MAR-FL: A Communication Efficient Peer-to-Peer Federated Learning System

Felix Mulitze
Technical University of Munich
f.mulitze@tum.de

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

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

Abstract

The convergence of next-generation wireless systems and distributed Machine Learning (ML) demands Federated Learning (FL) methods that remain efficient and robust with wireless connected peers and under network churn. Peer-to-peer (P2P) FL removes the bottleneck of a central coordinator, but existing approaches suffer from excessive communication complexity, limiting their scalability in practice. We introduce **MAR-FL**, a novel P2P FL system that leverages iterative group-based aggregation to substantially reduce communication overhead while retaining resilience to churn. MAR-FL achieves communication costs that scale as $\mathcal{O}(N \log N)$, contrasting with the $\mathcal{O}(N^2)$ complexity of previously existing baselines, and thereby maintains effectiveness especially as the number of peers in an aggregation round grows. The system is robust towards unreliable FL clients and can integrate private computing.

1 Introduction

The convergence of Artificial Intelligence (AI) and next-generation wireless networks is driving a fundamental transformation in how we approach distributed computing and collaborative learning. As 6G and WiFi 9 standardization efforts begin to shape the future of global communication infrastructure, the ability to leverage distributed computational resources across wireless networks becomes not just an opportunity but a necessity for realizing the vision of AI-native wireless systems. The rapid proliferation of data across distributed sources – from edge devices to base stations – has created unprecedented opportunities for Machine Learning (ML), yet accessing and utilizing these dispersed data repositories remains a fundamental challenge (Kairouz et al., 2021; Li et al., 2020). While centralized ML has driven remarkable advances in AI, it faces increasing limitations: data privacy regulations restrict data movement across organizational and geographical boundaries, bandwidth constraints in wireless environments make centralized data aggregation impractical, and the concentration of computational resources in large-scale data centers creates geographical and economic disparities in AI development capabilities (Kairouz et al., 2021; Zhang et al., 2020). Federated Learning (FL) has emerged as a useful paradigm that enables collaborative model training over wide-area networks while keeping data localized, effectively tapping into vast data silos that would otherwise remain inaccessible for AI development (McMahan et al., 2017; Kairouz et al., 2021; Li et al., 2020; Zhang et al., 2020).

The promise of FL extends beyond privacy preservation to address a critical infrastructure challenge particularly relevant to next-generation wireless networks: the democratization of AI training capabilities at the network edge. Current AI development is increasingly dominated by regions with

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

Table 1: Related work overview. Our system is the first to deliver communication-efficient end-to-end P2P FL.

Approach	Allows partial communication	Provides global aggregation	No sparsification	Peer dropout tolerance	Enables private training	Source
RDFL	–	✓	✓	–	–	Hu et al. (2020)
SAPS	✓	–	–	–	–	Tang et al. (2020)
BrainTorrent	✓	–	✓	✓	–	Roy et al. (2019)
MAR-FL (ours)	✓	✓	✓	✓	✓	*This paper*

access to massive, centralized computing infrastructure and abundant power resources. However, many regions – particularly in Europe – face significant constraints in building comparable large-scale AI data centers due to power grid limitations, environmental regulations, and infrastructure costs (EU Agency for the Cooperation of Energy Regulators, 2024). This disparity threatens to create a widening gap in AI capabilities between regions with different infrastructure capacities.

In the context of emerging wireless systems, where edge intelligence and distributed processing are fundamental design principles, FL offers a compelling alternative by enabling the orchestration of scattered computational resources – from mobile devices to small cell base stations – into a collective training infrastructure without requiring massive capital investments or power concentration (McMahan et al., 2017; Kairouz et al., 2021; Li et al., 2020).

Peer-to-peer (P2P) FL systems represent the natural evolution of this distributed paradigm, aligning perfectly with the vision of AI-native wireless networks where intelligence is embedded throughout the network and does not require a centralized coordination server. By eliminating the central coordinator, P2P FL removes the communication and memory bottleneck of client-server FL – where the server must coordinate massive numbers of unreliable devices in cross-device settings and shuttle large models in cross-silo settings – thereby throttling scalability and slowing training (Alqahtani and Demirbas, 2019; Huang et al., 2023). It also eliminates the single point of failure: progress no longer hinges on server-side compute or networking capacity, which can otherwise jeopardize training (Tang et al., 2020). Freed from these constraints, P2P FL can harness available computational resources wherever they exist—from idle GPUs in edge servers to distributed computing nodes in radio access networks—creating a resilient, fault-tolerant training infrastructure that adapts to the dynamic resource availability inherent in wireless environments. This decentralized approach is particularly valuable in scenarios where network topology changes rapidly, devices join and leave unpredictably, and no single entity can or should control the training process (e.g., multi-operator collaborations or community-driven deployments). These challenges create a fundamental research question: *Can we design a communication-efficient P2P FL system that maintains training quality while handling the high peer churn rates and sudden training dropouts characteristic of wireless environments?*

Contributions. In this paper, we present **Moshpit All-Reduce FL (MAR-FL)**, a novel P2P FL system that builds on dynamic iterative group formation to significantly improve communication efficiency and tolerance towards unexpected peer churn. MAR-FL allows scalable decentralized learning by reducing the overall communication load and the required number of interactions between peers. Our system incorporates Knowledge Distillation (KD) to boost training performance and supports optional Differential Privacy (DP) to mitigate remaining risks of private information leakage. We conduct a comprehensive experimental evaluation that compares MAR-FL against client–server FL and P2P FL techniques, assessing communication efficiency, scalability, and robustness to network churn.

Related work. Despite compelling advantages over client-server FL, existing P2P FL systems face severe practical limitations preventing deployment in bandwidth-limited wireless networks (Table 1). The Galaxy Federated Learning system's Ring Decentralized FL (RDFL) (Hu et al., 2020) incurs communication costs orders of magnitude higher than centralized FedAvg, making it economically infeasible for wireless environments. Moreover, RDFL's closed ring topology cannot tolerate the dynamic participation and node failures characteristic of wireless networks due to mobility, channel fading, or varying signal conditions. Sparsification and Adaptive Peer Selection (SAPS) (Tang et al., 2020) improves communication efficiency through model sparsification and single high-throughput peer exchanges per round, but spreads information only locally with-

out synchronized global aggregation, slowing convergence and making progress sensitive to churn. BrainTorrent (Roy et al., 2019) provides serverless P2P flexibility through dynamic model fetching and merging, but relies on uncoordinated gossip-based learning that suffers from inefficient global information propagation and vulnerability to node churn.

Structure. We introduce our new MAR-FL system in Section 2 and evaluate it in Section 3. We conclude in Section 4.

2 MAR-FL: Communication-efficient P2P Federated Learning

The overall objective of our P2P FL system is to reduce the communicational effort required to obtain globally averaged models, while retaining resilience to real-world network churn. Consequently, we deploy Moshpit All-Reduce (MAR) as fully decentralized aggregation mechanism.

2.1 Problem Formulation

We consider a P2P FL setting with N peers, each holding a private local dataset $\mathcal{D}_i$, which may be heterogeneous and non-i.i.d. across peers. Training proceeds over T iterations; in each iteration, peers perform local updates and exchange models over bandwidth-limited wireless links to conduct global aggregation. The system thereby faces the central FL challenge of communication costs: due to wireless links and connections operating at lower rates than intra- or inter-datacenter links, communication is costly and often by orders of magnitude slower than local computation (Kairouz et al., 2021; Li et al., 2020). Consequently, our objective is to minimize the communication cost of P2P FL systems.

2.2 Proposed System

Integrating MAR into FL. For global model aggregation in fully decentralized FL, we adopt the idea of Moshpit SGD (Ryabinin et al., 2021), where peers conduct MAR to dynamically form small independent groups and repeat this group matchmaking across multiple rounds until local information from all peers has propagated through the network. This procedure has two main benefits: global model averaging can be achieved without all-to-all communication, and peer dropouts only affect a single group (i.e., a very restricted number of peers). The overall MAR-FL training process (Algorithm 1) starts for every peer $i \in [N]$ with the same randomly initialized model θ^0 and momentum vector m^0, where N denotes the total number of peers and T the total number of FL iterations. In each FL iteration $t \in \{1, \dots, T\}$, every participating peer $i \in \mathcal{U}_t$, where $\mathcal{U}_t \subseteq [N]$, performs a local Momentum-SGD update on B mini-batches of its local data D_i, using stepsize η and momentum μ. The update follows the damped momentum update proposed by Reddi et al. (2020) and yields a local peer state (θ_i^t, m_i^t). A set of aggregation peers $\mathcal{A}_t \subseteq [N]$ then performs MAR on its aggregation state set $\mathcal{S}_t$, where $\mathcal{S}_t := \{(j, \theta_j^t, m_j^t) \mid j \in \mathcal{A}_t\}$, to obtain a globally averaged state (θ^t, m^t). This is done in multiple group formation rounds $g \in G^t$ per FL iteration t. KD is integrated if the $\mathrm{use_{kd}}$ flag is set. After T iterations, each peer holds the final collaboratively trained global model θ^T.

Algorithm 1: MAR-FL (for i-th peer)

Input: $\theta^0, m^0, \eta, \mu, D_i, B, T, N, \mathrm{use_{kd}}$
for $t = 1, 2, \dots, T$ **do**
  if $i \in \mathcal{U}_t$ **then**
    $(\theta_i^t, m_i^t) \leftarrow$ `Momentum-SGD`$(\theta^{t-1}, m^{t-1}, D_i, B, \eta, \mu)$
  else
    $(\theta_i^t, m_i^t) \leftarrow (\theta^{t-1}, m^{t-1})$
  if $i \in \mathcal{A}_t$ **then**
    $\mathcal{S}_t := \{(j, \theta_j^t, m_j^t) \mid j \in \mathcal{A}_t\}$
    if $\mathrm{use_{kd}}$ **then**
      $(\theta_i^t, m_i^t) \leftarrow$ `Moshpit-KD`$(\mathcal{S}_t)$
    $(\theta^t, m^t) \leftarrow$ `Moshpit-AR`$(\mathcal{S}_t)$
return θ^T

Coordinating FL peers. Synchronization of peers during group formation is coordinated through Distributed Hash Tables (DHT). Our system thereby relies on a Hivemind Kademlia DHT solely for lightweight coordination – barriers and group-formation metadata – while model and momentum weights never traverse the DHT. A single DHT get/store involves at most $\mathcal{O}(\log N)$ hops. In our implementation, coordination occasionally scans peer announcements (issuing $\mathcal{O}(N)$ look-ups), so the control-plane cost per round is $\mathcal{O}(N \log N)$ and remains negligible compared to model-exchange traffic. To assemble into groups, each peer manages its own group key and forms groups with peers sharing the same key value in the DHT. To avoid redundant information exchange in consecutive MAR rounds, peers are prevented from revisiting one another within a single FL iteration by group

key initialization and updates that leverage their chunk indices from $d-1$ previous MAR rounds. We therefore adopt techniques proposed by Ryabinin et al. (2021). In an optimal MAR-FL setup, exact global averaging can be achieved after d rounds of MAR when the group size is M and the group key dimension is d, so that the total number of peers N satisfies $N = M^d$. With fixed MAR group size M, each round makes a peer talk to at most $(M-1)$ others, and achieving (near-)global averaging needs $G \approx \lceil \log_M N \rceil$ rounds (exactly $G{=}d$ when $N{=}M^d$). Hence, each peer performs $\mathcal{O}(\log_M N)$ exchanges per iteration and, over all peers, the system incurs $\mathcal{O}(N \log_M N) = \mathcal{O}(N \log N)$ communication per iteration, versus $\mathcal{O}(N^2)$ for P2P FL systems using all-to-all communication.

Concept of KD. Our MAR-FL system allows the integration of KD to accelerate model convergence. Let $C_g \subseteq \mathcal{A}_t$ be the candidate teacher peers in MKD round g with local models $\{\theta_c^{g-1}\}_{c \in C_g}$, where $\mathcal{A}_t$ refers to Algorithm 1. Candidate teachers are collected using the same procedure MAR uses for global model averaging; hence, we call this mechanism Moshpit-KD (MKD). The MKD process of an entire FL iteration proceeds over multiple MKD rounds $g \in \{1, \ldots, G\}$, where each round g includes group formation and candidate teacher collection followed by the actual distilling of knowledge. To balance model utility and communication overhead, we use MKD only in the first K FL iterations. Algorithm 2 illustrates MKD round g in FL iteration $t \in \{1, \ldots, K\}$, where $K \leq T$ denotes the number of FL iterations in which we actually apply MKD, with T being the total number of MAR-FL iterations in Algorithm 1. To account for data heterogeneity in FL (Shao et al., 2024), MKD selects a subset of top-ℓ teachers $\mathcal{C}_g^{\text{top}} \subseteq \mathcal{C}_g$ with the lowest Kullback–Leibler (KL) divergence, where ρ_ℓ is the selection ratio (details in Appendix A.1). Student i then distills knowledge from these selected teachers: over E local epochs, starting from the previous MKD round's state $(\theta_i^{g-1}, m_i^{g-1})$, the student updates on each available local mini-batch $b \in \mathcal{B}$ by computing a student loss L and applying Momentum-SGD (Reddi et al., 2020) with learning rate η and momentum μ to eventually obtain an updated state (θ_i^g, m_i^g). In MKD round $g = 1$, the previous state (θ_i^0, m_i^0) refers to the student's state before any MKD is applied (i.e., after local model update). The student loss L aligns to the loss term proposed by Hinton et al. (2015): L is the weighted sum of the KL divergence D_{KL} between softened probability distributions over teacher-ensemble and student classes, rescaled by the squared temperature τ^2, and a CE term L_{CE} on hard labels y_b. Averaged teacher-ensemble logits are hereby denoted as $\bar{z}_b$ and student logits as $\hat{s}_b$. As we use MKD only in the first K FL iterations, we facilitate a gradual transition from the use of MKD to its complete omission by linearly reducing the weighting λ of the KL term L_{KL}.

Algorithm 2: Moshpit-KD (for i-th peer in MKD round g of FL iteration t)

Input: $\theta_i^{g-1}, m_i^{g-1}, \mathcal{C}_g, \{\theta_c^{g-1}\}_{c \in C_g}, \mathcal{B}, E, \eta, \mu, \tau, \rho_\ell, K$
Output: θ_i^g, m_i^g

$(\mathcal{C}_g^{\text{top}}, \ell, \{z_b^{(c)}\}_{b \in \mathcal{B},\, c \in \mathcal{C}_g^{\text{top}}}) \leftarrow$ `TeacherSel`$(\theta_i^{g-1}, m_i^{g-1}, \mathcal{C}_g, \{\theta_c^{g-1}\}_{c \in C_g}, \mathcal{B}, \tau, \rho_\ell)$
for $b \in \mathcal{B}$ **do**
  $\bar{z}_b \leftarrow \frac{1}{\ell} \sum_{c \in \mathcal{C}_g^{\text{top}}} z_b^{(c)}$.
$(\theta_i^g, m_i^g) \leftarrow (\theta_i^{g-1}, m_i^{g-1})$
for $e = 1, 2, \ldots, E$ **do**
  for $b \in \mathcal{B}$ **do**
    $\hat{s}_b \leftarrow f_{\theta_i^g}(x_b)$
    $L_{\text{KL}} \leftarrow \tau^2 \cdot D_{\text{KL}}(\text{softmax}(\bar{z}_b/\tau) \,\|\, \text{softmax}(\hat{s}_b/\tau))$
    $L_{\text{CE}} \leftarrow \text{CE}(y_b, \text{softmax}(\hat{s}_b))$
    $\lambda \leftarrow \max(0,\, 1 - \frac{t-1}{K})$
    $L \leftarrow \lambda\, L_{\text{KL}} + (1-\lambda)\, L_{\text{CE}}$
    $m_i^g \leftarrow \mu\, m_i^g + (1-\mu)\, \nabla_{\theta_i^g} L$
    $\theta_i^g \leftarrow \theta_i^g - \eta\, m_i^g$
return θ_i^g, m_i^g

Privacy considerations. To allow privacy preserving training, we adapt the DP-FedAvg with adaptive clipping (Andrew et al., 2021) to fit our serverless P2P system (Algorithm 4, see Appendix A.2). In each FL iteration, every peer first computes the difference between its current local model and the previously aggregated global model. This update is then clipped to an adaptive bound and perturbed with Gaussian noise. The privatized update is used to compute a DP-safe local model and peers run MAR. After the final round of MAR, the clipping bound is updated to track a globally averaged clipping rate. This procedure fully decentralizes DP with adaptive clipping and renders it ready to use with MAR-FL: privacy loss accrues entirely from local computations, while MAR merely averages privatized models across groups.

2.3 Convergence Analysis

The convergence of MAR-FL follows from the model mixing dynamics of MAR, analyzed by Ryabinin et al. (2021). In the optimal case where the total number of peers N forms a perfect d-dimensional grid $N = M^d$ and there are no peer dropouts, MAR computes the exact global average after exactly d rounds of communication – i.e., within a single FL iteration t when that

iteration schedules d MAR rounds. For general settings, MAR exhibits exponential convergence to the global average $\bar{\theta}$. Specifically, if peers are randomly partitioned each iteration into r groups that average locally, the expected average distortion after T averaging iterations satisfies:

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\|\theta_i^T - \bar{\theta}\|^2\right] = \left(\frac{r-1}{N} + \frac{r}{N^2}\right)^T \frac{1}{N}\sum_{i=1}^{N}\|\theta_i - \bar{\theta}\|^2. \tag{1}$$

While this rate is derived for a simplified random-grouping model, our system's MAR implementation avoids revisiting peers via deterministic key updates, which in practice accelerates mixing relative to purely random grouping. Crucially, the bound is independent of the spectral properties of the communication graph, avoiding the scaling limits typical of gossip-based decentralized FL.

3 Experiments

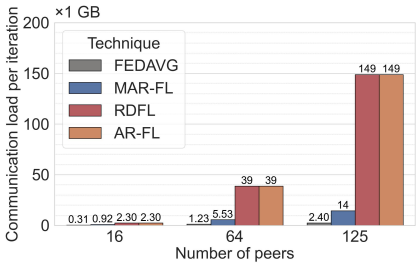
This section presents our experimental setup and evaluates results in detail, while emphasizing communication cost, scalability, robustness and trade-offs concerning model utility.

3.1 Setup

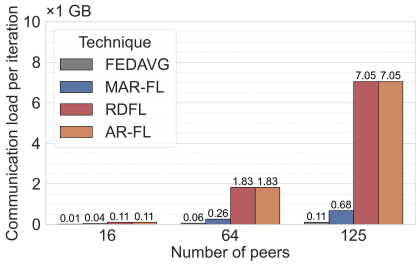
In the following we delineate ML datasets and models, reference baselines, and parametrization used to evaluate and contextualize MAR-FL. Underlying objectives are described. We use a simulation environment for all of our experiments. Due to constraints of our simulation environment, model evaluation is conducted every fifth FL iteration. We simulate all experiments on a single node with 4×H100 GPUs, 768 GB of memory, and 96 CPU cores. Our code is publicly available.[1] Additional details on the experimental setup can be found in the appendix.

Datasets and models. We evaluate MAR-FL on two widely used ML datasets, namely MNIST (LeCun et al., 2010) and 20 Newsgroups (20NG) (Lang, 1995). For MNIST, we use a CNN-based architecture and for 20NG we use a frozen DistilBERT model (Sanh et al., 2019) with a classification head. We employ a Latent Dirichlet Allocation ($\alpha = 1.0$) to create non-i.i.d. subsets for 16, 64, and 125 FL peers. If not specified otherwise, experiments run on 125 FL peers. Per aggregation round, each peer trains on 64 and 16 samples for MNIST and 20NG, respectively.

FL baselines. We directly compare MAR-FL to the client-server FedAvg standard and established P2P FL techniques, specifically RDFL (ring all-reduce), which is at the center of the Galaxy Federated Learning framework.[2] We further evaluate MAR-FL against a naïve all-to-all All-Reduce FL algorithm (AR-FL), where all peers communicate with each other. Even though we discuss BrainTorrent and SAPS in our related work section, their limitations regarding communication efficiency make practical deployments prohibitively expensive, which is why we omit the two techniques as baselines.



(a) MNIST



(b) 20NG

Figure 1: Performance gap evaluation. MAR-FL improves communication efficiency by up to $10\times$ compared to existing P2P FL baselines.

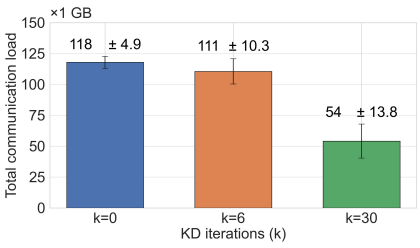


Figure 2: With MKD, the communication load of systems using MAR-FL is further reduced, as we require over $2\times$ less communication to reach 50% accuracy. Plot shows results on 20NG. Results on MNIST are available in the appendix.

[1] Github: `https://github.com/felix-fjm/mar-fl`

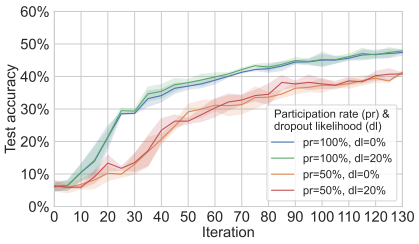
[2] We do not compare against Galaxy Federated Learning as a whole since the framework largely depends on a distributed ledger/blockchain for training verification. Verification is beyond the scope of our work.

Parametrization of MAR-FL. We use exact aggregation of peer groups, if not specified otherwise. For evaluating the compound benefits of MAR-FL and KD, we use a teacher selection ratio $\rho_\ell = 0.4$ (Hu et al., 2020), student loss temperature $\tau = 3.0$ (Hinton et al., 2015) and one epoch. Parameter choices for adaptive DP align to Andrew et al. (2021) and are listed in Appendix A.2.

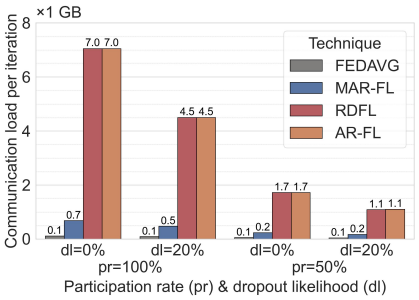
Local model aggregation. For peer-side local aggregation, we use SGD with momentum (Reddi et al., 2020), set the learning rate to $\eta = 0.1$ and the momentum to $\mu = 0.9$. Across techniques, we use full peer participation if not specified otherwise (typical setup for cross-silo FL applications).

Partial participation and network churn. To assess the effect of partial participation and network churn, we vary participation rates and dropout likelihoods. Participation rates control how many peers participate in an entire FL iteration consisting of local updates and global aggregation, while dropout likelihoods simulate unreliable peer connectivity (i.e., peer has conducted local update but does not participate in global aggregation).

Privacy. To investigate privacy-preserving training and its effect on model utility, we vary the noise multiplier to control the extent of privatization. The peer sampling rate, where lower values reduce the privacy loss, is fixed at 100%. Results on scalability and partial participation will reveal whether our system can leverage this rate to enhance privacy without degrading training performance.



(a) Model Performance – MAR-FL – 20NG



(b) Communication Cost – 20NG

Figure 3: MAR-FL is affected by partial participation but resilient towards sudden dropouts.

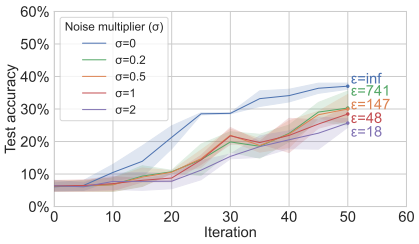


Figure 4: MAR-FL is compatible with DP and exhibits the same performance characteristics as FedAvg. Plot shows results on 20NG. Results on MNIST are available in the appendix.

3.2 Results

Communication efficiency and scalability. Across both ML tasks (MNIST and 20NG), MAR-FL matches the training performance of all three baselines (see Appendix C.1). This parity is expected because, with suitable MAR parameters, each iteration of MAR-FL attains an exact global average (e.g., group size 5 and 3 MAR rounds for 125 peers: $125 = 5^3$). While obtaining identical model utility, MAR-FL requires far less communication per iteration, up to $10\times$ less communication than RDFL or AR-FL. The communication complexity of MAR-FL, $\mathcal{O}(N \log N)$, yields stronger performance as systems scale (Figure 1). In contrast, RDFL and AR-FL exhibit a complexity of $\mathcal{O}(N^2)$.

Improving communication efficiency with MKD. MAR-FL achieves substantially higher communication efficiency than our P2P FL baselines, narrowing the gap to the client-server FedAvg standard. To improve communication efficiency even further, MKD can be used. MKD accelerates model convergence so that a target accuracy can be reached with less total communication (Figure 2), although increasing the per-iteration load. The trade-off between communication costs and model utility can be controlled by the number of KD iterations.

Partial participation and network churn. Partial participation leads to a substantial degradation of MAR-FL's training performance, while configured network churn and unreliable connectivity do not cause additional accuracy drops (Figure 3); our three baselines show the same pattern. While the training performance of all three P2P techniques is equally affected by these real-world system disturbances, MAR-FL consistently preserves its net benefit over all baselines in communication efficiency, providing evidence for the enhanced practicality of our system. Even with 50% participation and 20% dropout likelihood, RDFL and AR-FL require more than $5\times$ the communication of MAR-FL to reach the same model utility. The robustness towards unreliable connectivity (i.e.,

peer has conducted local update but does not participate in global aggregation) can be attributed to the fact that averaging incomplete global models over multiple FL iterations eventually converges to almost exact global averages. In Appendix C.2, where we provide further results on partial participation, we outline how this phenomenon can be exploited to increase the communication efficiency of MAR-FL. The appendix also includes additional results for FedAvg, RDFL, and AR-FL.

Differentially private training. When conducting DP-safe model aggregation in MAR-FL, increasing the strength of DP by raising the noise multiplier σ reduces the privacy loss ε but eventually degrades model utility (Figure 4). Since our observations align with the effect of DP on standard FedAvg (Andrew et al., 2021; Wei et al., 2020), this confirms that DP is readily supported within our fully decentralized system. We emphasize that the privacy loss ε can be substantially reduced by decreasing the peer-sampling rate (i.e., partial participation in local updates) (Wei et al., 2020; Mironov, 2017), so that our communication-efficient and scalable MAR-FL system provides a foundation for comprehensive privacy preservation.

4 Conclusions

We introduce MAR-FL, a P2P FL system that leverages iterative group-based aggregation to substantially reduce communication costs compared to existing P2P FL techniques. On 125 peers, MAR-FL requires about $10\times$ less total communication than RDFL or AR-FL while achieving identical model utility. MAR-FL scales with $\mathcal{O}(N \log N)$, enabling efficient training as the number of peers grows. Moreover, our system remains robust under unreliable peers, supports KD to further reduce communication, and integrates DP. Our findings position MAR-FL as a practical foundation for scalable, communication-efficient P2P FL in next-generation wireless settings.

Limitations. While MAR-FL improves the communication efficiency of P2P FL, there is still a performance gap towards client-server FL. Such performance penalties cause higher operating costs, which typically hinders practical adoption. We offer a starting point for using DP with MAR-FL but analyzing the impact of group-based aggregation in combination with momentum on DP dynamics remains open.

Future work. Future work includes a thorough analysis of partial participation and network churn – bringing our system even closer to real-world applicability. Exploring approximate aggregation and adaptive group-based information propagation could further improve communication efficiency and narrow the gap to client–server FedAvg. Experimental evaluations of integrating DP into MAR-FL should exploit our system's scalability to compress peer-sampling rates; maintaining model utility while reducing the privacy loss. Finally, we emphasize the importance of P2P FL: by omitting a centralized server, MAR-FL avoids communication and memory bottlenecks inherent in client–server FL and moves FL closer to its promising applications.

References

S. Alqahtani and M. Demirbas. Performance analysis and comparison of distributed machine learning systems. *arXiv preprint arXiv:1909.02061*, 2019.

G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems*, volume 34, pages 17455–17466, 2021.

EU Agency for the Cooperation of Energy Regulators. Transmission capacities for cross-zonal trade of electricity and congestion management in the eu, jul 2024. URL `https://www.acer.europa.eu/monitoring/MMR/crosszonal_electricity_trade_capacities_2024`.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Y. Hu, Y. Zhou, J. Xiao, and C. Wu. Gfl: A decentralized federated learning framework based on blockchain. *arXiv preprint arXiv:2010.10996*, 2020.

C. Huang, M. Tang, Q. Ma, J. Huang, and X. Liu. Promoting collaboration in cross-silo federated learning: Challenges and opportunities. *IEEE Communications Magazine*, 62(4):82–88, 2023.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, S. Chakraborty, D. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

K. Lang. NewsWeeder: Learning to Filter Netnews. In A. Prieditis and S. Russell, editors, *Machine Learning Proceedings 1995*, pages 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.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*, 2019.

M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, and G. Pekhimenko. Moshpit sgd: Communication-efficient decentralized training on heterogeneous unreliable devices. In *Advances in Neural Information Processing Systems*, volume 34, pages 18195–18211, 2021.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT: a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

J. Shao, F. Wu, and J. Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nature Communications*, 15(1):349, 2024.

Z. Tang, S. Shi, and X. Chu. Communication-efficient decentralized learning with sparsification and adaptive peer selection. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1207–1208. IEEE, 2020.

K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, and H. V. Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 493–506, 2020.

Appendix

A Additional Details on the MAR-FL Method

A.1 Details on MKD

Teacher selection in MKD. In MKD round g, candidate teachers $\mathcal{C}_g$ depict a collected subset of participating aggregation peers $\mathcal{A}_t$ of FL iteration t from Algorithm 1. From these candidate teachers, a top-ℓ ratio is selected for actual distilling of knowledge. A peer i thereby selects ℓ teachers $\mathcal{C}_g^{\text{top}} \subseteq \mathcal{C}_g$ which yield the ρ_ℓ smallest KL divergence when for each candidate teacher model $\{\theta_c^{g-1}\}_{c\in C_g}$ comparing the softened output distribution $\mathrm{softmax}(z_b^{(c)}/\tau)$ with its own softened output distribution $\mathrm{softmax}(s_b/\tau)$. The softening of output distributions is conducted by normalizing the logits $z_b^{(c)}$ and s_b with a temperature τ. Student logits s_b are obtained by passing local mini-batches $\mathcal{B}$ through the student model θ_i^{g-1}, while teacher logits $z_b^{(c)}$ are obtained by passing local mini-batches $\mathcal{B}$ through a candidate teacher model θ_c^{g-1}. We use the KL-based rating of collected peer models to account for non-i.i.d. data in FL. Shao et al. (2024) emphasize that non-i.i.d. data distributions depict a crucial challenge for KD in FL, because local models cannot produce meaningful predictions on data outside of their own distributions. Even softening of output distributions is not solving this issue, as ensembles of inconsistent local predictions still exhibit high entropy, which leads to distilling ambiguous and misleading knowledge.

Algorithm 3: Teacher selection in MKD (for i-th peer in MKD round g of FL iteration t)

Input: $\theta_i^{g-1}, m_i^{g-1}, \mathcal{C}_g, \{\theta_c^{g-1}\}_{c\in C_g}, \mathcal{B}, \tau, \rho_\ell$
Output: $\mathcal{C}_g^{\text{top}}, \ell, \{ z_b^{(c)} \}_{b\in\mathcal{B},\, c\in\mathcal{C}_g^{\text{top}}}$

for $b \in \mathcal{B}$ **do**
  $s_b \leftarrow f_{\theta_i^{g-1}}(x_b)$
for $c \in \mathcal{C}_g$ **do**
  for $b \in \mathcal{B}$ **do**
    $z_b^{(c)} \leftarrow f_{\theta_c^{g-1}}(x_b)$
  $S(c) \leftarrow \sum_{b\in\mathcal{B}} D_{\mathrm{KL}}(\mathrm{softmax}(z_b^{(c)}/\tau) \,\|\, \mathrm{softmax}(s_b/\tau))$.
$\ell \leftarrow \max\{1, \lfloor \rho_\ell\, |\mathcal{C}_g| \rfloor\}$.
$\mathcal{C}_g^{\text{top}} \leftarrow \ell$ teachers with smallest $S(c)$.
return $C_g^{\text{top}}, \ell, \{ z_b^{(c)} \}_{b\in\mathcal{B},\, c\in C_g^{\text{top}}}$

Deriving the MKD student loss. In Algorithm 2 we denote the student loss term L as weighted sum of the KL divergence D_{KL} between softened probability distributions over teacher-ensemble and student classes, rescaled by the squared temperature τ^2, and a CE term L_{CE} on hard labels y_b. This student loss can be derived from the student loss proposed by Hinton et al. (2015). Let $p_z = \mathrm{softmax}(z/\tau)$ be the teacher distribution and $p_s = \mathrm{softmax}(s/\tau)$ the student distribution at the same temperature τ, where teacher logits are denoted by z and student logits by s. Higher values of τ shrink differences between logits so that the distribution is softer, which can reveal relative similarities among classes (i.e., dark knowledge). Hinton et al. (2015) train the student with a weighted sum of CE to soft targets at $\tau > 1$ and CE to the hard labels y at $\tau = 1$, scaling the soft-target gradients by τ^2:

$$L_{\text{Hinton}} = (1-\alpha)\ \mathrm{CE}(y, \mathrm{softmax}(s)) + \alpha\tau^2\ \mathrm{CE}(p_z, p_s). \tag{2}$$

Starting from this two-term objective, one can use the identity

$$\mathrm{CE}(p_z, p_s) = H(p_z) + D_{\mathrm{KL}}(p_z \parallel p_s). \tag{3}$$

and note that $H(p_z)$ is constant with respect to the student. Dropping that constant and absorbing $\alpha\tau^2$ into the KL weight gives

$$L \equiv (1-\alpha)\ \mathrm{CE}(y, \mathrm{softmax}(s)) + \alpha\tau^2\, D_{\mathrm{KL}}(p_z \parallel p_s), \tag{4}$$

which is the student loss term L used in our MKD approach when $\alpha = \max\big(0,\, 1 - \frac{t-1}{K}\big)$.

A.2 Fully Decentralized DP

Andrew et al. (2021) propose DP-FedAvg with adaptive gradient clipping for client-server FL, in which a central server mediates the DP-safe model aggregation. We adapt this approach to fit our serverless P2P system. Our system's DP-safe model aggregation illustrated in Algorithm 4 corresponds to MAR in Algorithm 1 when DP is activated. For simplicity, the local pre-aggregation state

$(\theta_i^t,\, m_i^t)$, the peer's last obtained global model $\bar{\theta}_i^{t-1}$, and the peer's last obtained smoothed delta $\bar{\Delta}_i^{t-1}$ are all denoted as if peer i had participated in the previous local update and aggregation. This is not necessarily the case, since our system allows for partial participation and network churn. The last global model $\bar{\theta}_i^{t-1}$ could, for example, date back to the penultimate aggregation step (i.e., to FL iteration $(t-2)$). To clarify that the last global model and last obtained smoothed delta might differ among peers, both are denoted using the peer indicator i. After initializing the noise-calibrating parameters σ_b and σ_Δ using the number of participating aggregation peers n_t and noise multiplier σ_{mult}, peer i prepares its DP-safe local model $\hat{\theta}_i^{t,0}$. This is done by computing the local model update vector Δ_i, clipping, blurring, and smoothing it with factor β to obtain $\bar{\Delta}_i^{t,0}$, and then finally deriving $\hat{\theta}_i^{t,0}$, where η_u denotes the stepsize (we set $\beta = 0.9$ and $\eta_u = 0.1$). The noisy clipped local delta is denoted by $\widetilde{\Delta}_i$. A binary indicator $b_i^{t,0}$ reveals whether peer i has clipped its Δ_i to the clipping bound C_t. The squared noise calibration σ_Δ^2 is rescaled by n_t to account for noising local model deltas instead of their aggregated sum as Andrew et al. (2021) do.

Algorithm 4: DP-safe model aggregation in MAR-FL (for i-th peer in FL iteration t)

Input: $G, \mathcal{A}_t, n_t, \theta_i^t, m_i^t, \bar{\theta}_i^{t-1}, \bar{\Delta}_i^{t-1}, \beta, \sigma_{\text{mult}}, C_t, \gamma, \eta_u, \eta_C$
Output: $\theta^t, m^t, C_{t+1}, \bar{\Delta}^t$

$\sigma_b \leftarrow n_t/20$
$z_\Delta \leftarrow \left(\sigma_{\text{mult}}^{-2} - (2\sigma_b)^{-2}\right)^{-1/2}$
$\sigma_\Delta \leftarrow z_\Delta C^t$
$\Delta_i \leftarrow \theta_i^t - \bar{\theta}_i^{t-1}$
$b_i^{t,0} \leftarrow \mathbf{1}\{\, \|\Delta_i\| \le C_t \,\}$
$\widetilde{\Delta}_i \leftarrow \Delta_i \cdot \min\left(1,\, \frac{C_t}{\|\Delta_i\|}\right) + \mathcal{N}\left(0,\, I\frac{\sigma_\Delta^2}{n_t}\right)$
$\bar{\Delta}_i^{t,0} \leftarrow \begin{cases} \beta\, \bar{\Delta}_i^{t-1} + \widetilde{\Delta}_i, & \bar{\Delta}_i^{t-1} \neq \perp \\ \widetilde{\Delta}_i, & \text{otherwise} \end{cases}$
$\hat{\theta}_i^{t,0} \leftarrow \bar{\theta}_i^{t-1} + \eta_u\, \bar{\Delta}_i^{t,0}$
$m_i^{t,0} \leftarrow m_i^t$
for $g = 1, 2, \ldots, G$ **do**
  $\mathcal{P}_t := \{\, (j, \hat{\theta}_j^{t,g-1},\, m_j^{t,g-1},\, b_j^{t,g-1},\, \bar{\Delta}_j^{t,g-1}) \mid j \in \mathcal{A}_t \,\}$
  if $g < G$ **then**
    $(\hat{\theta}_i^{t,g},\, m_i^{t,g},\, b_i^{t,g},\, \bar{\Delta}_i^{t,g}) \leftarrow \texttt{MAR}_g(\mathcal{P}_t)$
  else
    $(\theta^t,\, m^t,\, \bar{b}^t,\, \bar{\Delta}^t) \leftarrow \texttt{MAR}_g(\mathcal{P}_t)$
    $\tilde{b}^{\,t} \leftarrow \bar{b}^t + \frac{\mathcal{N}(0,\, \sigma_b^2)}{n_t}$
    $C_{t+1} \leftarrow C_t \cdot \exp\big(-\eta_C\, (\tilde{b}^{\,t} - \gamma)\big)$
return $\theta^t,\, m^t,\, C_{t+1},\, \bar{\Delta}^t$

Over G rounds of MAR, each group-based MAR aggregation step $\texttt{MAR}_g$ iteratively averages relevant peer information $\mathcal{P}_t$ from the set of participating aggregation peers $\mathcal{A}_t$ until each peer $i \in \mathcal{A}_t$ obtains: (i) a global state $(\theta^t,\, m^t)$, (ii) a global clipping indicator $\bar{b}^t$, and (iii) a global smoothed delta $\bar{\Delta}^t$. The information peer i has so far aggregated up to MAR round $g \in \{1, 2, ..., G\}$ of the current FL iteration t is denoted as $(\hat{\theta}_i^{t,g},\, m_i^{t,g},\, b_i^{t,g},\, \bar{\Delta}_i^{t,g})$. A simple aggregation of binary indicators is not DP-safe as it reveals whether a peer i has clipped its model update vector Δ_i. To prevent this sensitive information leakage, a privacy-preserving mechanism (e.g., Secure Aggregation) has to be deployed for global binary indicator computation. When blurring the averaged binary indicator, sampled noise $\mathcal{N}(0, \sigma_b^2)$ is rescaled by the number of participating peers n_t, because we add noise to an average value and not to a sum as Andrew et al. (2021) do. The global averaging of smoothed deltas from all participating peers $i \in \mathcal{A}_t$ ensures that during global model aggregation of the next FL iteration $(t+1)$, the privatized local delta $\widetilde{\Delta}_i$ is mixed with a privatized global momentum delta $\bar{\Delta}_i^t$ before being applied to the last global model $\bar{\theta}_i^t$. This yields variance reduction and global alignment when computing a DP-safe local model. Eventually, the clipping bound is updated to C_{t+1}, tracking a target quantile γ of globally averaged clipping, where η_C denotes the stepsize (we set $\gamma = 0.5$ and $\eta_C = 0.2$). After each aggregation, the DP-safe global model θ^t is stored as the peer's last global model $\bar{\theta}_i^t$, to be used in its next global aggregation iteration; analogously for $\bar{\Delta}^t$. We note that the local momentum vectors m_i^t are not private as noise is applied only when each peer communicates their final model update for an aggregation round.

B Additional Experimental Details

B.1 Simulation Environment

We run all experiments on a high-performance computing (HPC) cluster using `Slurm` as the job scheduler. Each experiment runs on a single node with 4×H100 GPUs, 768 GB memory, and 96 CPU cores, reserving the entire node. After resource allocation, the job launches an `Enroot` runtime inside the allocation. The runtime is built from an `Enroot` SquashFS image created by im-

porting the NGC container nvcr.io/nvidia/pytorch:22.04-py3. Inside the container we use Python 3.8, PyTorch, and our MAR-FL and baseline implementations.

B.2 Implementation Details

Process model. We simulate peers as separate Python multiprocessing processes, each spawned by a dispatcher. Processes are created under a spawn context, assigned a unique peer ID, and pinned to specific CPU cores to simulate vCPUs. A shared multiprocessing.Manager() exposes two queues (task/results) and a shared dictionary for model exchange between the dispatcher and peers.

Dispatcher. A central dispatcher loop orchestrates FL iterations by: (i) selecting participating peers for local updates and aggregation (modeling partial participation and churn), (ii) enqueueing per-peer tasks (update, aggregate, skip, shutdown) on the task queue, (iii) collecting results, logging timings, and monitoring communication volume, (iv) performing early-stopping and robustness checks, and (v) periodically clearing stale entries from the shared dictionary.

Peer lifecycle. Each peer process follows three steps: (i) initialize a Hivemind DHT node to synchronize lightweight barriers and group-formation metadata (note that no model tensors are sent over the DHT), (ii) load its local data partition (MNIST or 20NG) and the ML model (CNN or DistilBERT head), and (iii) repeatedly execute tasks pulled from the task queue.

Group formation and synchronization (MAR-FL). At the beginning of the first MAR round, every peer initializes its group key. In each MAR round, peers then: (i) publish their presence via the DHT and collect peers with the same key, (ii) enforce group symmetry by cross-checking gathered group members through DHT keys, (iii) perform communication and aggregation within that group, and (iv) update the group key via a deterministic schedule before the next round. To prevent repeatedly matching with the same peers, group key updates leverage each peer's chunk index. This procedure aligns with the MAR algorithm of Ryabinin et al. (2021).

B.3 Experimental Setup

Datasets and models. To evaluate MAR-FL and all baselines on two distinct learning problems, we use one vision task (image classification) and one language task (text classification). For handwritten-digit recognition we employ a small two-block convolutional network with a compact multilayer-perceptron head that outputs class logits. MNIST images are loaded via torchvision and normalized in the usual way. For topic classification we use a lightweight classifier head on top of a frozen DistilBERT encoder (Sanh et al., 2019); the sequence representation is obtained from the classification token's (CLS token) hidden state, and the head produces 20-way logits. Text is tokenized with a BERT-base uncased tokenizer and sequences are padded to a fixed length. The 20 Newsgroups dataset is loaded from Hugging Face Datasets (SetFit/20_newsgroups).

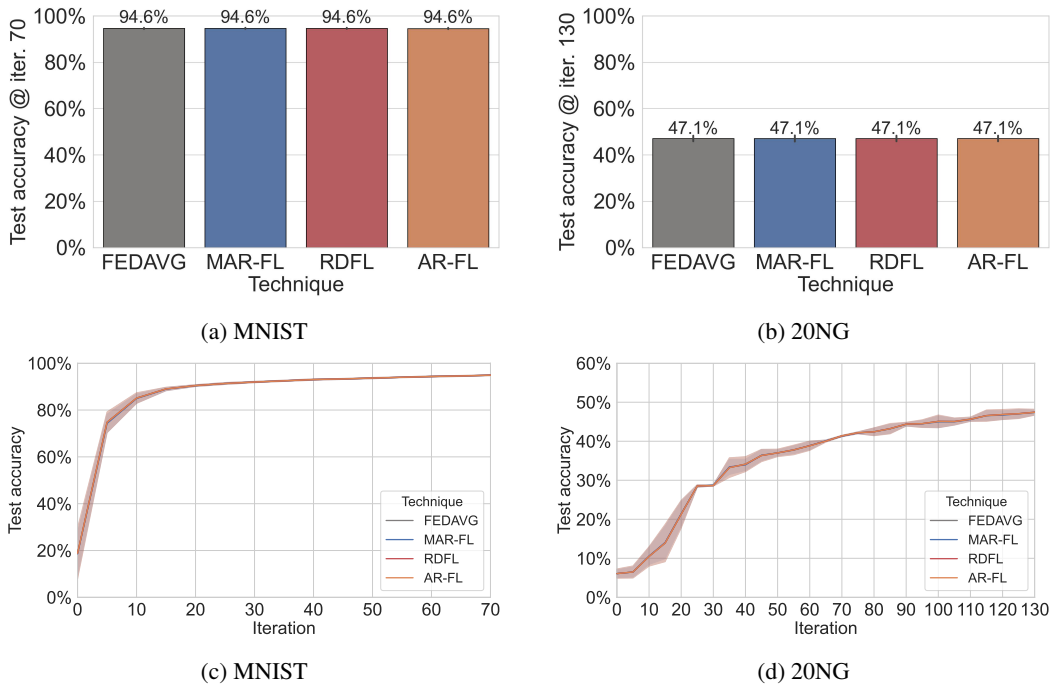
FL baselines. We do not utilize Butterfly All-Reduce (BAR) as an additional P2P FL baseline. BAR aims to reduce total communication load by assigning disjoint parameter chunks to different peers and only partially aggregating at each node. Under heterogeneous participation or network churn this yields incomplete/partially aggregated models, where the network might be stalled waiting for entire chunks of the model architecture. BAR consequently requires peers to be totally reliable. Hence we compare MAR-FL against FedAvg, RDFL, and AR-FL, which better reflect the characteristics of aggregation relevant to FL.

C Additional Results

C.1 Qualitative Results between MAR-FL and our Baselines

Qualitative identity. On MNIST and 20NG, MAR-FL achieves the same training performance as client-server FedAvg and the two P2P FL baselines (see Figure 5), as all four techniques yield identical global model averages under the given configurations.

Partial participation. On MNIST, MAR-FL incurs some loss in model utility under partial participation (see Figure 6), though the degradation is milder than on 20NG. However, even with only 50% peer participation and a 20% dropout likelihood, MAR-FL remains more than $5\times$ as



(a) MNIST (b) 20NG

(c) MNIST (d) 20NG

Figure 5: MAR-FL yields the same test accuracy as client-server FedAvg and our P2P FL baselines.

communication-efficient as our two P2P FL baselines. On 20NG, Figure 7 shows that FedAvg and both P2P baselines degrade to a similar extent, consistent with the behavior observed for MAR-FL (see Figure 3).

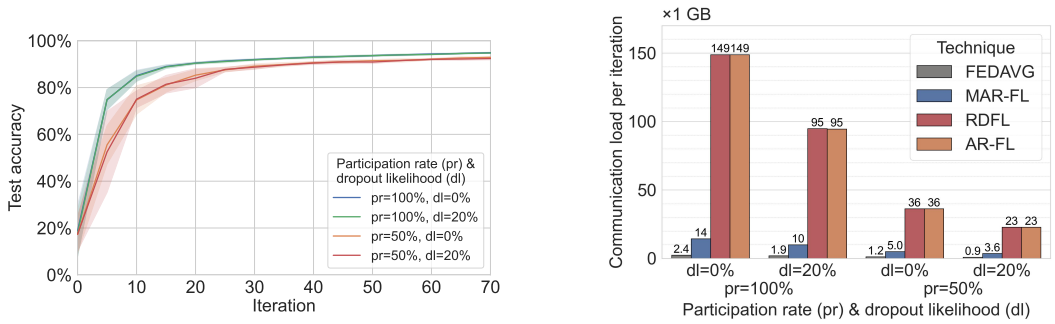
C.2 Qualitative Results of MAR-FL

Heterogeneous peer data. We employ Latent Dirichlet Allocation ($\alpha = 1.0$) to create non-i.i.d. local data splits among participating peers. While our simulation of real-world heterogeneous data distributions has no significant effect when training MAR-FL on MNIST, performance on 20NG is noticeably impaired compared to training with nearly i.i.d. local data splits (see Figure 8).

Improving communication efficiency with MKD. As on 20NG, MKD also accelerates convergence for MAR-FL on MNIST, enabling a target accuracy of 95% to be reached with up to $3\times$ lower total communication (see Figure 9), despite the increased per-iteration load from global aggregation. The number of KD iterations k is chosen such that – without data loader shuffling – for $k = 8$ on MNIST and $k = 6$ on 20NG each peer processes its entire local dataset twice, while for $k = 40$ on MNIST and $k = 30$ on 20NG it is seen ten times.

Differentially private training. As observed for 20NG, increasing privatization for MAR-FL on MNIST (i.e., raising the noise multiplier value σ) eventually degrades training performance (see Figure 10).

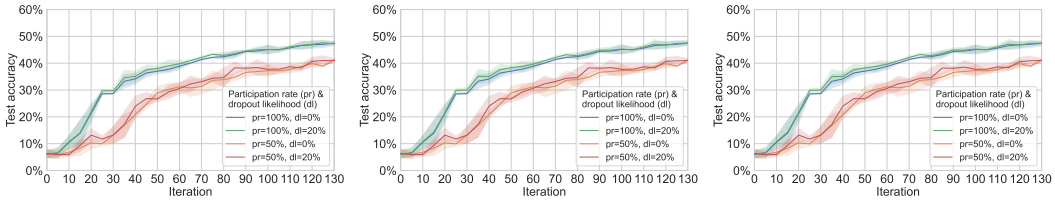
Leveraging approximate aggregation. As illustrated in Figure 11, MAR-FL's iterative group-based aggregation mechanism can be tuned to reduce communication while maintaining model utility. For example, with 125 peers, MAR-FL achieves an exact global model average when using group size 5 and 3 MAR rounds, since $5^3 = 125$ (with group key dimension $d = 3$). By relaxing these parameters (e.g., group size 3 and 4 MAR rounds), each iteration yields only approximate model averages. A well-designed group key update strategy is thus essential to closely approximate global averaging while minimizing the number of peer interactions per iteration. Over multiple iterations, these approximations converge to near-exact global averages, ensuring no substantial loss in model utility while significantly lowering communication cost. In our experiments, communication was reduced by up to 33% when using group size 3 and 4 MAR rounds with 125 peers (group key dimension $d = 4$).



(a) Model Performance for MAR-FL only

(b) Communication Cost

Figure 6: MAR-FL is affected by partial participation but resilient towards sudden dropouts. Plots show results on MNIST.

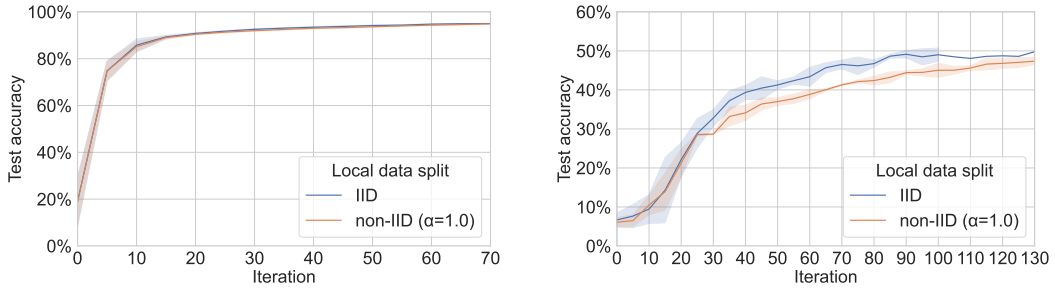


(a) FEDAVG

(b) RDFL

(c) AR-FL

Figure 7: Under partial participation and unreliable clients, FedAvg and our P2P FL baselines exhibit the same impact on training performance as shown for MAR-FL in Figure 3. Plots show results on 20NG.



(a) MNIST

(b) 20NG

Figure 8: Training performance of MAR-FL under i.i.d. and non-i.i.d. local data splits: performance on MNIST remains stable, whereas on 20NG non-i.i.d. splits lead to a noticeable degradation.

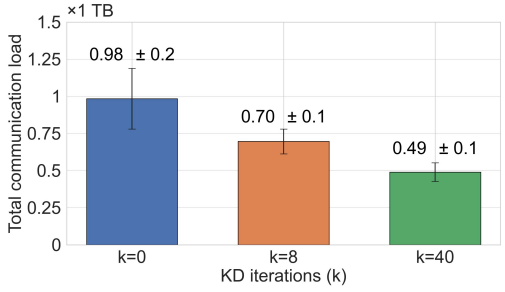


Figure 9: On MNIST, KD enables MAR-FL to reach a target accuracy of 95% with substantially lower communication cost (up to 3×).

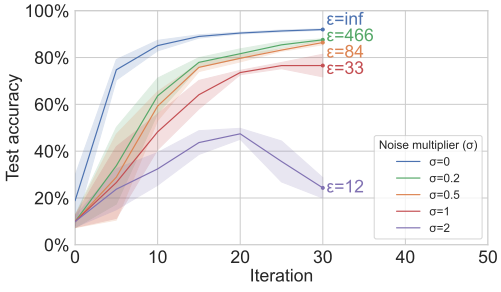
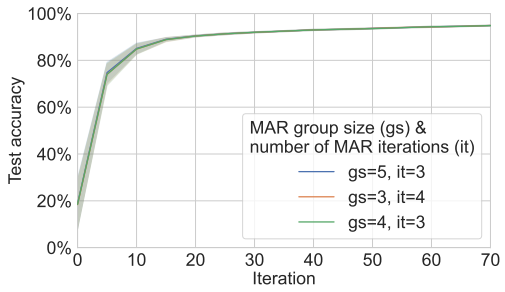
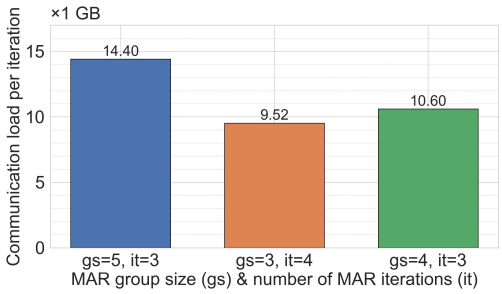


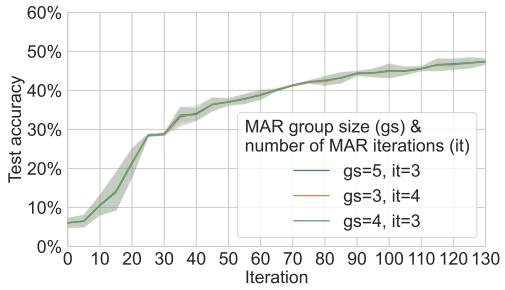
Figure 10: Integrating DP into MAR-FL results in the same performance degradation patterns as when applying DP to FedAvg. Plot shows results on MNIST.



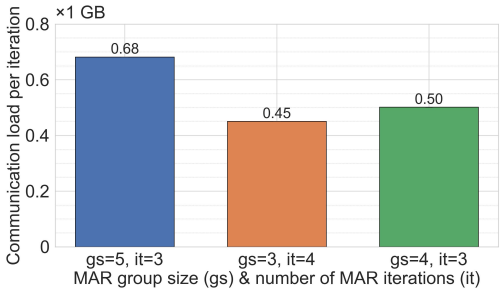
(a) Model Performance on MNIST



(b) Communication Cost on MNIST



(c) Model Performance on 20NG



(d) Communication Cost on 20NG

Figure 11: On both ML tasks, appropriately configured approximate aggregation enables MAR-FL to preserve model utility while further reducing communication costs by up to 33%.