table 2: Communication-efficient FL methods. Their centralized learning pendants are often tied to specific domains: CV [Habib *et al.*, 2023], NLP [He *et al.*, 2021], and Audio [Perez *et al.*, 2020].

While DL models often operate on full precision (32-bit), this high degree of detail is not necessarily required [Zhou *et al.*, 2018]. FedPAQ leverages this to reduce the communication intensity of FL applications. Q calculates the optimal float precision of a model update to preserve all required information: $Q(w) = \|w\| \cdot \text{sign}(w) \cdot \xi(w, s)$, as proposed in QSGD by Alistarh *et al.* [2017]. ξ formulates a stochastic process to dynamically tune s, the level of precision. FedPAQ has a significantly lower time to accuracy than QSGD, which is attributable to dynamizing s. However, it must be noted that dynamic quantization only yields benefits for communication. Depending on the infrastructure, the model updates have to be cast back to full precision, creating additional computational overhead on the client and server. Also, the method has been only tested with small models (< 100K parameters).

FedOBD [Chen *et al.*, 2023] quantizes models with transformer block dropout, i.e., the random removal of entire model blocks. The dropout mechanism is carried out during training by each client and returns only the top-k most important model blocks to communicate. Additionally, FedOBD includes the ideas discussed in Alistarh *et al.* [2017] and Reisizadeh *et al.* [2020] but proposes an optimization problem out of the stochastic quantization where the trade-off originates from entropy and update size. The communication required for FL with a 17M parameter transformer model shows FedOBD to cut communication cost by $2\times$ vs. FedPAQ and by $8\times$ compared to vanilla FedAvg [Chen *et al.*, 2023].

Sparsification (S). While model pruning and sparsification technically have the same objective, pruned models do not necessarily resemble sparse models. A model is sparse once more than 50% of weights are set to 0 [Frankle and Carbin, 2019]. However, pruning can also change the model architecture, i.e., not return the full model. Since it lends its idea from [Frankle and Carbin, 2019], FedPM [Isik *et al.*, 2022] (see Section 4.1) can be considered as a model sparsification technique but does not necessarily lead to sparse networks and may return partial networks. SoteriaFL [Li *et al.*, 2022] guarantees sparse networks while maintaining differential privacy. Equation 7 is amended in such a way that $Q(w_{t+1}^n - w_t)$ is replaced by $C(w_{t+1}^n + \mathcal{N}(0, \sigma^2\mathcal{I}))$ with C resembling a sparse client update through shifted compression that has proven to improve convergence of DL models in FL settings compared to direct compression [Mitra *et al.*, 2021; Mishchenko *et al.*, 2019]. Overall, SoteriaFL mitigates the trade-off between model utility and compression, i.e., the differentially private models converge faster in stricter compression regimes than previously existing non-compressed differentially private approaches.

Gradient projection (GP). Qin *et al.* [2023] introduce FedKSeed tailored to efficiently train FMs in FL applications. They do so by using seeds in the form of scalar vectors to create gradient projections. As only the scalar vectors have to be transmitted, the total amount of communication is reduced by $\geq 99\%$ compared to applications that would send the original multi-dimensional gradients. This is the first technique that enables communication efficient FL applications with FMs, regardless of the training regime (pre-training or fine-tuning).

4.3 Discussion

To date, advancements in communication-efficient methods for FL systems have predominantly focused on training small, full models. The communication paradigm shifts with the emergence of FMs in FL applications. With fine-tuning tasks, we only need to train a small fraction of model parameters, and thus, only trainable parameters have to be communicated. However, each trainable parameter usually contains a high degree of information for a downstream task. As such, the effectiveness of model pruning techniques is unclear as they would cut away fine-tuned parameters. An unexplored space in FL research is the pruning of FMs with subsequent fine-tuning for downstream tasks. Pruning FMs can lead to smaller transformer layers and, consequently, smaller PEFT layers with fewer parameters. In turn, this could positively affect computational and communication efficiency. Full-model compression techniques do not alter the model structure but rather reduce the parameter precision. Thus, these techniques can be used with FL applications and FMs in order to further reduce the size of the communicated updates. Furthermore, Qin *et al.* [2023] have shown that gradient projection is a promising direction to compress model updates without sacrificing significant information. This can be beneficial for PEFT applications as we have small specific adapters for downstream tasks. However, the remaining key challenge is the effect of non-IID data on PEFT. Potential compounding effects of communication compression and lossy compression remain open for investigation.

Framework	Secure Aggregation	Training Efficiency	Communication Efficiency	FM Training / Fine-Tuning	Edge Ready
FLARE	DP, HEC			✓	
FedML	DP, HEC	PEFT		✓	✓
FederatedScope		PEFT		✓	✓
Flower	DP, SMPC	PEFT	Currently, none of the Frameworks implements communication efficient FL methods.	✓	✓
FATE	HEC, SMPC	PEFT		✓	✓
Substra					
PySyft	DP				
OpenFL					
TFF	DP				✓
IBM FL	HEC				

Table 3: Current capabilities of state-of-the-art FL framework with respect to our taxonomy and their ability to run in resource-limited environments. Key: DP = Differential Privacy, HEC = Homomorphic Encryption, SMPC = Secure Multi-Party Computation.

5 Are FL Frameworks Ready for FMs?

The backbones for making FL applications available to a broad audience are FL frameworks implementing recent advancements in FL research. We investigate widely used frameworks for their FM readiness and progress in integrating computational and communication efficiency (Table 3).

In theory, all FL frameworks could handle FMs with sufficient hardware availability. Yet, only some implement efficient training methods. To further drive the adoption of FL in times of FMs, the frameworks need to improve both computationally- and communication-efficient methods for training. FL frameworks that are characterized by their active open-source community, FLARE [Roth *et al.*, 2022], FATE [Fan *et al.*, 2023; Liu *et al.*, 2021], FedML [He *et al.*, 2020], TensorFlow Federated (TFF) [Google, 2019], FederatedScope [Kuang *et al.*, 2023; Xie *et al.*, 2023] and Flower [Beutel *et al.*, 2020], have adopted recent advancements in FL research. The frameworks especially allow for PEFT of FMs with LoRA. Substra [Galtier and Marini, 2019], PySyft [Ziller *et al.*, 2021], OpenFL [Foley *et al.*, 2022], and IBM FL [Ludwig *et al.*, 2020], in their versions as of 2023, focus on training smaller FL tasks with 100K up to a 10M parameters and, therefore, do not provide adapters for FM workloads with more than 100M parameters. Yet, a consistent observation across all frameworks is their lack of efficient communication techniques (e.g., FedOBD). Workloads with FMs will significantly increase communication costs, and the growing use cases involving resource-constrained edge and IoT devices require high efficiency for computation and communication. As such, only those FL frameworks enabling training efficiency are viable choices for working with FMs.

6 Related Work

While there are ample surveys that provide a broad perspective on FL [Li *et al.*, 2023; Banabilah *et al.*, 2022; Liu *et al.*, 2022; Nguyen *et al.*, 2021; Zhang *et al.*, 2021; Aledhari *et al.*, 2020], there are two closely related surveys to our work as they also focus on FMs.

In their survey, Zhuang *et al.* [2023] introduce a broad and general perspective on FMs and FL. They extensively discuss data modalities. This includes access to data across a large number of highly distributed clients and the quality of data that lives on these clients. Currently, FM training or fine-tuning requires datasets with high data quality, i.e., the instructions or texts used for MLM must be curated very carefully. Thus, their survey identifies a stark need for methods to train or fine-tune FMs on a scattered data basis with (highly) varying data quality. Further, Zhuang *et al.* [2023] discuss approaches to integrate FL applications into the lifecycle of FMs, i.e., how FMs can benefit from a continuously evolving system. While their survey briefly touches upon computational efficiency, our study provides an in-depth overview of state-of-the-art training and fine-tuning techniques to render FMs in FL applications a reality. Furthermore, our study includes a comprehensive overview of communication techniques that can enhance the adoption of FMs in communication resource-constrained environments.

Yu *et al.* [2023] provide an overview of FMs and FL with a special focus on privacy, an integral component of FL. Their survey includes a comprehensive overview of different fields of application for FMs, which they divide into once-off training and continual learning. The authors elaborate on technical challenges that may arise for specific use cases, such as robustness towards unreliable clients, varying data quality, the degree of non-IID data, and scalability. In contrast, our survey provides an application-agnostic, in-depth study of existing methods suitable for FM training. Our focus is to outline the technical challenges that currently hinder the operationalization of FM for use cases in federated applications.

7 Conclusions & Future Directions

In this paper, we survey the current landscape of computational and communication efficiency methods in FL systems and introduce a novel taxonomy based on the key techniques. While efficient FL methods have been separate topics on their own in the past, they become closely intertwined as we start using FL systems to leverage FMs. Consequently, the following three questions arise:

What are good and realistic evaluation strategies for generative downstream tasks in FL settings where we do not have control of data? Fine-tuning generative FM requires high-quality data. However, we do not have access to data on the clients to monitor data quality before or during training. As such, estimating data quality is of utmost importance.

How does hyperparameter optimization work for FMs in continuously evolving FL systems? While hyperparameter optimization in FL has been a key challenge, PEFT adds additional complexity. As such FL systems must adapt to the era of FMs by introducing adaptive parameterization for PEFT techniques that can account for changing environmental conditions.

We must develop an understanding of the interplay between PEFT and privacy in FL systems. Communication-efficient FL techniques have been studied for their effect on privacy, but this is still an open topic for PEFT, PT, and IT. While it is proven that PEFT is more sensitive to data heterogeneity, the effects of perturbation through differential privacy are still subject to further studies. The same is true for PT and IT, as both techniques require precise prompts and instructions, respectively. As such, noise may have significantly negative effects here, as well.

Acknowledgments

This work is partially supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01).

References

[Aledhari *et al.*, 2020] Mohammed Aledhari, Rehma Razzak, Reza M. Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[Alistarh *et al.*, 2017] Dan Alistarh, Demjan Grubic, et al. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[Babakniya *et al.*, 2023] Sara Babakniya, Ahmed R. Elkordy, et al. Slora: Federated parameter efficient fine-tuning of language models, 2023.

[Banabilah *et al.*, 2022] Syreen Banabilah, Moayad Aloqaily, et al. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information Processing & Management*, 59(6), 2022.

[Beutel *et al.*, 2020] Daniel J. Beutel, Taner Topal, et al. Flower: A friendly federated learning research framework, 2020.

[Bommasani *et al.*, 2021] Rishi Bommasani, Drew A. Hudson, et al. On the opportunities and risks of foundation models, 2021.

[Chen *et al.*, 2023] Yuanyuan Chen, Zichen Chen, et al. Fedobd: Opportunistic block dropout for efficiently training large-scale neural networks through federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. IJCAI Org., 2023.

[Diao *et al.*, 2020] Enmao Diao, Jie Ding, et al. Heterofl: Computation and communication efficient federated learning for heterogeneous clients, 2020.

[Ding *et al.*, 2023] Ning Ding, Yujia Qin, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, March 2023.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[Erben *et al.*, 2023] Alexander Erben, Ruben Mayer, and Hans-Arno Jacobsen. How can we train deep learning models across clouds and continents? an experimental study, 2023.

[Fan *et al.*, 2023] Tao Fan, Yan Kang, et al. Fate-llm: A industrial grade federated learning framework for large language models, 2023.

[Foley *et al.*, 2022] Patrick Foley, Micah J Sheller, et al. Openfl: the open federated learning library. *Physics in Medicine & Biology*, 67(21), 2022.

[Frankle and Carbin, 2019] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019.

[Galtier and Marini, 2019] Mathieu N Galtier and Camille Marini. Substra: a framework for privacy-preserving, traceable and collaborative machine learning, 2019.

[Google, 2019] Inc. Google. Tensorflow federated: Machine learning on decentralized data. https://www.tensorflow.org/federated, 2019. Accessed: 2023-12-01.

[Habib *et al.*, 2023] Gousia Habib, Tausifa Jan Saleem, and Brejesh Lall. Knowledge distillation in vision transformers: A critical review, 2023.

[He *et al.*, 2020] Chaoyang He, Songze Li, et al. Fedml: A research library and benchmark for federated machine learning, 2020.

[He *et al.*, 2021] Haoyu He, Xingjian Shi, et al. Distiller: A systematic study of model distillation methods in natural language processing, 2021.

[Horváth *et al.*, 2021] Samuel Horváth, Stefanos Laskaridis, et al. FjORD: Fair and accurate federated learning under heterogeneous targets with ordered dropout. In *Advances in Neural Information Processing Systems*, 2021.

[Houlsby *et al.*, 2019] Neil Houlsby, Andrei Giurgiu, et al. Parameter-efficient transfer learning for nlp, 2019.

[Hu *et al.*, 2021] Edward J. Hu, Yelong Shen, et al. Lora: Low-rank adaptation of large language models, 2021.

[Huang *et al.*, 2022] Hong Huang, Lan Zhang, et al. Distributed pruning towards tiny neural networks in federated learning, 2022.

[Isik *et al.*, 2022] Berivan Isik, Francesco Pase, et al. Sparse random networks for communication-efficient federated learning, 2022.

[Jia *et al.*, 2022] Menglin Jia, Luming Tang, et al. Visual prompt tuning, 2022.

[Jiang *et al.*, 2022] Yuang Jiang, Shiqiang Wang, et al. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Kuang *et al.*, 2023] Weirui Kuang, Bingchen Qian, et al. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning, 2023.

[Lagunas *et al.*, 2021] Francois Lagunas, Ella Charlaix, et al. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on EMNLP*, Online and Punta Cana, Dominican Republic, November 2021. ACL.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

[Li *et al.*, 2020] Ang Li, Jingwei Sun, et al. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets, 2020.

[Li *et al.*, 2022] Zhize Li, Haoyu Zhao, et al. Soteriafl: A unified framework for private federated learning with communication compression. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[Li *et al.*, 2023] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, April 2023.

[Liu *et al.*, 2021] Yang Liu, Tao Fan, et al. Fate: An industrial grade platform for collaborative learning with data protection. *JMLR*, 22(1), 2021.

[Liu *et al.*, 2022] Ji Liu, Jizhou Huang, et al. From distributed machine learning to federated learning: a survey. *Knowledge and Information Systems*, 64(4), 2022.

[Longpre *et al.*, 2023] Shayne Longpre, Le Hou, et al. The flan collection: Designing data and methods for effective instruction tuning, 2023.

[Lu *et al.*, 2023] Wang Lu, Xixu Hu, et al. Fedclip: Fast generalization and personalization for clip in federated learning, 2023.

[Ludwig *et al.*, 2020] Heiko Ludwig, Nathalie Baracaldo, et al. Ibm federated learning: an enterprise framework white paper v0.1, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*. PMLR, 2017.

[Mishchenko *et al.*, 2019] Konstantin Mishchenko, Eduard Gorbunov, et al. Distributed learning with compressed gradient differences, 2019.

[Mitra *et al.*, 2021] Aritra Mitra, Rayana Jaafar, George J. Pappas, and Hamed Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. In *Advances in Neural Information Processing Systems*, 2021.

[Nguyen *et al.*, 2021] Dinh C. Nguyen, Ming Ding, et al. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3), 2021.

[OpenAI, 2023] OpenAI. GPT-4 technical report, 2023.

[Penedo *et al.*, 2023] Guilherme Penedo, Quentin Malartic, et al. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

[Peng *et al.*, 2020] Xingchao Peng, Zijun Huang, et al. Federated adversarial domain adaptation. In *ICLR*, 2020.

[Perez *et al.*, 2020] Andres Perez, Valentina Sanguineti, et al. Audio-visual model distillation using acoustic images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[Qin *et al.*, 2023] Zhen Qin, Daoyuan Chen, et al. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes, 2023.

[Radford *et al.*, 2019] Alec Radford, Jeff Wu, et al. Language models are unsupervised multitask learners. 2019.

[Reisizadeh *et al.*, 2020] Amirhossein Reisizadeh, Aryan Mokhtari, et al. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108. PMLR, 2020.

[Roth *et al.*, 2022] Holger R. Roth, Yan Cheng, et al. Nvidia flare: Federated learning from simulation to real-world. 2022.

[Ryabinin *et al.*, 2023] Max Ryabinin, Tim Dettmers, et al. Swarm parallelism: Training large models can be surprisingly communication-efficient, 2023.

[Sanh *et al.*, 2020] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

[Sun *et al.*, 2023] Guangyu Sun, Matias Mendieta, et al. Exploring parameter-efficient fine-tuning for improving communication efficiency in federated learning, 2023.

[Taori *et al.*, 2023] Rohan Taori, Ishaan Gulrajani, et al. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[Tian *et al.*, 2022] Yuanyishu Tian, Yao Wan, et al. Fedbert: When federated learning meets pre-training. *ACM Transactions on Intelligent Systems and Technology*, 13(4), 2022.

[Woisetschläger *et al.*, 2023] Herbert Woisetschläger, Alexander Erben, et al. Federated fine-tuning of llms on the very edge: The good, the bad, the ugly, 2023.

[Xie *et al.*, 2023] Yuexiang Xie, Zhen Wang, et al. Federatedscope: A flexible federated learning platform for heterogeneity. *Proceedings of the VLDB Endowment*, 16(5), 2023.

[Xu *et al.*, 2023] Zheng Xu, Yanxiang Zhang, et al. Federated learning of gboard language models with differential privacy. 2023.

[Yang *et al.*, 2023] Dongchao Yang, Jinchuan Tian, et al. Uniaudio: An audio foundation model toward universal audio generation, 2023.

[Yang *et al.*, 2024] Yiyuan Yang, Guodong Long, Taoshu Shen, Jing Jiang, and Michael Blumenstein. Dual-personalizing adapter for federated foundation models. *ArXiv*, abs/2403.19211, 2024.

[Yousefpour *et al.*, 2023] Ashkan Yousefpour, Shen Guo, et al. Green federated learning, 2023.

[Yu *et al.*, 2023] Sixing Yu, J. Pablo Muñoz, and Ali Jannesari. Federated foundation models: Privacy-preserving and collaborative learning for large models, 2023.

[Zaken *et al.*, 2021] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2021.

[Zhang *et al.*, 2021] Chen Zhang, Yu Xie, et al. A survey on federated learning. *Knowledge-Based Systems*, 216, 2021.

[Zhang *et al.*, 2023a] Jianyi Zhang, Saeed Vahidian, et al. Towards building the federated gpt: Federated instruction tuning, 2023.

[Zhang *et al.*, 2023b] Zhuo Zhang, Yuanhang Yang, et al. FedPETuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*. ACL, 2023.

[Zhao *et al.*, 2022] Haodong Zhao, Wei Du, et al. Fedprompt: Communication-efficient and privacy preserving prompt tuning in federated learning, 2022.

[Zheng *et al.*, 2023] Rui Zheng, Shihan Dou, et al. Secrets of rlhf in large language models part i: Ppo, 2023.

[Zhou *et al.*, 2018] Yiren Zhou, Seyed Moosavi-Dezfooli, et al. Adaptive quantization for deep neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[Zhu and Gupta, 2017] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression, 2017.

[Zhu *et al.*, 2019] Maohua Zhu, Tao Zhang, et al. Sparse tensor core: Algorithm and hardware co-design for vector-wise sparse neural networks on modern gpus. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52. ACM, 2019.

[Zhuang *et al.*, 2023] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions, 2023.

[Ziller *et al.*, 2021] Alexander Ziller, Andrew Trask, et al. *PySyft: A Library for Easy Federated Learning*. Springer International Publishing, 2021.

INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE

TRANSFER OF COPYRIGHT AGREEMENT

Title of Article/Paper: A Survey on Efficient Federated Learning Methods for Foundation Model Training

Publication in Which Article Is to Appear: 33rd International Joint Conference on Artificial Intelligence

Author's Name(s): Herbert Woisetschläger, Alexander Erben, Shiqiang Wang, Ruben Mayer, Hans-Arno Jacoobsen

Please type or print your name as you wish it to appear in print

(Please read and sign Part A only, unless you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign Part A and see item 6 under returned rights.)

PART A–Copyright Transfer Form

The undersigned, desiring to publish the above article/paper in a publication of the International Joint Conferences on Artificial Intelligence, Inc., hereby transfer their copyrights in the above paper to the International Joint Conferences on Artificial Intelligence, Inc. (IJCAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights currently exist or hereafter come into effect, and also the exclusive right to create electronic versions of the paper, to the extent that such right is not subsumed under copyright. The undersigned warrants that he/she is the sole author and owner of the copyright in the above paper, except for those portions shown to be in quotations; that the paper is original throughout; and that the undersigned's right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify IJCAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense IJCAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to IJCAI in the paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorney's fees incurred therein.

Returned Rights

In return for these rights, IJCAI hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. retain all proprietary rights (such as patent rights) other than copyright and the publication rights transferred to IJCAI;
2. personally reuse all or portions of the paper in other works of their own authorship;
3. make oral presentation of the material in any forum;
4. reproduce, or have reproduced, the above paper for the author's personal use, or for company use provided that IJCAI copyright and the source are indicated, and that the copies are not used in a way that implies IJCAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own World Wide Web page or ftp site. Such Web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the IJCAI electronic server (http://www.ijcai.org), and shall not post other IJCAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without IJCAI's written permission;
5. make limited distribution of all or portions of the above paper prior to publication.
6. In the case of work performed under U.S. Government contract, IJCAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes. In the event the above paper is not accepted and published by IJCAI, or is withdrawn by the author(s) before acceptance by IJCAI, this agreement becomes null and void.

Appendix D

Federated Learning Priorities Under the European Union Artificial Intelligence Act

Printed with the permission of

Herbert Woisetschläger, Alexander Erben, Bill Marino, Shiqiang Wang, Nicholas D. Lane, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning Priorities Under the European Union Artificial Intelligence Act." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2402.05968. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/48.pdf

Federated Learning Priorities Under the European Union Artificial Intelligence Act

Herbert Woisetschläger [1] **Alexander Erben** [1] **Bill Marino** [2] **Shiqiang Wang** [3]
Nicholas D. Lane [2 4] **Ruben Mayer** [5] **Hans-Arno Jacobsen** [6]

Abstract

The age of AI regulation is upon us, with the *European Union Artificial Intelligence Act* (AI Act) leading the way. Our key inquiry is how this will affect *Federated Learning* (FL), whose starting point of prioritizing data privacy while performing ML fundamentally differs from that of centralized learning. We believe the AI Act and future regulations could be the missing catalyst that pushes FL toward mainstream adoption. However, this can only occur if the FL community reprioritizes its research focus. In our position paper, we perform a first-of-its-kind interdisciplinary analysis (legal and ML) of the impact the AI Act may have on FL and make a series of observations supporting our primary position through quantitative and qualitative analysis. We explore data governance issues and the concern for privacy. We establish new challenges regarding performance and energy efficiency within lifecycle monitoring. Taken together, our analysis suggests there is a sizable opportunity for FL to become a crucial component of AI Act-compliant ML systems and for the new regulation to drive the adoption of FL techniques in general. Most noteworthy are the opportunities to defend against data bias and enhance private and secure computation.

[1]School of Computation, Information and Technology, Technical University of Munich, Germany [2]Department of Computer Science and Technology, University of Cambridge, United Kingdom [3]IBM T.J. Watson Research Center, United States [4]Flower Labs, Germany [5]Department of Computer Science, University of Bayreuth, Germany [6]Department of Electrical and Computer Engineering, University of Toronto, Canada. Correspondence to: Herbert Woisetschläger <herbert.woisetschlaeger@tum.de>, Alexander Erben <alex.isenko@tum.de>, Bill Marino <wlm27@cam.ac.uk>, Shiqiang Wang <wangshiq@us.ibm.com>, Nicholas D. Lane <ndl32@cam.ac.uk>, Ruben Mayer <ruben.mayer@uni-bayreuth.de>, Hans-Arno Jacobsen <jacobsen@eecg.toronto.edu>.

1. Introduction

On December 8$^{\text{th}}$, 2023, the European Union (EU) Commission and Parliament found a political agreement on an unprecedented regulatory framework – the *EU Artificial Intelligence Act* (AI Act) (Council of the European Union, 2021; European Commission, 2023b). This is the first, but likely one of many regulations that will affect how ML applications are developed, deployed, and maintained. In order to comply with this new landscape, ML of all kinds will likely need to undergo significant changes. Our main focus is on what this means for Federated Learning (FL) (Zhang et al., 2021), a fundamentally different approach to ML that offers unique benefits, such as privacy (Mothukuri et al., 2021) and access to siloed data, compared to its more centralized counterpart. FL enables distributed privacy-preserving learning of models between several clients and a server at scale (McMahan et al., 2017a; Tian et al., 2022) while the training data never leaves the clients, and only the models are communicated. We believe that the AI Act and subsequent regulations could serve as the catalyst to pushing FL towards mainstream adoption. However, this will require the FL community to shift some of its research priorities.

In this position paper, we perform a first-of-its-kind interdisciplinary analysis (legal and ML) of the AI Act and FL (Section 2). Based on our methodology that aligns with the priorities set out by the AI Act (Section 3), we make several key observations in support of our primary position (Section 4):

First, FL struggles to cope with the new performance trade-offs highlighted in the AI Act. As a result, there is a need for a reconsideration of FL research priorities to address these issues, particularly in terms of energy efficiency and the computational costs of privacy. While governance has been a focus for FL in the past, the AI Act brings new challenges, such as performance parity with centralized approaches and lifecycle monitoring under privacy-preserving operations.

Second, FL has inherent advantages over centralized approaches with respect to data lineage and the ability to address bias and related concerns through access to siloed data. However, there are remaining technical hurdles for data man-

agement and governance issues. At the same time, these technical hurdles have been solved in centralized learning due to its lack of concern for data movement and its effects on privacy. It is currently unclear how to cope with GDPR at scale and how the right to privacy will be expressed in practice.

Our analysis indicates due to AI regulation that FL has a significant opportunity to become even more widely adopted. If the FL community can redirect their research efforts to address the new priorities highlighted by the AI Act, and combine this with the inherent advantages of FL, it could become the go-to approach for building compliant ML systems. Therefore, we advocate for a large fraction of the energy that will undoubtedly go into revising all forms of ML to align with the societal values encoded in this act to be directed into FL rather than centralized approaches. This will lead to us more quickly having access to suitable methods for deployment in this new landscape, and we expect the act to be a new driver (along with the long-standing issue of pure privacy) towards the adoption of FL techniques in general.

Our contributions:

- **Requirement analysis for FL based on the AI Act**. We examine the impact of the AI Act on FL systems and methods, outlining requirements and linking them to challenges in FL, aiming to align the legal and ML perspectives.
- **Quantitative and qualitative analysis of FL under the AI Act**. We quantify the costs associated with FL, identify the current inefficiencies, and discuss the potential energy implications. Through our experiments, we introduce the privacy-energy trade-off that arises when fine-tuning a large model in a practical FL framework while aiming to be compliant with the AI Act. Further, we provide a qualitative understanding of the potential of FL under the AI Act.
- **Future outlook on novel research priorities for the FL community.** By distilling our results into a list of future research priorities, we aim to provide guidance such that FL can become the go-to choice for applications incorporating governing EU fundamental rights.

2. The EU Artificial Intelligence Act

The AI Act's latest draft as of January 23rd, 2024 is referenced throughout this section (European Parliament and Council, 2024). This first-of-its-kind, comprehensive, legal framework around AI development and application aims "[...] *to promote* [...] *trustworthy artificial intelligence while ensuring a high level of protection of health, safety, fundamental rights enshrined in the Charter, including* [...] *environmental protection* [...]" (Rec. 1)[1]. While it is not finalized yet and must be implemented as national law in every EU country, it may set the basis for other non-EU jurisdictions to decide their legislation (The White House, 2023; House Of Commons of Canada, 2022). The penalties for violations of the obligations outlined in the AI Act are currently set at a maximum of €35M or 7% of the company's worldwide annual turnover, whichever is higher (Art. 71.1). As such, the fines range in similar dimensions as those of the General Data Protection Regulation (Regulation (EU) 2016/679) ("GDPR") Art. 83.5.

The AI Act differentiates in its classification of AI applications within two dimensions: risk-based (Art. 6) and general-purpose AI models (GPAI) (Art. 52). We specifically cover the risk-based classification and the associated requirements for high-risk systems (Art. 8). If an application falls under this "high-risk" category, it must follow strict robustness and cybersecurity (Art. 15) and data governance guidelines (Art. 10), including compliance with GDPR. Additionally, high-risk system providers may soon have to follow energy-efficiency standards once those are finalized by EU standardization entities (Art. 40.2). As it happens, most applications that benefit from federated aspects fall under this category by default, such as medical applications (Pfitzner et al., 2021) or management of critical infrastructure (electricity, water, gas, heating, or road traffic) (Wang et al., 2021; El Hanjri et al., 2023; Tun et al., 2021; Liu et al., 2020).

The root cause of most GDPR infringements is data collection and unlawful processing (CMS Law, 2024). The AI Act recognizes this fact and emphasizes the importance of the GDPR in its legal text, naming "*data protection by design and default*" and "[...] *ensuring compliance* [...] *may include* [...] *the use of technology that permits algorithms to be brought to the data* [...] *without the transmission between parties*" (Rec. 45a). This aligns with the Act's broad insistence that "*right to privacy and to protection of personal data* [...] *be guaranteed throughout the entire lifecycle of the AI system*" (Rec. 45a). Since FL specifically addresses these privacy concerns and removes data movement and direct access by definition, we must now understand how we can leverage the introduction of the AI Act to enable its legal compliance. For FL, the following three aspects of the AI Act are relevant to understand.

Data Governance. The biggest hurdle that the AI Act imposes on high-risk FL applications is data governance, which requires strong oversight of the data that is being used for the entire model lifecycle of development, training, and deployment (Art. 10.2). The practices shall include an "*examination in view of possible biases that are likely to*

[1]We explain the difference between an article (Art.) and recital (Rec.) in Appendix A. When not specified otherwise, Rec. and Art. refer to the EU AI Act.

affect the health and safety of persons [...]" and "*appropriate measures to detect, prevent and mitigate possible biases*" (Art. 10.2f,fa). With these requirements, we can foresee a future where data access is necessary to comply with forthcoming rules. However, this data access is a reason why data providers might be hesitant to participate, as it is currently unclear how privacy preservation will be enacted and if they might be liable under GDPR.

Federated Learning provides another outlook on this issue. While the training data is not accessible by design with FL and, thus, cannot easily comply with the requirements under Article 10.2, it can ease the angst of data providers as data is processed on a strict "need-to-know" basis and will never be moved from the source. This can be a more promising path forward to create access to data in a privacy-preserving manner, simply due to the number of participants. Additionally, FL includes an emerging research field that implements different techniques to reach specific privacy guarantees, which we cover in Section 4.1. While data quality and techniques to detect biases are recognized as having a high priority when developing DL applications (Whang et al., 2023), FL has to close this gap with indirect techniques to comply with Article 10. It is up to debate if techniques, e.g., that combat non-IID data in a federated setting (Zhao et al., 2018), provide adequate robustness guarantees or if additional safeguards will be needed.

As high-risk applications typically involve personal data and are required to conform to the GDPR Article 10, we take a closer look at how FL is meeting the key requirements of the GDPR:

Security while processing data. GDPR Art. 32.2 calls for strict security guidelines when processing data: "[...] *the appropriate level of security account shall be taken* [...] *that are presented by processing, in particular* [...] *unauthorized disclosure of, or access to personal data transmitted, stored or otherwise processed*". While minimizing the risk of data leakage without any data movement, FL shares the model updates during training, providing an attack vector. To combat this, threat models and security measures for misuse of data by gradient inversion or membership inference attacks have been explored thoroughly (Zhang et al., 2023; Huang et al., 2021; Geiping et al., 2020). FL is also vulnerable to data poisoning attacks whereby attackers corrupt client-side data in an attempt to sabotage the model, which is being combated by comprehensive benchmarking (Han et al., 2023; Zhao et al., 2023). Nevertheless, research on FL security remains a key task, as new attacks could emerge.

The right to information. While access to data is minimized in FL by only sharing model updates, the GDPR reserves the right for individuals to request all information a service provider has stored (GDPR Art. 15, GDPR Rec. 63 & 64). This also includes how data has been used for learning models, which is already being evaluated as client participation is a key priority in FL systems. Existing studies on personalized FL have established accuracy variance and client update norm as metrics to evaluate the value add a client generates for an FL system (Tan et al., 2023; Chen et al., 2022; Fallah et al., 2020).

The right of clients to revoke their consent at any time. With the AI Act installing GDPR as the adjacent privacy regulation, clients in FL systems may make a request to delete their data or revoke their consent to use it at any time (AI Act Art. 17; Art. 7). This can lead to two consequences: the removal of any user data stored in the FL system and, depending on interpretations, the need to unlearn the client's training progress from the global model. Removing the data is trivial, as the data lineage guarantees provided by FL guard the data from being moved from the clients. There are a few approaches to machine unlearning (Xu et al., 2023), such as the teacher-student framework (Kurmanji et al., 2023) or amnesic unlearning (Graves et al., 2021). However, both techniques need access to the training data or even the entire training progress with client-level model snapshots that are usually unavailable in a federated setting. In the specific case of FL, existing works focus on unlearning entire clients and provide a possibility for GDPR compliance (Halimi et al., 2022) without direct data access.

Energy Efficiency. While we focus on high-risk applications, the AI Act also promotes the environmentally sustainable development of AI systems regardless of the application. A voluntary Code of Conduct (CoC) will be drawn up to create clear objectives and key performance indicators (Art. 69) to help set best practices regarding, among others, energy efficiency. It is still up to discussion which high-risk requirements will be included in this CoC, but it is clear that the position of the AI Act reflects a fundamental value of the EU, namely, sustainability. While state-of-the-art data centers are designed to be energy-efficient and capable of running on mostly regenerative energy (Google, 2023), edge clients used in FL are powered by the average energy mix at their locations (Yousefpour et al., 2023; Ritchie & Rosado, 2020). This is echoed by the current trends, which indicate that specialized edge devices can compete with data-center GPUs on sample efficiency (sample-per-Watt) (Woisetschläger et al., 2023), but only when looked at the raw throughput, and not in time-to-accuracy comparing FL to centralized training (cf. Section 4). As such, we find a natural trade-off between energy efficiency and privacy that has yet to be quantified (cf. Section 4.2). Although we see promising progress toward quantifying how and where energy is being consumed in FL applications (Mehboob et al., 2023; Qiu et al., 2023), there are still many fundamental open challenges. For instance, we need to find consensus on how the energy-cost responsibility is being assigned in FL with devices not owned by the training provider and how it

will compare to future energy-consumption baselines.

Robustness and Quality Management. Unsurprisingly, high-risk AI systems should have an "*appropriate level of accuracy, robustness* [...] *and perform consistently*" (Art. 15.1), and this should be guaranteed by a quality management system that takes "[...] *systematic actions to be used for the development, quality control, and quality assurance*" (Art. 17.1c). While it is in the interest of the AI providers to guarantee specific performance goals when deploying, the development of an AI system could be severely prolonged by the need for training to be as energy-efficient as possible. One technique to guarantee model robustness is early stopping to avoid overfitting, which tracks the model performance on a validation dataset (Prechelt, 2002). From the earlier data governance requirements on representative data, the time to validate a model may increase as the validation dataset becomes large (cf. Section 4.3). Combining frequent validation with the need for energy consumption monitoring poses a new optimization problem: Is it more energy-efficient to keep clients idle while a subset validates, or should the next round start in parallel with a chance of overfitting and wasting the energy? As FL stands currently, this shifts the focus towards techniques that increase the validation efficiency per data sample, e.g., as done in dataset distillation (Lei & Tao, 2023). As we anticipate a trade-off between energy efficiency and quality management, this could lead to an increased performance gap between centralized learning and FL.

3. Methodology

Our analysis in Section 2 has highlighted a series of core challenges pertaining to data governance without direct data access, energy efficiency, robustness, and overall quality management. This section presents our evaluation criteria and how they align with the AI Act. We also introduce the methodologies for our qualitative and quantitative analysis.

3.1. Evaluation criteria

Data Governance. Data governance in the AI Act focuses on data bias reduction and strict enforcement of regulatory privacy. Our qualitative analysis focuses on identifying the potential of FL to mitigate data bias. Therefore, we study the effect FL can have on the availability of data such that a broader data basis becomes available for training. A broader and potentially continuously evolving training dataset could improve the generalization capability of a model and better account for minority groups (Torralba & Efros, 2011). For privacy, we look into the technical capabilities of private and secure computing currently available to FL applications. We study whether there is a gap between state-of-the-art technical privacy methods and the regulatory privacy requirements introduced by the AI Act and GDPR.

Energy Efficiency. In centralized DL, we often fine-tune FMs on servers with multiple GPUs and, thus, require very high bandwidth interconnects ($> 200\mathrm{GB/s}$) between the GPUs either via NVLink or Infiniband (Li et al., 2020a; Appelhans & Walkup, 2017). FL only requires low bandwidth interconnects ($< 1\mathrm{GB/s}$) since communication happens sparingly compared to multi-GPU centralized learning (Xu & Wang, 2021). This creates major design differences in the training process and an entirely different cost model. In the following, we point out essential components of the cost model for FL.

The AI Act indicates that further guidelines around energy efficiency are forthcoming. When it comes to how those guidelines define and measure energy efficiency, we propose using a holistic methodology that accounts for computation and communication. Based on such conservative methodology, we can develop comprehensive baselines to compare against. The total energy consumption P consists of two major components, computational P_c and communication energy P_t, i.e., $P = P_c + P_t$.

P_c can be measured directly on the clients via the real-time power draw with an on-board energy metering module (Beutel et al., 2020) or deriving the energy consumption based on floating point operations and a client's system specifications (Desislavov et al., 2023). At the same time, P_t is generally more challenging to measure as multiple network hops are involved. Often, the network infrastructure components, such as switches and routers, are owned by multiple parties and are impossible to monitor for a service provider. However, the bit-wide energy consumption model is available to calculate the cost of transmitting data (Vishwanath et al., 2015). The costs are directly tied to the number of parameters of a client update in an FL system (Yousefpour et al., 2023). As such, we can calculate the total energy consumption of communication as

$$P_\mathrm{t} = E_t \cdot \mathcal{B} = (n_\mathrm{as} \cdot E_\mathrm{as} + E_\mathrm{bng} + n_e \cdot E_e + n_c \cdot E_c + n_d \cdot E_d) \cdot \mathcal{B}. \quad (1)$$

From a client to a server, the communication network and its total energy consumption E_t is organized as follows: E_as, E_bng, E_e, E_c, E_d are the per-bit energy consumption of edge ethernet switches, the broadband network gateway (BNG), one or more edge routers n_e, one or more core routers n_c, and one or more data center Ethernet switches n_d, respectively. To get the total energy consumption for communication, we multiply E_t with the size of a model update d in bits b, $\mathcal{B} = d \cdot b$. Usually, a model parameter has a precision of $b = 32$ bits but can vary based on the specific application (Gupta et al., 2015). Jalali et al. (2014) present the per-bit energy consumption for at least one device per network hop that can be used as a guideline. While it is possible to trace what route a network package takes (Butskoy, 2023), it is currently impossible to track

Table 1: The algorithmic costs estimate how well the privacy mechanisms scale. Especially, the server-side communication provides evidence that the cryptographic methods are significantly more expensive than (ϵ, δ)-DP.

Privacy Technique	Pot. AI Act compliant*	Client Computation	Client Communication	Client Space	Server Computation	Server Communication	Server Space	Algorithm
(ϵ, δ)-DP**	✓	$\mathcal{O}(d)$***	$\mathcal{O}(1)$	$\mathcal{O}(d)$	$\mathcal{O}(\lvert K\rvert)$	$\mathcal{O}(\lvert K\rvert)$	$\mathcal{O}(\lvert K\rvert)$	Andrew et al. (2021)
SMPC	✓	$\mathcal{O}(\lvert K\rvert^2 + \lvert K\rvert \times d)$	$\mathcal{O}(\lvert K\rvert + d)$	$\mathcal{O}(\lvert K\rvert + d)$	$\mathcal{O}(\lvert K\rvert^2 \times d)$	$\mathcal{O}(\lvert K\rvert^2 + \lvert K\rvert \times d)$	$\mathcal{O}(\lvert K\rvert^2 + d)$	Bonawitz et al. (2017)
HEC	Limited	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(\lvert K\rvert \times d)$	$\mathcal{O}(\lvert K\rvert \times d)$	$\mathcal{O}(d)$	Jin et al. (2023)

* Potential evaluation for future AI Act compliance
** $\mathcal{O}(d)$ for computation originates from clipping a model update. When the FL aggregator is running in a secure enclave, we can also clip updates on the server at cost $\mathcal{O}(\lvert K\rvert \times d)$
*** d is the dimensionality of w

the real energy consumption of a data package sent over the network. It specifically depends on what device has been used at what point in the communication chain. As such, if the AI Act requires us to track the *total energy* consumed by a service, we have to develop solutions to track the networking-related energy consumption. We already see promising progress towards holistically accounting for energy efficiency in FL applications (Mehboob et al., 2023; Qiu et al., 2023; Wiesner et al., 2023).

Robustness and Quality Management. Aside from energy, the AI Act also requires FL service providers to provide a robust model with consistently high performance. Since FL does not allow immediate data access, we must find indirect ways to evaluate the model quality and ensure robustness against over-time-evolving input data. We look into what indirect strategies exist to control model quality and measure the cost of existing solutions. Further, we study existing secure and private computing methods with regard to their applicability in FL applications under the AI Act that holds the FL service provider liable for any robustness or quality management issues.

3.2. Quantitative Analysis

We design experiments to quantify those measurable components of changes we have to introduce to FL systems to comply with the AI Act. We use FL to fine-tune a 110M parameter BERT (Devlin et al., 2018) model to classify emails of the 20 News Group dataset (Lang, 1995). Such a setup can be found in job application pre-screening tools, which are classified as *high-risk applications* under the AI Act. Details on the training pipeline and the exact experimental setup are available in Appendix B.

The AI Act *data governance* regulation requires FL service providers to adhere to GDPR and protect data by design. With the absence of data movement in FL applications, we have already taken a major step toward private-by-design applications. However, existing research demonstrates that there are still open attack vectors (Geiping et al., 2020), and closing them comes at a cost. We aim to understand the trade-off between scaling a system and the cost incurred by introducing private and secure computation methods (Section 4.1).

The forthcoming introduction of the AI Act *energy efficiency* directives may require us to implement FL applications with sustainable and energy-saving techniques in mind. However, the additional duties to account for data governance, robustness, and quality management require us to frequently analyze the FL model, track the energy consumption of the whole system, and ensure privacy throughout the entire application. As this introduces a computational overhead, we aim to understand exactly where potentials for improved energy efficiency can be found and how to address them (Section 4.2).

The *robustness and quality management* requirements introduce the necessity of closely monitoring the FL model while training. This is to ensure consistently high performance. Close monitoring naturally increases the requirement to communicate and validate the FL model. This incurs additional costs. We evaluate the question of how expensive robustness and quality management are in FL applications and how they could be mitigated (Section 4.3).

3.3. Qualitative Analysis

In our qualitative analysis, we focus specifically on the characteristics of FL that are not empirically measurable. To do so, we take legislators' perspective and look at the qualitative potential of FL. We aim to identify the potential of FL to serve the fundamental rights of privacy and data bias prevention. Our objective is to evaluate whether FL has the significant potential to become *the* most adopted privacy-preserving ML technique for high-risk applications under the AI Act.

Overall, our analysis aims to add to the understanding of the future potential of FL under the AI Act and derive research priorities to help with the broad adoption of FL.

4. Analysis

Our analysis combines quantitative analysis considering data governance, energy efficiency, as well as robustness, and quality management. We expand on our empirical results with a qualitative analysis to identify the characteristics of FL under the AI Act that cannot be easily measured. The **key insight** are highlighted.

4.1. Data Governance

Secure Multi-Party Computation (SMPC), Homomorphic Encryption (HEC), and (ϵ, δ)-Differential Privacy ((ϵ, δ)-DP) all provide technical measures to improve data privacy in FL. SMPC and HEC are cryptographic methods that rely on key exchange between clients (Bonawitz et al., 2017; Jin et al., 2023). The client model update encryption removes the ability to track a client's individual contribution toward a global model. At the same time, aggregation remains possible as SMPC and HEC keep arithmetic properties.

A clear strategy to employing the *right* private and secure computation technique in an AI Act-compliant FL system is required. We find all methods to come at significant costs (Figure 1 and Table 1). While the cryptographic methods keep the original shape of the model updates in an encrypted form, they require extensive communication and, in the worst case, point-to-point communication between clients. This creates practical challenges when scaling an FL system (Jin et al., 2023). However, this is where (ϵ, δ)-DP excels (McMahan et al., 2017b; Andrew et al., 2021). Instead of requiring the clients K to establish a joint secure computation regime, (ϵ, δ)-DP introduces privacy by model parameter perturbation. In detail, we perturb and clip each model weight $w_{t+1}^k \in \mathbb{R}^d$ with dimension d of a client $k \in K$ with random noise ξ sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_\Delta^2)$. The variance σ_Δ^2 depends on the number of clients per aggregation round and how many clients have exceeded the clipping threshold in the previous training round. The quantity z scales the noise that is actually applied to local client update w_{t+1}^k and ultimately determines the degree of privacy we achieve under a constrained privacy budget ϵ and a data leakage risk δ,

$$w_{t+1} = \frac{1}{|K|} \sum_{k=1}^{|K|} \left(w_{t+1}^k + z \cdot \xi \right). \tag{2}$$

As can be seen, the perturbation mechanism benefits from increasing the number of clients in an aggregation round. Thus, (ϵ, δ)-DP is particularly useful for scaled systems, while the cryptographic methods can be useful in smaller systems. The optimal strategy for choosing the right privacy technique is yet to be found.

It is unclear whether (ϵ, δ)-DP can be compliant with regulatory privacy as enacted by GDPR. While we know that an $\epsilon \leq 1$ provides strong guarantees for privacy, the guarantee always depends on δ (Dwork & Roth, 2013). While setting δ is trivial in centralized learning, as we know the dataset size before training, it is challenging in FL. We cannot be certain about how many clients will eventually participate in the training process and how many data points each client contributes. As such, we require heuristics to set δ appropriately for training in a dynamically evolving FL system. An effort to evaluate (ϵ, δ)-DP for regulatory compliance is planned in the United States (The White House, 2023); the same should be done in the EU, potentially as a joint effort.

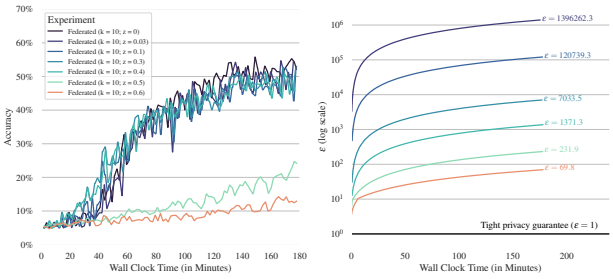


Figure 1: To achieve high privacy guarantees in small systems, we require high z that come at significant model performance and efficiency costs. Training stability also diminishes with increasing z. ϵ is calculated based on $\delta = \frac{1}{16{,}000}$.

4.2. Energy Efficiency

Optimizing for energy efficiency must become a priority in FL research. In our experiment with the BERT pipeline, we find FL to be $5\times$ less energy efficient than centralized training when looking at the computational and communication effort to reach 50% accuracy (TTA50) (Figure 2). Communication accounts for 26% of the total energy draw. With parameter-efficient fine-tuning (PEFT), we save on computational resources and reduce communication costs significantly.[2] As such, efficient methods for computation in large models can also reduce communication. First works have shown significant potential for PEFT methods to address energy efficiency but still come at extensive warm-up costs (Babakniya et al., 2023) that we need to mitigate in the future. As the EU has introduced the Emission Trading System-2 (ETS-2), CO_2 emissions are capped by the number of total emission certificates available (Abrell et al., 2023). This immediately impacts the electricity price, which will surge as the ETS-2 Market Mechanism comes into effect in 2027. The price for CO_2 certificates is expected to rise by as much as $6\times$, from €45 in 2024 to €300 with a market-made price. Overall, this increases the need for energy efficiency improvements.

The AI Act introduces a privacy-energy efficiency trade-off. As pointed out in Section 4.1, we do not know about the *right* choice of private and secure computation for any given FL application as it depends on the number of clients in the system, the number of clients per training round, and the amount of input data available on each client. The cryp-

[2] It is important to note if we were to use full-model fine-tuning, the power consumption for computing would amount to 0.48 kWh and communication to 196 kWh to reach TTA50 ($2100\times$ more than PEFT). We communicate 110M parameters over 1,057 rounds.

tographic methods introduce significant computational and communication overhead, while (ϵ, δ)-DP does not. However, for small-scale FL systems (< 100 clients per aggregation round), the z has to be comparably larger than in large-scale systems (McMahan et al., 2017b). This significantly reduces the model performance and slows the training process (Figure 1). As such, we face a privacy-energy trade-off in current-state FL systems, regardless of the private and secure computation technique. We must address this challenge in light of the AI Act and its call for more energy efficiency.

4.3. Robustness & Quality Management

We pay significantly for robustness guarantees. Frequent validation in FL under the control of the service provider (Rec. 45), i.e., the server, is a necessity to track model performance, understand a model's robustness against data heterogeneity (Li et al., 2021), and domain shifts (Huang et al., 2023). However, the energy consumption of idle clients while waiting for a model to validate and be ready for the next aggregation round has not been part of the power equation thus far. With the AI Act, a service provider may have to account for the *total energy* consumed during training (Art. 40, Rec. 85a). Thus, we must account for these idle times as well. As seen in Figure 2, these idle times consume 31% of all power. To address this challenge, we could regulate the validation process. Similar to what has been done for fair FL methods, we can make validation depend on the loss volatility (Li et al., 2020b;c) and validate as follows:

1. *Only validate the final model*. The fastest way to train is to only validate the final model. However, this approach induces the risk of creating a model with no utility and wasting all energy consumed. Also, legal compliance is in doubt since sparse monitoring contradicts the AI Act requirements (Art. 17).
2. *Validate after every i^{th} aggregation round*. While a frequent validation strategy reduces the risk of overfitting a model, it creates significant idle time. Trading off the validation frequency for energy efficiency could be a promising approach to achieving full compliance with the AI Act.
3. *Validate asynchronously*. We may validate models while starting the next aggregation round to avoid any idle energy consumption. This bears the risk of producing an overfitted model but can save energy after all. A careful trade-off can help create an energy-efficient system while producing robust models.

The applicability of HEC under the AI Act is potentially limited. Since HEC denies server-side model evaluation by design (Jin et al., 2023), we must rely on client-side validation techniques. This is only feasible in applications with trustworthy clients and where validation datasets can be distributed to clients. Promising directions for trustworthy computing are secure enclaves and trusted execution environments (Sabt et al., 2015). In the case of client-side validation under the AI Act, the FL service provider still remains liable for a consistent and high-quality model.

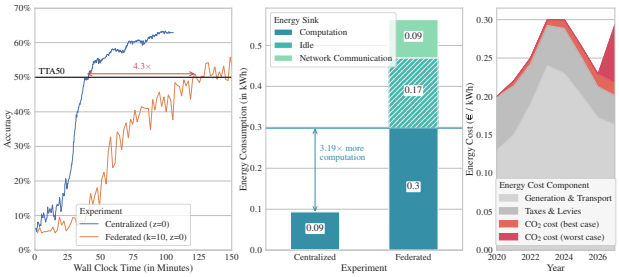


Figure 2: Baseline Experiments. We identify major causes of energy efficiencies in FL systems. The projected energy costs in the EU, especially CO_2 pricing, require us to focus on improving the energy efficiency of FL.

4.4. Qualitative Analysis

Access to siloed data. Creating data sets is complex and can, at best, be based on the entire internet (Schuhmann et al., 2022; Gao et al., 2020). Storing and transmitting such huge amounts of data can quickly become costly. Additionally, data quality is just as important as the data itself (Whang et al., 2023). We assume that a lot of high-quality, simultaneously personally identifiable data is naturally not publicly accessible. Despite the EU's plan to make anonymized data available worldwide (Rec. 45), collecting such data poses significant challenges, as we outlined in this section. FL can provide us access to this data, potentially greatly improving the high-risk application's functionality.

With broader data access, we generate more representative models and data. The AI Act emphasizes the importance of examining and mitigating potential biases in the data used for training. This is particularly important if these biases affect fundamental rights (Art. 10.2f). To achieve this, the datasets must be curated and prepared for training after they are centrally aggregated. If a concept shift alters the basic assumptions about the data (Lu et al., 2019), the dataset must be adjusted anew. FL offers a potential solution to this problem. As FL operates on the clients close to the data source, it means that, by definition, we have access to the latest and most representative data. Given that we train for many rounds and randomly sample clients for aggregation out of an evolving client base, we automatically create a representative global model over time since the model evolves along with the client base. As such, it can be easier to comply with the AI Act requirements by design.

FL provides simple data lineage. Since the training data never leaves the clients in FL, it is less complex to track the

data lineage, meaning where the data originated, where it has gone, and its usage. On the one hand, as data is easy to trace, the GDPR requirement to know how the data is being used (cf. Section 2) is easy to answer and easier to ensure. On the other hand, every time data is sent, it is open to man-in-the-middle attacks (Conti et al., 2016), and when it is stored in multiple locations, all data hosts are vulnerable to unauthorized access. Additionally, every time a human is in the loop regarding data management, there is a potential risk of error (Evans et al., 2016), which can lead to data leakage to a third party. This lack of vulnerability in FL systems removes most of the potential penalties under the GDPR, which are closing in at €4.5B over the last five years by January 2024 (CMS Law, 2024). This fact alone can encourage data providers to make data available to FL applications as the risk on their side is significantly lower than before.

5. Future Research Priorities

The challenges highlighted in our analysis indicate that FL can strongly align with the needs of the AI Act if the core challenges are being addressed soon. To do so, we outline the future research priorities that we see as a necessary redirection for the FL community to make FL a legally compliant and commercially viable solution. The **research priorities** are highlighted.

The data quality requirements are currently not amenable to FL. We need to find solutions to meet the data quality requirements of the AI Act under Art. 10 without having direct access to the data. First, if data quality can be indirectly inferred through techniques with heavy energy investment, how do they compare to direct techniques that require direct data access? Second, do techniques that combat non-IID data provide enough robustness guarantees to qualify for compliance, and if not, what is missing? Third, the AI Act mentions that data processing methods at the source are desirable (cf. Section 2), but it is not made clear who is responsible for the data if multiple parties are involved. To make meaningful progress towards the goals of the AI Act, it is imperative that the FL research community focuses on improving data quality techniques and ensures that Art. 10, under legal guidance, can be effectively implemented in real-world systems.

CO_2-based optimization to compete with centralized training. FL is currently not achieving competitive energy efficiency compared to centralized training (cf. Section 4.2). Even if using DP will be considered partially compliant regarding data governance, it results in extensive energy costs, just as training and quality monitoring do. Therefore, we require new techniques to address these costs concurrently. While there is ongoing research focused on energy efficiency in specific use cases (Yousefpour et al., 2023; Salh et al., 2023; Kim et al., 2023; Albelaihi et al., 2022), there is a need for a designated effort to bridge the gap to centralized baselines, which might be running on fully renewable energy or be more energy efficient by default due to locality.

Expression of privacy in the context of the EU AI Act. FL is private by design and should fit EU AI Act compliance well. Unfortunately, we found significant shortfalls in energy efficiency and data governance compared to centralized training (Section 4). If the FL research community does not act now, centralized training may be seen as the best approach for high-risk applications. This could pose a problem for individuals if privacy is not considered a key component from the outset. If centralized training is deemed the best approach due to better energy efficiency and easier data governance compliance, it is unclear how the right to privacy will be expressed in practice. It is crucial that the interpretation of the law, such as with the GDPR and subsequent cookie banners (The European Commission, 2023), does not result in the end-user bearing the entire burden while operators take no responsibility.

Privacy-preserving techniques alignment within the AI Act. We evaluated SMPC, HEC, and (ϵ, δ)-DP within their current applicability to the energy and data governance aspects and found them to be lacking in multiple ways (cf. Section 4.1). From a technical point of view, we need to work on improving these techniques to be more energy-efficient. However, researchers should advocate for concrete privacy goals to help align legal and arithmetic privacy.

Technical framework for regulatory compliance and representative AI Act baselines. We require a framework that specifically caters to FL, as it has distinct differences from centralized DL in terms of model lifecycle and data access. This framework is necessary so that not everyone is faced with complying with the AI Act from the outset, but to propose best practices to provide a solid basis (in conjunction with the standardization organizations in Art. 40). Through this framework, the development of comparable baselines is necessary to set the standard on privacy-by-design deep learning in high-risk applications. Specifically, this framework should strive to standardize edge hardware comparisons, clarify who is responsible for customer energy costs, and establish clear targets for training and deployment.

6. Conclusion

In this position paper, we analyze the AI Act and its impact on FL. We outline how we need to redirect research priorities with regard to achieving regulatory compliance, the energy-privacy trade-off introduced by the AI Act, and the need for new optimization dimensions in FL. Depending on forthcoming energy efficiency requirements, it may also

require us to think about holistic monitoring systems while staying energy efficient. It is also important to address challenges that have been solved in centralized learning such that FL can keep up. With this, we, as the FL research community, can send a clear signal to legislation and the broad public that we have a strong interest in making FL *the* distributed privacy-preserving DL technology of the future by incorporating societal priorities into our research. We can do so by answering the open call by the EU Commission to support the newly established EU AI Office to close the gap between regulatory framing and technical implementation (Nature, 2024).

Impact Statement

This paper presents work whose goal is to suggest future research directions that will help ensure that FL, with its worthwhile goal of preserving privacy, aligns with other societal values espoused by the EU AI Act, such as keeping AI systems robust, unbiased, energy efficient, transparent, ethical, and secure, especially for high-risk use cases. This paper transparently addresses the challenges that FL may encounter as regards the data governance, energy efficiency, and robustness provisions of the Act and the associated trade-offs that AI providers must be aware of and responsibly navigate when complying with the Act and the societal ideals it encapsulates.

Acknowledgements

We would like to thank Hadrien Pouget at the Carnegie Endowment for International Peace, Yulu Pi at the Leverhulme Center for the Future of Intelligence, Bill Shen at the University of Cambridge, and The Future Society for their help in understanding the legal aspects of the AI Act and the White House Executive Order on AI.

This work is partially funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01) and the German Research Foundation (Grant: 392214008).

We welcome feedback on our work and are open to including aspects that we might have missed in future versions of this paper.

References

Abrell, J., Bilici, S., et al. Optimal allocation of the EU carbon budget: A multi-model assessment, may 2023. URL https://ariadneprojekt.de/media/2022/06/Ariadne-Analysis_Carbon-Budget-multi-model-assessment_June2022.pdf. Document 32023L0959.

Albelaihi, R., Yu, L., Craft, W. D., Sun, X., Wang, C., and Gazda, R. Green Federated Learning via Energy-Aware Client Selection. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 13–18, 2022. doi: 10.1109/GLOBECOM48099.2022.10001569.

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

Appelhans, D. and Walkup, B. Leveraging NVLINK and asynchronous data transfer to scale beyond the memory capacity of GPUs. In *Proceedings of the 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, SC '17. ACM, November 2017. doi: 10.1145/3148226.3148232. URL http://dx.doi.org/10.1145/3148226.3148232.

Babakniya, S., Elkordy, A. R., Ezzeldin, Y. H., Liu, Q., Song, K.-B., El-Khamy, M., and Avestimehr, S. SLoRA: Federated parameter efficient fine-tuning of language models, 2023. URL https://arxiv.org/abs/2308.06522.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K. H., Parcollet, T., de Gusmão, P. P. B., and Lane, N. D. Flower: A Friendly Federated Learning Research Framework, 2020. URL https://arxiv.org/abs/2007.14390.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17. ACM, October 2017. doi: 10.1145/3133956.3133982. URL http://dx.doi.org/10.1145/3133956.3133982.

Butskoy, D. Linux Traceroute, 12 2023. URL http://traceroute.sourceforge.net/.

Chen, D., Gao, D., Kuang, W., Li, Y., and Ding, B. pFL-Bench: A Comprehensive Benchmark for Personalized Federated Learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=2ptbv_JjYKA.

CMS Law. GDPR Enforcement Tracker, 01 2024. URL https://www.enforcementtracker.com/.

Conti, M., Dragoni, N., and Lesyk, V. A Survey of Man In The Middle Attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016. doi: 10.1109/COMST.2016.2548426.

Council of the European Union. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL - LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, apr 2021. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206. Document 52021PC0206.

Desislavov, R., Martínez-Plumed, F., and Hernández-Orallo, J. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857, April 2023. ISSN 2210-5379. doi: 10.1016/j.suscom.2023.100857. URL http://dx.doi.org/10.1016/j.suscom.2023.100857.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2013. ISSN 1551-3068. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

El Hanjri, M., Kabbaj, H., Kobbane, A., and Abouaomar, A. Federated learning for water consumption forecasting in smart cities. In *ICC 2023-IEEE International Conference On Communications*, pp. 1798–1803. IEEE, 2023.

European Commission. Market analysis - Electricity market - recent developments, 07 2023a. URL https://energy.ec.europa.eu/data-and-analysis/market-analysis_en.

European Commission. A European approach to artificial intelligence, 2023b. URL https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence.

European Environment Agency. Greenhouse gas emission intensity of electricity generation, Oct 2023. URL https://www.eea.europa.eu/data-and-maps/daviz/co2-emission-intensity-14#tab-chart_7.

European Parliament and Council. Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, 01 2024. URL https://www.linkedin.com/posts/dr-laura-caroli-0a96a8a_ai-act-consolidated-version-activity-7155181240751374336-B3Ym/.

Eurostat. Electricity prices for household consumers, Oct 2023. URL https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Electricity_price_statistics.

Evans, M., Maglaras, L. A., He, Y., and Janicke, H. Human behaviour as an aspect of cybersecurity assurance. *Security and Communication Networks*, 9(17):4667–4679, 2016.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3557–3568. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/24389bfe4fe2eba8bf9aa9203a44cdad-Paper.pdf.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. Inverting Gradients - How easy is it to break privacy in federated learning? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16937–16947. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf.

Google. 24/7 Carbon-Free Energy by 2030, 2023. URL https://www.google.com/about/datacenters/cleanenergy/.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021.

Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. Deep Learning with Limited Numerical Precision. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1737–1746. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/gupta15.html.

Halimi, A., Kadhe, S., Rawat, A., and Baracaldo, N. Federated Unlearning: How to Efficiently Erase a Client in FL?, 2022. URL https://arxiv.org/abs/2207.05521.

Han, S., Buyukates, B., Hu, Z., Jin, H., Jin, W., Sun, L., Wang, X., Wu, W., Xie, C., Yao, Y., Zhang, K., Zhang, Q., Zhang, Y., Avestimehr, S., and He, C. FedMLSecurity: A Benchmark for Attacks and Defenses in Federated Learning and Federated LLMs, 2023.

He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Zhu, X., Wang, J., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., and Avestimehr, S. FedML: A Research Library and Benchmark for Federated Machine Learning, 2020. URL https://arxiv.org/abs/2007.13518.

House Of Commons of Canada. An Act to enact the Consumer Privacy Protection Act, the Personal Information and Data Protection Tribunal Act and the Artificial Intelligence and Data Act and to make consequential and related amendments to other Acts, 6 2022. URL https://www.parl.ca/DocumentViewer/en/44-1/bill/C-27/first-reading.

Huang, W., Ye, M., Shi, Z., Li, H., and Du, B. Rethinking Federated Learning With Domain Shift: A Prototype View. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16312–16322, June 2023.

Huang, Y., Gupta, S., Song, Z., Li, K., and Arora, S. Evaluating Gradient Inversion Attacks and Defenses in Federated Learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=0CDKgyYaxC8.

Jalali, F., Ayre, R., Vishwanath, A., Hinton, K., Alpcan, T., and Tucker, R. Energy Consumption of Content Distribution from Nano Data Centers versus Centralized Data Centers. *ACM SIGMETRICS Performance Evaluation Review*, 42(3):49–54, December 2014. ISSN 0163-5999. doi: 10.1145/2695533.2695555. URL http://dx.doi.org/10.1145/2695533.2695555.

Jin, W., Yao, Y., Han, S., Joe-Wong, C., Ravi, S., Avestimehr, S., and He, C. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System, 2023. URL https://arxiv.org/abs/2303.10837.

Kim, M., Saad, W., Mozaffari, M., and Debbah, M. Green, Quantized Federated Learning over Wireless Networks: An Energy-Efficient Design. *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023. doi: 10.1109/TWC.2023.3289177.

Klimas, T. and Vaiciukaite, J. The law of recitals in European Community legislation. *ILSA J. Int'l & Comp. L.*, 15:61, 2008.

Kurmanji, M., Triantafillou, P., and Triantafillou, E. Towards Unbounded Machine Unlearning. *arXiv preprint arXiv:2302.09880*, 2023.

Lang, K. NewsWeeder: Learning to Filter Netnews. In Prieditis, A. and Russell, S. (eds.), *Machine Learning Proceedings 1995*, pp. 331–339. Morgan Kaufmann, San Francisco (CA), 1995. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50048-7. URL https://www.sciencedirect.com/science/article/pii/B9781558603776500487.

Lei, S. and Tao, D. A comprehensive survey to dataset distillation. *arXiv preprint arXiv:2301.05603*, 2023.

Li, A., Song, S. L., Chen, J., Li, J., Liu, X., Tallent, N. R., and Barker, K. J. Evaluating Modern GPU Interconnect: PCIe, NVLink, NV-SLI, NVSwitch and GPUDirect. *IEEE Transactions on Parallel and Distributed Systems*, 31(1):94–110, January 2020a. ISSN 2161-9883. doi: 10.1109/tpds.2019.2928289. URL http://dx.doi.org/10.1109/TPDS.2019.2928289.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated Optimization in Heterogeneous Networks. In Dhillon, I., Papailiopoulos, D., and Sze, V. (eds.), *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020b. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*, 2020c. URL https://openreview.net/forum?id=ByexElSYDr.

Li, X., JIANG, M., Zhang, X., Kamp, M., and Dou, Q. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6YEQUn0QICG.

Liu, Y., James, J., Kang, J., Niyato, D., and Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, 7(8):7751–7763, 2020.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.

McMahan, B., Moore, E., et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017a. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning Differentially Private Recurrent Language Models, 2017b. URL https://arxiv.org/abs/1710.06963.

Mehboob, T., Bashir, N., Iglesias, J. O., Zink, M., and Irwin, D. Cefl: Carbon-efficient federated learning, 2023. URL https://arxiv.org/abs/2310.17972.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

Nature. There are holes in Europe's AI Act — and researchers can help to fill them. *Nature*, 625(7994):216–216, January 2024. ISSN 1476-4687. doi: 10.1038/d41586-024-00029-4. URL http://dx.doi.org/10.1038/d41586-024-00029-4.

Pfitzner, B., Steckhan, N., and Arnrich, B. Federated learning in a medical context: a systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31, 2021.

Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pp. 55–69. Springer, 2002.

Qiu, X., Parcollet, T., Fernandez-Marques, J., Gusmao, P. P. B., Gao, Y., Beutel, D. J., Topal, T., Mathur, A., and Lane, N. D. A first look into the carbon footprint of federated learning. *Journal of Machine Learning Research*, 24(129):1–23, 2023. URL http://jmlr.org/papers/v24/21-0445.html.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization, 2020. URL https://arxiv.org/abs/2003.00295.

Ritchie, H. and Rosado, P. Energy Mix. *Our World in Data*, 2020. https://ourworldindata.org/energy-mix.

Sabt, M., Achemlal, M., and Bouabdallah, A. Trusted Execution Environment: What It is, and What It is Not. In *2015 IEEE Trustcom/BigDataSE/ISPA*. IEEE, August 2015. doi: 10.1109/trustcom.2015.357. URL http://dx.doi.org/10.1109/Trustcom.2015.357.

Salh, A., Ngah, R., Audah, L., Kim, K. S., Abdullah, Q., Al-Moliki, Y. M., Aljaloud, K. A., and Talib, H. N. Energy-Efficient Federated Learning With Resource Allocation for Green IoT Edge Intelligence in B5G. *IEEE Access*, 11:16353–16367, 2023. doi: 10.1109/ACCESS.2023.3244099.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, December 2023. ISSN 2162-2388. doi: 10.1109/tnnls.2022.3160699. URL http://dx.doi.org/10.1109/TNNLS.2022.3160699.

The European Commission. Cookie Pledge, 2023. URL https://commission.europa.eu/live-work-travel-eu/consumer-rights-and-complaints/enforcement-consumer-protection/cookie-pledge_en.

The White House. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence, 10 2023. URL https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/.

Tian, Y., Wan, Y., et al. FedBERT: When Federated Learning Meets Pre-training. *ACM Transactions on Intelligent Systems and Technology*, 13(4):1–26, August 2022. ISSN 2157-6912. doi: 10.1145/3510033. URL http://dx.doi.org/10.1145/3510033.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Tun, Y. L., Thar, K., Thwal, C. M., and Hong, C. S. Federated learning based energy demand prediction with clustered aggregation. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 164–167. IEEE, 2021.

Vishwanath, A., Jalali, F., Hinton, K., Alpcan, T., Ayre, R. W. A., and Tucker, R. S. Energy Consumption Comparison of Interactive Cloud-Based and Local Applications. *IEEE Journal on Selected Areas in Communications*, 33(4):616–626, April 2015. ISSN 0733-8716. doi: 10.1109/jsac.2015.2393431. URL http://dx.doi.org/10.1109/JSAC.2015.2393431.

Wang, Y., Bennani, I. L., Liu, X., Sun, M., and Zhou, Y. Electricity consumer characteristics identification: A federated learning approach. *IEEE Transactions on Smart Grid*, 12(4):3637–3647, 2021.

Whang, S. E., Roh, Y., Song, H., and Lee, J.-G. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

Wiesner, P., Khalili, R., Grinwald, D., Agrawal, P., Thamsen, L., and Kao, O. Fedzero: Leveraging renewable excess energy in federated learning. 2023. doi: 10.48550/ARXIV.2305.15092. URL https://arxiv.org/abs/2305.15092.

Woisetschläger, H., Isenko, A., et al. FLEdge: Benchmarking Federated Machine Learning Applications in Edge Computing Systems. *arXiv preprint arXiv:2306.05172*, 2023.

Xu, H., Zhu, T., Zhang, L., Zhou, W., and Yu, P. S. Machine Unlearning: A Survey. *ACM Comput. Surv.*, 56(1), aug 2023. ISSN 0360-0300. doi: 10.1145/3603620. URL https://doi.org/10.1145/3603620.

Xu, J. and Wang, H. Client Selection and Bandwidth Allocation in Wireless Federated Learning Networks: A Long-Term Perspective. *IEEE Transactions on Wireless Communications*, 20(2):1188–1200, February 2021. ISSN 1558-2248. doi: 10.1109/twc.2020.3031503. URL http://dx.doi.org/10.1109/TWC.2020.3031503.

Yousefpour, A., Guo, S., Shenoy, A., Ghosh, S., Stock, P., Maeng, K., Krüger, S.-W., Rabbat, M., Wu, C.-J., and Mironov, I. Green Federated Learning, 2023. URL https://arxiv.org/abs/2303.14604.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, March 2021. ISSN 0950-7051. doi: 10.1016/j.knosys.2021.106775. URL http://dx.doi.org/10.1016/j.knosys.2021.106775.

Zhang, L., Li, L., Li, X., Cai, B., Gao, Y., Dou, R., and Chen, L. Efficient Membership Inference Attacks against Federated Learning via Bias Differences. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID 2023. ACM, October 2023. doi: 10.1145/3607199.3607204. URL http://dx.doi.org/10.1145/3607199.3607204.

Zhao, H., Du, W., Li, F., Li, P., and Liu, G. FedPrompt: Communication-Efficient and Privacy-Preserving Prompt Tuning in Federated Learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Appendix

A. Details on the AI Act

In this appendix section, we provide additional background on the legal aspects of our work.

A.1. Article vs. Recitals in the AI Act

In our main paper, we argue with Articles and Recitals. Understanding the difference between both is vital. The following explanations are based on Klimas & Vaiciukaite (2008).

Article. An article formulates the actual binding law and defines requirements that need to be implemented in technical solutions. This is ultimately what decides on violations. However, some parts can appear ambiguous and leave room for interpretation. This is where Recitals come into play.

Recitals. They provide interpretation to the Articles and help in guiding what needs to be done to ensure full compliance by reciting elements of the Articles and putting them into context. As such, Recitals provide procedural details on how to implement a law in practice. While they form the basis for a common understanding of the AI Act, they are not legally binding.

A.2. The latest AI Act version

By the time of writing this paper late 2023 and early 2024, the official Journal of the European Union hosts the original draft of the AI Act, which was released on Apr. 21st, 2021. In January 2024, EU policymakers and journalists released the pre-final version of the AI Act based on the high public demand. Our work is based on this latest version since it contains the final regulation as it will eventually come into effect. It is available here: https://www.linkedin.com/posts/dr-laura-caroli-0a96a8a_ai-act-consolidated-version-activity-7155181240751374336-B3Ym/ and https://drive.google.com/file/d/1xfN5T8VChK8fSh3wUiYtRVOKIi9oIcAF/view.

B. Additional Experimental Details

Table 2: Training hyperparameters per training regime.

Training regime	Data Dist.	Tot. Samples Seen	Client							Server				
			MB Size	Optimizer	LR	WD	Mom.	Damp.	Loc. Iter.	K	k	Strategy	LR	Mom.
Centralized	IID	80K	20	SGD	0.01	0.001	0.9	0.9	5	–	–	–	–	–
Federated	non-IID	80K	2	SGD	0.01	0.001	0.0	0.0	2	100	10	FedAvgM	1.0	0.9

Here, we provide additional details about our experimental results. For our empirical evaluations, we fine-tune the 110M parameter BERT transformer (Devlin et al., 2018) over the 20 News Group Dataset (Lang, 1995) such that we can reliably classify emails into one of 20 categories. For example, such a classification application can be used in a company's human resource processes to screen job applications. Under the AI Act, such a system is considered a high-risk application.

B.1. Dataset

In our empirical analysis, we use a state-of-the-art text classification task in FL research by means of the 20 Newsgroup Dataset (Lang, 1995), which consists of 18,000 email bodies that each belong to one of 20 classes. The dataset has a total of $18,000$ samples, of which we use $16,000$ for training, $1,000$ for validation, and $1,000$ for testing. As our work aims to quantify the cost of FL and associated private computing methods in realistic systems in line with the EU AI Act requirements (Council of the European Union, 2021), we chose to sample 100 non-IID client subsets via a Latent Dirichlet Allocation (LDA) with $\alpha = 1.0$, which is widely used in FL research (Babakniya et al., 2023; He et al., 2020; Reddi et al., 2020). The data distribution is visualized in Figure 3.

B.2. Model

We fine-tune the BERT model (Devlin et al., 2018) with 110M parameters by using the parameter-efficient fine-tuning technique Low-Rank Adapters (LoRA). We use a LoRA configuration that has been well explored in FL settings (Babakniya

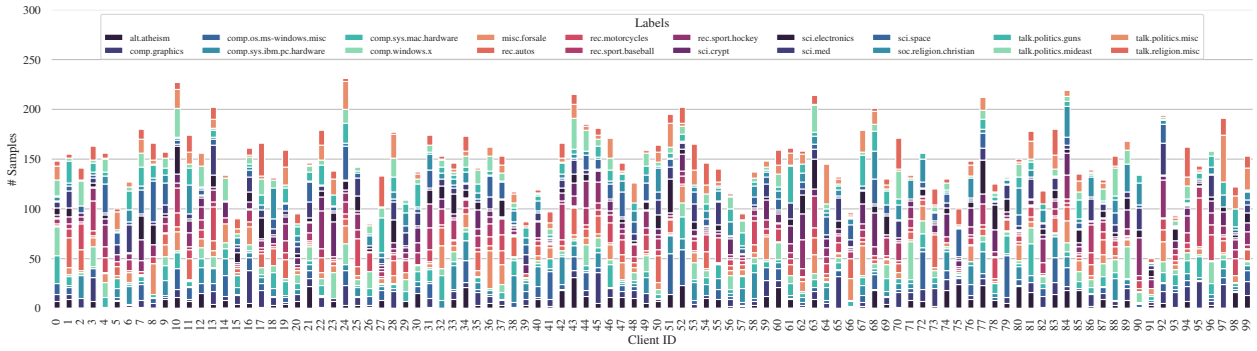


Figure 3: Visualization of client subsets for all of our experiments.

et al., 2023), which results in 52K trainable parameters (0.05% of total model parameters). This reduces the computational intensity of the task at hand and minimizes the communication load for the FL setup, as we must only communicate the trainable parameters. The BERT model is used to classify the emails into the 20 distinct categories in the dataset, which resembles a realistic task as it is frequently found in job application pre-screening applications, where the email bodies (input data) often contain sensitive and personal data.

FL configuration. We use the Federated Averaging (FedAvg) algorithm to facilitate all FL experiments (McMahan et al., 2017a) and train for 2000 aggregation rounds. We choose a participation rate of 10% for each aggregation round, i.e., $k = 10$ out of $K = 100$.

(ϵ, δ)-DP configuration. We employ sample-level (ϵ, δ)-DP for centralized learning, and for FL, we use user-level (ϵ, δ)-DP. Both methods provide the same privacy guarantees (Dwork & Roth, 2013). The parameterization for both is identical with $z = [0.0, 0.03, 0.1, 0.3, 0.4, 0.5, 0.6]$ and $\delta = \frac{1}{16{,}000}$, setting the data leakage risk to the inverse of the number of total training samples (Andrew et al., 2021; McMahan et al., 2017b). For the experiment with $z = [0.5; 0.6]$, we had to change the Learning Rate from 0.01 to 0.001.

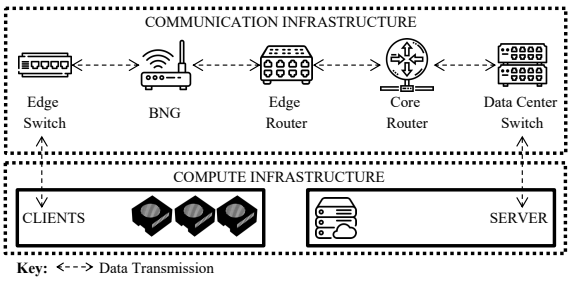


Figure 4: FL system design depicting the network topology for an aggregation round in FL between clients and the aggregation server. Every communication point consumes energy per transmitted bit, which must be accounted for.

Energy monitoring. We monitor our dedicated clients - NVIDIA Jetson AGX Orin - with 2Hz and measure their total energy consumption while participating in our FL setup. We also use a single Orin device for the centralized experiments for a fair comparison. For our cost estimations, we use the average price per kWh in the EU, $0.29\ \frac{€}{\text{kWh}}$ (Eurostat, 2023). The EU Commission produces quarterly reports on the electricity price trends (European Commission, 2023a). Directly proportional to the power consumption, we emit $252\ \frac{gCO_2e}{\text{kWh}}$ (European Environment Agency, 2023). Regarding communication energy, we assume the average communication route from a private household to a data center with $n_{as} = 1$, $n_e = 3$, $n_c = 5$, and $n_d = 2$ (cf. Equation (1)) (Jalali et al., 2014). For the energy consumption per transmitted bit per network hop, we adopt the values from Vishwanath et al. (2015); Jalali et al. (2014) (Table 3).

Table 3: Energy consumption per bit network communication for our holistic energy monitoring approach. Values are adopted from Vishwanath et al. (2015); Jalali et al. (2014).

Network Location	Device Name	Upload Cost (nJ/bit)	Download Cost (nJ/bit)
Edge Switch	Fast Ethernet Gateway	352	352
BNG	ADSL2+ Gateway (100 Mbit/s)	14809	2160
Edge Router	–	37	37
Core Router	–	12.6	12.6
Data Center Switch	Ethernet Switch	19.6	19.6

B.3. Hardware

We evaluate the training pipeline on a state-of-the-art embedded computing cluster with NVIDIA Jetson AGX Orin 64 GB devices (Orin), where each device has 12 ARMv8 CPU cores, an integrated GPU with 2048 CUDA cores, and 64 Tensor cores. The CPU and GPU share 64 GB of unified memory. The network interconnect is 10 GBit/s per client. We monitor the system metrics with a sampling rate of 2 Hz, including energy consumption in Watt (W). We use a data center server as an FL server. The server has 112 CPU cores, 384 GB of memory, an NVIDIA A40 GPU, and a 40 GBit/s network interface.

C. Algorithmic Cost Analysis for Private and Secure Computing Techniques in FL

In this section, we outline how we identified the algorithmic costs of state-of-the-art secure and private computing techniques. We omit the algorithmic costs of FedAvg and focus only on the privacy overhead. We discuss (ϵ, δ)-DP as introduced by Andrew et al. (2021), SMPC as introduced by Bonawitz et al. (2017), and HEC as introduced by Jin et al. (2023).

C.1. (ϵ, δ)-Differential Privacy

The following algorithm (Algorithm 1) is taken verbatim from Andrew et al. (2021). For the client, the computational complexity $O(d)$ originates from adding ξ to each parameter of a model update as well as by computing Δ. The communication complexity is $O(1)$ as we need to communicate the standard deviation to parameterize ξ as well as the clipping threshold. The space complexity $O(d)$ originates from storing θ.

The server computational complexity $O(|K|)$ originates from computing $\tilde{b}^t$ and the communication complexity $O(|K|)$ as we only communicate constants between clients and the server. The space complexity $O(|K|)$ comes from storing b_i.

Algorithm 1 DPFedAvg-M with adaptive clipping

function Train(m, γ, η_c, η_s, η_C, z, σ_b, β)
  Initialize model θ^0, clipping bound C^0
  $z_\Delta \leftarrow \left(z^{-2} - (2\sigma_b)^{-2}\right)^{-\frac{1}{2}}$
  for each round $t = 0, 1, 2, \ldots$ **do**
    $\mathcal{Q}^t \leftarrow$ (sample m users uniformly)
    for each user $i \in \mathcal{Q}^t$ **in parallel do**
      $(\Delta_i^t, b_i^t) \leftarrow \text{FedAvg}(i, \theta^t, \eta_c, C^t)$
    end for
    $\sigma_\Delta \leftarrow z_\Delta C^t$
    $\tilde{\Delta}^t = \frac{1}{m}\left(\sum_{i\in\mathcal{Q}^t} \Delta_i^t + \mathcal{N}(0, I\sigma_\Delta^2)\right)$
    $\bar{\Delta}^t = \beta\bar{\Delta}^{t-1} + \tilde{\Delta}^t$
    $\theta^{t+1} \leftarrow \theta^t + \eta_s\bar{\Delta}^t$
    $\tilde{b}^t = \frac{1}{m}\left(\sum_{i\in\mathcal{Q}^t} b_i^t + \mathcal{N}(O, \sigma_b^2)\right)$
    $C^{t+1} \leftarrow C^t \cdot \exp\left(-\eta_C(\tilde{b}^t - \gamma)\right)$
  end for
end function

function FedAvg(i, θ^0, η, C)
  $\theta \leftarrow \theta^0$
  $\mathcal{G} \leftarrow$ (user i's local data split into batches)
  for batch $g \in \mathcal{G}$ **do**
    $\theta \leftarrow \theta - \eta\nabla\ell(\theta; g)$
  end for
  $\Delta \leftarrow \theta - \theta^0$
  $b \leftarrow \mathbb{I}_{||\Delta|| \le C}$
  $\Delta' \leftarrow \Delta \cdot \min\left(1, \frac{C}{||\Delta||}\right)$
  return (Δ', b)
end function

C.2. Secure Multi-Party Computation

The SecAgg algorithmic costs (Table 4) are taken from Bonawitz et al. (2017) Table 1. The naming convention has been adapted to our paper.

Table 4: SecAgg costs

computation	
User	$O(\|K\|^2 + d \cdot \|K\|)$
Server	$O(d \cdot \|K\|^2)$
communication	
User	$O(\|K\| + d)$
Server	$O(\|K\|^2 + d \cdot \|K\|)$
storage	
User	$O(\|K\| + d)$
Server	$O(\|K\|^2 + d)$

C.3. Homomorphic Encryption

The following algorithm (Algorithm 2) is taken verbatim from Jin et al. (2023). For the client, computational complexity $O(d)$ originates from encrypting and decrypting the model. The communication complexity $O(d)$ comes from communicating the aggregation mask once. The space complexity $O(d)$ is created by storing the aggregation mask.

The server computational complexity $O(|K| \times d)$ originates from the server-side model aggregation while the communication complexity $O(|K| \times d)$ comes from sending the encryption mask once. Storing the encryption mask on the server results in space complexity $O(d)$.

Algorithm 2 HE-Based Federated Aggregation

- $[\![\mathbf{W}]\!]$: the fully encrypted model | $[\mathbf{W}]$: the partially encrypted model;
- p: the ratio of parameters for selective encryption;
- b: (optional) differential privacy parameter.

`// Key Authority Generate Key`
$(pk, sk) \leftarrow HE.KeyGen(\lambda)$;
`// Local Sensitivity Map Calculation`
for *each client* $i \in [N]$ **do in parallel**
  $\mathbf{W}_i \leftarrow Init(\mathbf{W})$;
  $\mathbf{S}_i \leftarrow Sensitivity(\mathbf{W}, \mathcal{D}_i)$;
  $[\![\mathbf{S}_i]\!] \leftarrow Enc(pk, \mathbf{S}_i)$;
  Send $[\![\mathbf{S}_i]\!]$ to server;
end
`// Server Encryption Mask Aggregation`
$[\![\mathbf{M}]\!] \leftarrow Select(\sum_{i=1}^{N} \alpha_i [\![\mathbf{S}_i]\!], p)$;
`// Training`
for $t = 1, 2, \ldots, T$ **do**
  for *each client* $i \in [N]$ **do in parallel**
    if $t = 1$ **then**
      Receive $[\![\mathbf{M}]\!]$ from server;
      $\mathbf{M} \leftarrow HE.Dec(sk, [\![\mathbf{M}]\!])$;
    end
    if $t > 1$ **then**
      Receive $[\mathbf{W}_{\text{glob}}]$ from server;
      $\mathbf{W}_i \leftarrow HE.Dec(sk, \mathbf{M} \odot [\mathbf{W}_{\text{glob}}]) + (\mathbf{1} - \mathbf{M}) \odot [\mathbf{W}_{\text{glob}}]$;
    end
    $\mathbf{W}_i \leftarrow Train(\mathbf{W}_i, \mathcal{D}_i)$;
    `// Additional Differential Privacy`
    if *Add DP* **then**
      $\mathbf{W}_i \leftarrow \mathbf{W}_i + Noise(b)$;
    end
    $[\mathbf{W}_i] \leftarrow HE.Enc(pk, \mathbf{M} \odot \mathbf{W}_i) + (\mathbf{1} - \mathbf{M}) \odot \mathbf{W}_i$;
    Send $[\mathbf{W}_i]$ to server $\mathcal{S}$;
  end
  `// Server Model Aggregation`
  $[\mathbf{W}_{\text{glob}}] \leftarrow \sum_{i=1}^{N} \alpha_i [\![\mathbf{M} \odot \mathbf{W}_i]\!] + \sum_{i=1}^{N} \alpha_i ((\mathbf{1} - \mathbf{M}) \odot \mathbf{W}_i)$;
end

CC BY-NC-ND 4.0

Attribution-NonCommercial-NoDerivatives 4.0 International

Deed

Canonical URL : https://creativecommons.org/licenses/by-nc-nd/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes .

NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Appendix E

Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis

Printed with the permission of

Herbert Woisetschläger*, Simon Mertel*, Christoph Krönke, Ruben Mayer, and Hans-Arno Jacobsen. "Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis." In: *Second Workshop on Generative AI + Law 2024 in conjunction with ICML'24*. GenLaw'24. 2024. DOI: 10.48550/ARXIV.2407.08105. URL: https://blog.genlaw.org/pdfs/genlaw_icml2024/16.pdf

Federated Learning and AI Regulation in the European Union: Who is Responsible? – An Interdisciplinary Analysis

Herbert Woisetschläger [*1] **Simon Mertel** [*2] **Christoph Krönke** [3] **Ruben Mayer** [2] **Hans-Arno Jacobsen** [4]

Abstract

The European Union Artificial Intelligence Act mandates clear stakeholder responsibilities in developing and deploying machine learning applications to avoid substantial fines, prioritizing private and secure data processing with data remaining at its origin. Federated Learning (FL) enables the training of generative AI Models across data siloes, sharing only model parameters while improving data security. Since FL is a cooperative learning paradigm, clients and servers naturally share legal responsibility in the FL pipeline. Our work contributes to clarifying the roles of both parties, explains strategies for shifting responsibilities to the server operator, and points out open technical challenges that we must solve to improve FL's practical applicability under the EU AI Act.

1. Introduction

With the introduction of the European Union Artificial Intelligence Act (AI Act) (Council of the European Union, 2021) and other international regulations being on the horizon, e.g., in the United States (The White House, 2023) and Canada (House Of Commons of Canada, 2022), everyone concerned with the development and deployment of AI has to adapt to new game rules. This entails data governance, robustness against adversarial scenarios, and energy considerations (Woisetschläger et al., 2024a). The AI Act puts the *service provider* into the spotlight, who has to assume responsibility for model development and deployment within the meaning of Article 3. Especially regarding data governance, the AI Act instantiates extensive rules for high-risk and general-purpose AI applications (GPAI, Article 52) that cater to data privacy and system security. The majority of generative AI applications fall under the GPAI definition in Article 3.

Federated Learning (FL) presents a privacy-enhancing and data-protecting machine learning technique (McMahan et al., 2017) that has recently received increased attention for enabling access to data silos for generative AI applications (Woisetschläger et al., 2024b). In FL, a server operator provides an ML model sent to several clients and then trained on the clients' local data, which collaboratively train a global model via a central server, aggregating their local model updates. Private and secure computing techniques like Differential Privacy or Trusted Execution Environments help improve data privacy and system security (Bonawitz et al., 2017; Andrew et al., 2021). FL's data locality removes the key challenge of monitoring data lineage and simplifies accounting for user consent. Specifically, we study the FL workflow in alignment with related work (Li et al., 2020; Hard et al., 2018; McMahan et al., 2017) to touch up on the following:

Data Acquisition. The server operator can only employ a variety of client sampling strategies (Malinovsky et al., 2023; Wang & Ji, 2022; McMahan et al., 2017) for an FL training round, without the ability to directly investigate client data or process integrity.

Data Storage. Similarly, the clients decide how, where, and when to store data. This has implications on data availability, which directly touches upon the AI Act data governance requirements (Article 10)[1].

Data Preprocessing. While the server operator can provide instructions on how to preprocess data so that the data is compatible with the ML model, the clients have the freedom to run additional preprocessing steps.

[*]Equal contribution [1]School of Computation, Information and Technology, Technical University of Munich, Germany [2]Department of Computer Science, University of Bayreuth, Germany [3]Department of Law & Economics, University of Bayreuth, Germany [4]Department of Electrical and Computer Engineering, University of Toronto, Canada. Correspondence to: Herbert Woisetschläger <herbert.woisetschlaeger@tum.de>, Simon Mertel <simon.mertel@uni-bayreuth.de>, Christoph Krönke <christoph.kroenke@uni-bayreuth.de>, Ruben Mayer <ruben.mayer@uni-bayreuth.de>, Hans-Arno Jacobsen <jacobsen@eecg.toronto.edu>.

Presented at the GenLaw '24 Workshop at the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]In the following, the term Article refers to articles in the AI Act if not specified otherwise

Since the server operator has no direct data access, verifying data integrity before training is challenging. For FL applications, there are numerous approaches to improve data integrity (Sánchez Sánchez et al., 2024; Roy Chowdhury et al., 2022).

Model Aggregation. While acting as the FL training orchestrator, the server operator handles the model integrity control mechanism when aggregating model updates. Thus, FL appears to be a well-suited solution to open up data silos and provide access to additional data. This would significantly benefit the training or fine-tuning of generative models due to their sheer appetite for ever-increasing amounts of data (Zhou et al., 2023).

One can think that the server operator is automatically also the *service provider*. Yet, FL is a cooperative ML training technique where a central entity typically provides the ML model, and clients can decide when to participate and what data to use for training. As such, we see that the server (model) and the clients (data) control parts of the FL lifecycle, rendering them both legally responsible for their respective parts. Thus, this opens up the question:

> *Who is the service provider at what point in the FL workflow, and how can each party assume adequate responsibility?*

Our paper studies technical and legal requirements that need to be established so that the FL server operator can assume responsibility as a service provider. This requires future technical work on auditability, verifiability, integrity, and privacy. Further, we need to establish regulatory references for the terms and services of FL applications.

2. Technical Solutions Need to Focus on Transferring Responsibility to the Server Operator

For practical FL applications, the server operator must assume the role of the service provider by employing appropriate technical solutions.

When establishing an FL system that could potentially entail thousands of clients at a time, managing responsibilities is likely to become a key challenge. Thus, we require solutions that provide for auditability, verifiability, integrity, and privacy.

Auditability & verifiability. There is a natural trade-off between privacy and data audits. The core paradigm of FL is to not share data beyond a client's area of control. Data is strictly inaccessible for everybody but the data owner, and even in-house restricted to authorized personnel. Thus, we face a challenge when aiming to audit all steps that happen on a client device or data server. For instance, a work by (Liu et al., 2023) uses a Bayesian Nash equilibrium and a market mechanism to incentivize truthful client behavior, i.e., submission of useful model updates. While this approach significantly reduces the risk of adversarial attacks, it does not meet the requirements for auditing in the context of the AI Act, which are well-defined. Quintessentially, any data that is being captured, processed, and used in a training process must be evaluated for potential bias or adversarial information. To achieve this, numerous works combining FL with blockchain technology explore auditing the data processing steps and the training itself (Nguyen et al., 2021; Ma et al., 2020). What remains open is to develop solutions against data tampering.

Integrity & privacy. Particularly, we have to rethink the obligations of the provider concerning data integrity and protection (Articles 8–10), such that responsibility is transferred to the FL server. To account for the asymmetry of access and control-by-design in FL systems, we must develop data integrity measures that capture the nature of client data at the time of collection, while preprocessing the data, and immediately before starting the training process. Peer-based verification schemes of model updates are a promising direction to identify adversarial clients (Roy Chowdhury et al., 2022). Extending such schemes from client models to client data without infringing privacy would be interesting. At the same time, technical solutions must be in line with the requirements set out in the GDPR, which are not (necessarily) aligned with the concepts and rules of the AI Act.

3. Regulatory Implementations Need to Foster Integrity and Verifiability

We need FL server operators to assume full responsibility; clients are technically and legally obligated to comply.

Service Provider. The GDPR (Council of the European Union, 2016) defines the term *data controller*. Complementary, the AI Act defines the *service provider* of AI systems. For data protection assessment when processing personal information at first, we need to clarify who is the data controller responsible and accountable for each distinct phase of the data processing and must demonstrate compliance with the requirements of the GDPR (Article 5). The AI Act does not have a differentiated allocation of roles for separate processing phases and focuses on one central "provider" of a (compliant) AI system, defined in Article 3, with the obligations arising from Article 8.

While both the FL server and clients could be considered providers under the AI Act, since the AI Act (unlike the GDPR) focuses less on responsibilities for individual, definable data processing phases and more on secure system design as a whole, the provider concept has to be teleologically limited to the FL server. Thus, the server acts as the

fully responsible service provider under the AI Act (Article 8), especially concerning data governance (Article 10) and General-Purpose AI service (Article 52).

General Terms and Conditions for AI Systems. While in Article 4 of the GDPR, the controller is the person who, alone or jointly with others, decides on the purposes ("why") and means ("how") of the processing of personal data, the AI Act focuses on the (traditionally single and) central provider of an AI system. However, since clients are autonomously in control of their data while the server is in control of the model, we see an inconsistency between what is controllable by the service provider and what he is responsible for. To close this gap, we need a two-pronged approach – technical and legal ("how" & "why"). Responsibility in FL should depend on the server's physical, technical, and legal ability to influence decentralized model training and configuration. This poses challenges. Unlike Article 26 and 28 GDPR, Article 8 et seq. of the AI Act do not provide details on governance in networked processing environments like FL systems.

The data protection assessment of the FL lifecycle may be impacted if the FL server sets requirements for the clients, which may lead to the server being classified as the controller under the GDPR. At first glance and from a strictly technical perspective, both the FL server and the FL clients fall under the provider concept of Article 3 GDPR, yet a "joint providership" (based on "joint controllership" under the GDPR, Article 26) does not exist under the AI Act. In fulfillment of its obligations under the AI Act, in particular Article 10, the FL server can set far-reaching requirements for the FL clients concerning the training of models and handling of training data. These requirements could lead to the FL server being classified as a controller under data protection law within the means of Article 4 GDPR, while the FL client is classified as a mere processor within the scope of Article 28 GDPR.

Thus, a key legal instrument for ensuring compliance with the AI Act and GDPR (Articles 26 & 28) is likely the development of specific General Terms and Conditions binding for both FL server and clients. The server operator, as the service provider, has to oblige clients to provide sufficient reporting compliant with the AI Act. This can be supported by cryptographic tools that minimize the need for trust among entities (Nguyen et al., 2021).

4. Considerations on Major Federated Learning Architectures

Cross-silo FL may allow for more flexibility in system design and responsibility distribution between clients and server than cross-device FL.

While Section 2 and Section 3 are generally applicable to any FL application, there are two major system architectures that create further opportunities to organize responsibilities: cross-silo and cross-device training.

4.1. Cross-Device Federated Learning

Cross-device FL typically entails a large number of devices ($> 1,000$). In such a setup, FL clients are characterized by having a very small number of local data samples and little participation time in the federated training process (Hard et al., 2018). This is a major challenge regarding client accountability and, ultimately, becomes problematic when a client should assume responsibility as a service provider. Hence, for practical considerations, all responsibility has to be assumed by the server in the cross-device setting and there is practically no room for client-side responsibility and a strong need for tools and methods that allow the FL server to cover all compliance criteria. This implies that the runtime environment on clients must be as encapsulated as possible, coupled with strict terms of service agreements.

4.2. Cross-Silo Federated Learning

In contrast, in cross-silo settings, individual clients hold a significant amount of data and participate in multiple training rounds, and usually come with higher computational capabilities than in cross-device FL. Typically, cross-silo FL can involve large institutions such as hospitals (Huang et al., 2022), which themselves have a high commitment to regulatory compliance and take strong precautions regarding security and privacy protection. As such, it is an open research direction to explore the synergies between established institutional processes (e.g., medical record keeping) and the AI Act requirements (e.g., on data transparency). The terms and conditions must be balanced between ensuring appropriate regulatory compliance and practical utility such that clients are incentivized to participate in training, and we can assume partial responsibility on the side of clients. Such synergies could help better balance the service provider responsibilities and reduce costs for clients and the server, not only improving the economic viability of FL but also its ecological footprint.

5. Conclusion

We study the FL life-cycle responsibilities under the AI Act. We find client-side responsibility for numerous steps, which practically limits the applicability of FL to open up additional data silos that would benefit the training of foundation models. Yet, there are promising directions that deserve increased attention such that a server operator can become the *service provider* without clients being required to assume extensive liability. With this, one can drive the adoption of FL and help decrease data bias by directly re-

lying on user data. Further clarifying the outlined service provider question directly responds to the EU AI Office's call for contributions to help implement the AI Act (Nature, 2024).

Acknowledgements

This work is partially funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (Grant: DIK0446/01).

References

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34: 17455–17466, 2021.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17. ACM, October 2017. doi: 10.1145/3133956.3133982. URL http://dx.doi.org/10.1145/3133956.3133982.

Council of the European Union. General data protection regulation (GDPR), apr 2016. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679. Document 32016R0679.

Council of the European Union. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL - LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, apr 2021. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206. Document 52021PC0206.

Hard, A., Kiddon, C. M., Ramage, D., Beaufays, F., Eichner, H., Rao, K., Mathews, R., and Augenstein, S. Federated learning for mobile keyboard prediction, 2018. URL https://arxiv.org/abs/1811.03604.

House Of Commons of Canada. An Act to enact the Consumer Privacy Protection Act, the Personal Information and Data Protection Tribunal Act and the Artificial Intelligence and Data Act and to make consequential and related amendments to other Acts, 6 2022. URL https://www.parl.ca/DocumentViewer/en/44-1/bill/C-27/first-reading.

Huang, C., Huang, J., and Liu, X. Cross-silo federated learning: Challenges and opportunities, 2022. URL https://arxiv.org/abs/2206.12949.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, May 2020. ISSN 1558-0792. doi: 10.1109/msp.2020.2975749. URL http://dx.doi.org/10.1109/MSP.2020.2975749.

Liu, Y., Lou, R., and Wei, J. Auditing for federated learning: A model elicitation approach. In *The Fifth International Conference on Distributed Artificial Intelligence*, DAI '23. ACM, November 2023. doi: 10.1145/3627676.3627683. URL http://dx.doi.org/10.1145/3627676.3627683.

Ma, C., Li, J., Ding, M., Shi, L., Wang, T., Han, Z., and Poor, H. V. When federated learning meets blockchain: A new distributed learning paradigm, 2020. URL https://arxiv.org/abs/2009.09338.

Malinovsky, G., Horváth, S., Burlachenko, K. P., and Richtárik, P. Federated learning with regularized client participation. In *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. URL https://openreview.net/forum?id=6CDBpf7kNG.

McMahan, B., Moore, E., et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Nature. There are holes in Europe's AI Act — and researchers can help to fill them. *Nature*, 625(7994): 216–216, January 2024. ISSN 1476-4687. doi: 10.1038/d41586-024-00029-4. URL http://dx.doi.org/10.1038/d41586-024-00029-4.

Nguyen, D. C., Ding, M., Pham, Q.-V., Pathirana, P. N., Le, L. B., Seneviratne, A., Li, J., Niyato, D., and Poor, H. V. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(16):12806–12825, August 2021. ISSN 2372-2541. doi: 10.1109/jiot.2021.3072611. URL http://dx.doi.org/10.1109/JIOT.2021.3072611.

Roy Chowdhury, A., Guo, C., Jha, S., and van der Maaten, L. Eiffel: Ensuring integrity for federated learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22.

ACM, November 2022. doi: 10.1145/3548606.3560611. URL http://dx.doi.org/10.1145/3548606.3560611.

Sánchez Sánchez, P. M., Huertas Celdrán, A., Xie, N., Bovet, G., Martínez Pérez, G., and Stiller, B. Federatedtrust: A solution for trustworthy federated learning. *Future Generation Computer Systems*, 152:83–98, March 2024. ISSN 0167-739X. doi: 10.1016/j.future.2023.10.013. URL http://dx.doi.org/10.1016/j.future.2023.10.013.

The White House. Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence, 10 2023. URL https://www.whitehouse.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/.

Wang, S. and Ji, M. A unified analysis of federated learning with arbitrary client participation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=qSs7C7c4G8D.

Woisetschläger, H., Erben, A., Marino, B., Wang, S., Lane, N. D., Mayer, R., and Jacobsen, H.-A. Federated learning priorities under the european union artificial intelligence act, 2024a. URL https://arxiv.org/abs/2402.05968.

Woisetschläger, H., Isenko, A., Wang, S., Mayer, R., and Jacobsen, H.-A. A survey on efficient federated learning methods for foundation model training, 2024b. URL https://arxiv.org/abs/2401.04472.

Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., Peng, H., Li, J., Wu, J., Liu, Z., Xie, P., Xiong, C., Pei, J., Yu, P. S., and Sun, L. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, 2023. URL https://arxiv.org/abs/2302.09419.

CC BY-NC-ND 4.0

Attribution-NonCommercial-NoDerivatives 4.0 International

Deed

Canonical URL : https://creativecommons.org/licenses/by-nc-nd/4.0/

See the legal code

You are free to:

Share — copy and redistribute the material in any medium or format

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes .

NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.