Preprints.org

Article

Not peer-reviewed version

Optimization, Communication, and Personalization in Federated Learning for Massive Networks

Sameera Gallus *, Alex Mercer, Priya Singh, Daniel Cho

Posted Date: 11 July 2025

doi: 10.20944/preprints202507.1037.v1

Keywords: federated learning; large-scale distributed learning; communication efficiency; personalization; client heterogeneity; incentive mechanisms; privacy preservation; optimization algorithms; edge computing; distributed optimization

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Optimization, Communication, and Personalization in Federated Learning for Massive Networks

Sameera Gallus *, Alex Mercer, Priya Singh and Daniel Cho

Department of Computer Science, The University of Bath, United Kingdom
* Correspondence: sameera.gallus@bath.ac.uk

Abstract

We consider the problem of collaborative model optimization over a distributed network of agents, each possessing locally held data drawn from potentially heterogeneous distributions. The system operates under constraints of limited communication, partial participation, and privacy preservation, thereby necessitating the design of algorithms that balance local computation and global aggregation. We investigate the convergence properties and trade-offs arising in such iterative optimization schemes, where updates are performed asynchronously or synchronously, and communication overheads are mitigated via compression or quantization techniques. The objective is to characterize the interplay between model fidelity, communication complexity, and heterogeneity of local objective functions. We explore frameworks that enable personalized solutions tailored to individual agents while leveraging shared representations, often framed as multi-task or meta-optimization problems. Incentive structures are incorporated to model rational agent behavior under resource constraints and strategic participation, formalized through utility maximization and game-theoretic constructs. This work lays a foundation for understanding the fundamental limits and algorithmic principles governing scalable distributed learning systems, emphasizing theoretical guarantees alongside system-level considerations. Our approach highlights open questions concerning the balance of privacy, robustness, and efficiency in decentralized optimization, motivating future exploration into principled design and analysis of federated learning methodologies.

Keywords: federated learning; large-scale distributed learning; communication efficiency; personalization; client heterogeneity; incentive mechanisms; privacy preservation; optimization algorithms; edge computing; distributed optimization

1. Introduction

The rapid proliferation of edge devices such as smartphones, wearables, and IoT sensors has created unprecedented opportunities for distributed machine learning paradigms. Among these, *Federated Learning* (FL) has emerged as a promising approach for collaboratively training machine learning models across decentralized data sources while preserving user privacy and minimizing raw data exchange. Unlike traditional centralized learning, where all training data is aggregated on a central server, FL enables participants (or clients) to train local models on their private data and share only model updates with a central server for aggregation. As FL matures from academic exploration to real-world deployment, it faces a myriad of challenges that threaten its scalability, efficiency, and robustness. Chief among these is the issue of **communication bottlenecks**, particularly in large-scale federated learning environments involving millions of devices with heterogeneous computational capabilities, network bandwidths, and availability patterns. In such settings, communication between clients and the central server quickly becomes the primary performance bottleneck, overshadowing computation time and severely impacting convergence rates. The fundamental federated learning algorithm, Federated Averaging (FedAvg), introduced by McMahan et al., mitigates communication costs by performing multiple local updates before communicating with the server. However, even with

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

this improvement, the algorithm still suffers from high communication overhead due to the iterative nature of model training and the high dimensionality of modern deep learning models. When scaled to tens or hundreds of millions of clients, these costs become prohibitive [1]. Moreover, the communication issue is exacerbated by the inherent heterogeneity in client devices. Network conditions can vary significantly across clients, especially when training spans different geographic regions or network infrastructures. Clients may have asymmetric uplink and downlink speeds, intermittent connectivity, or restrictive data plans, all of which contribute to unreliable and expensive communication [2]. As a result, the efficiency of the FL process becomes closely tied to the slowest or least reliable clients, a phenomenon often referred to as the *straggler effect* [3]. Various strategies have been proposed to tackle communication challenges in FL, including model compression and sparsification, client selection and participation scheduling, asynchronous updates, and novel aggregation protocols. While each of these methods provides partial solutions, none singularly resolves the tension between communication efficiency, convergence accuracy, and system robustness. Furthermore, there is a fundamental trade-off between reducing communication and maintaining model performance [4]. Aggressive compression or quantization techniques may reduce communication payloads but can degrade model accuracy or cause instability in convergence. Similarly, selective client participation reduces total bandwidth use but may introduce biases if the selected clients are not representative of the overall data distribution. This survey aims to provide a comprehensive overview of the communication bottleneck in large-scale federated learning, examining its root causes, associated trade-offs, and state-of-the-art solutions. We categorize the landscape of proposed methods into several major strategies, analyze their theoretical and empirical effectiveness, and discuss their practical implications. Additionally, we highlight open research questions and outline potential directions for future work in communication-efficient federated learning systems. The rest of this paper is organized as follows. Section 2 provides background on the federated learning paradigm and formulates the communication problem. Section 3 reviews model compression and quantization techniques. Section 4 discusses adaptive communication protocols [5]. Section **??** explores strategies for client selection and scheduling. Section 6 presents asynchronous and decentralized communication frameworks. Section **??** outlines current challenges and future research directions [6]. Finally, Section 13 concludes the survey.

2. Background and Problem Formulation

Federated learning (FL) is a distributed optimization paradigm in which a central server coordinates training across a large number of clients, each holding a local dataset that is never directly shared with the server or other clients. The central objective in FL is to minimize a global empirical risk defined over the union of all local data distributions, while maintaining data locality. Formally, let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the set of participating clients, and let $\mathcal{D}_k$ represent the local dataset held by client $k \in \mathcal{K}$, such that the total number of samples across all clients is $n = \sum_{k=1}^{K} n_k$, where $n_k = |\mathcal{D}_k|$. We define the global objective function as the following empirical risk minimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}) \quad \text{where} \quad F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{\xi_i \in \mathcal{D}_k} \ell(\mathbf{w}; \xi_i), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the model parameter vector, $\ell(\mathbf{w}; \xi_i)$ is the loss function evaluated on sample ξ_i from client k, and $F_k(\mathbf{w})$ is the local empirical loss function specific to client k [7]. The central goal of federated optimization is to obtain a model $\mathbf{w}^*$ that minimizes $F(\mathbf{w})$, while enforcing that data remains decentralized and communication between clients and the server is minimized [8]. The canonical optimization strategy in federated learning is the Federated Averaging (FedAvg) algorithm, which proceeds in rounds. At each communication round t, the server broadcasts the current global model $\mathbf{w}^t$ to a subset of clients $\mathcal{S}^t \subseteq \mathcal{K}$ [9]. Each participating client $k \in \mathcal{S}^t$ performs E steps of local stochastic gradient descent (SGD) on its own dataset to compute an updated model $\mathbf{w}_k^{t+1}$:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

$$\mathbf{w}_k^{t+1} = \mathbf{w}^t - \eta \sum_{e=1}^{E} \nabla F_k(\mathbf{w}_k^{t,e}), \tag{2}$$

where $\eta > 0$ is the local learning rate and $\mathbf{w}_k^{t,e}$ is the model parameter at local epoch e on client k. The server then aggregates the updates via a weighted average:

$$\mathbf{w}^{t+1} = \sum_{k \in \mathcal{S}^t} \frac{n_k}{\sum_{j \in \mathcal{S}^t} n_j} \mathbf{w}_k^{t+1} [10]. \tag{3}$$

While FedAvg reduces communication frequency by allowing multiple local updates before synchronization, the communication burden still remains a significant bottleneck. Each participating client must transmit a model vector $\mathbf{w}_k^{t+1} \in \mathbb{R}^d$ back to the server after each round, leading to $O(d)$ communication cost per client per round. In deep neural networks, where d often ranges from millions to billions of parameters, such communication can become prohibitively expensive, particularly when extended to large-scale settings involving thousands or millions of clients. Moreover, the number of communication rounds T required for convergence may increase under non-iid data distributions and client heterogeneity, further compounding the communication overhead. Additionally, the stochastic nature of client availability introduces further complexity. Let $\mathcal{A}_t \subseteq \mathcal{K}$ denote the set of clients available at round t, and $\mathcal{S}^t \subseteq \mathcal{A}_t$ be the subset selected for participation [11]. The randomness in $\mathcal{A}_t$ introduces uncertainty in the optimization trajectory. If the availability is highly skewed or correlated with data characteristics, it may result in biased updates that affect convergence stability and generalization performance. Communication cost can be mathematically characterized in terms of total bits transmitted per round [12]. Let C denote the number of participating clients per round, d be the number of model parameters, and b be the number of bits used to encode each parameter (e.g., 32 bits for standard float32) [13]. Then, the communication cost per round is approximately $C \cdot d \cdot b$. Aggregated over T rounds, the total communication cost is $O(CdbT)$, which scales linearly with the number of rounds, model size, and clients per round. In practice, this scaling renders vanilla FL methods impractical for massive-scale deployments unless specialized communication-efficient strategies are employed. The above formulation also assumes a synchronous update protocol, where the server waits for all selected clients in $\mathcal{S}^t$ to complete their local updates and send back their model parameters. However, in realistic scenarios, clients may have heterogeneous computational capabilities and communication bandwidths, leading to stragglers—clients that significantly delay the round. To address this, asynchronous FL variants have been proposed, where clients communicate with the server independently, and the server performs updates based on stale or partial information [14]. Let τ_k^t denote the staleness of client k's update at round t, defined as the number of rounds that have elapsed since the client received the global model $\mathbf{w}^{t-\tau_k^t}$ [15]. The presence of such stale gradients complicates convergence analysis and often necessitates additional mechanisms such as temporal weighting or delay compensation to mitigate performance degradation. Furthermore, in environments with millions of clients, a fundamental challenge arises in selecting a subset of clients at each round that is both representative of the global data distribution and capable of contributing timely updates [16]. Random selection may yield suboptimal coverage, while uniform stratification is difficult without centralized access to client data. Let $\mathbf{p}^t = [p_1^t, p_2^t, \ldots, p_K^t]$ be the probability vector for client selection at round t [17]. Designing $\mathbf{p}^t$ to maximize utility while satisfying fairness, availability, and efficiency constraints is an open optimization problem with significant implications on both communication efficiency and model performance [18]. In summary, the communication bottleneck in large-scale federated learning is a complex, multi-faceted problem governed by the interaction between model dimensionality, number of clients, data heterogeneity, availability dynamics, and system-level constraints. Mathematical formulations such as those in equations (1)–(3) provide a foundational lens for understanding the optimization landscape, yet they abstract away critical systems-level details that must be accounted for in real-world implementations [19]. As the scale of federated deployments

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

continues to grow, a deeper theoretical and practical understanding of communication-efficient FL becomes not only desirable but essential [20].

3. Model Compression and Quantization Techniques

To alleviate the communication bottleneck in large-scale federated learning systems, one of the most widely adopted strategies is model compression, which aims to reduce the volume of data transmitted between clients and the central server. Compression techniques achieve this by reducing the size of the model updates (or gradients) communicated during each training round, typically through quantization, sparsification, or a combination of both. In this section, we provide a comprehensive exploration of these techniques, emphasizing their mathematical underpinnings, operational trade-offs, and implications for convergence behavior in federated optimization.

3.1. Gradient and Model Quantization

Quantization techniques reduce communication costs by representing high-precision floating-point values with lower-precision approximations [21]. Let $\mathbf{g} \in \mathbb{R}^d$ denote the gradient or model update vector produced by a client after local training. Instead of transmitting $\mathbf{g}$ directly, the client transmits a quantized version $\mathcal{Q}(\mathbf{g})$, where $\mathcal{Q} : \mathbb{R}^d \to \mathcal{C}^d$ is a quantization operator mapping vectors to a finite codebook $\mathcal{C} \subset \mathbb{R}$. The expected communication savings depend on the cardinality of $\mathcal{C}$ and the encoding scheme used. For instance, if $\mathcal{C}$ contains 2^b symbols, then each coordinate can be represented using b bits, yielding a communication cost of $d \cdot b$ bits per transmission [22]. A commonly used stochastic quantization method is the s-level unbiased quantizer $\mathcal{Q}_s$, introduced in [23]. For a scalar value $g_i \in \mathbb{R}$, the quantized output is:

$$\mathcal{Q}_s(g_i) = \|\mathbf{g}\|_2 \cdot \text{sign}(g_i) \cdot \xi_i, \quad \text{where } \xi_i \in \left\{ \frac{j}{s} : j \in [s] \right\}, \tag{4}$$

and ξ_i is sampled such that $\mathbb{E}[\mathcal{Q}_s(g_i)] = g_i$, thereby ensuring unbiasedness. The quantization level s controls the trade-off between communication cost and quantization error, with higher values of s offering better fidelity at the expense of increased bit complexity. The quantization error can be bounded as:

$$\mathbb{E}\left[\|\mathcal{Q}_s(\mathbf{g}) - \mathbf{g}\|_2^2 \right] \leq \min\left\{ \frac{d}{s^2}, \frac{1}{s} \right\} \|\mathbf{g}\|_2^2, \tag{5}$$

implying that lower quantization levels incur larger approximation errors that may slow convergence or destabilize training. Another popular quantization technique is ternary quantization, where gradients are restricted to the set $\{-1, 0, +1\}$ with scalar scaling, allowing efficient encoding using just 2 bits per coordinate.

3.2. Sparsification and Top-k Compression

Another prominent compression technique involves sparsification, wherein only the most informative components of the gradient vector are communicated. In Top-k sparsification, a client transmits only the k largest-magnitude components of its update $\mathbf{g}$, with the rest set to zero. Let $\mathcal{S}_k(\mathbf{g})$ be the sparsification operator that retains the k largest elements in magnitude:

$$\mathcal{S}_k(\mathbf{g})_i = \begin{cases} g_i & \text{if } i \in \text{Top-}k(\mathbf{g}), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

To ensure information is not permanently lost during sparsification, an error feedback mechanism is typically employed. Each client maintains a residual vector $\mathbf{r}_t$ representing the accumulated quantization or sparsification error. Before computing the update at round t, the client adds $\mathbf{r}_t$ back into the local update:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

$$\tilde{\mathbf{g}}_t = \mathbf{g}_t + \mathbf{r}_t, \quad \mathbf{r}_{t+1} = \tilde{\mathbf{g}}_t - \mathcal{S}_k(\tilde{\mathbf{g}}_t). \tag{7}$$

This error feedback loop ensures that untransmitted components are eventually communicated, enabling convergence under certain conditions [24]. The convergence rate of sparsified SGD with error feedback has been analyzed in, showing that sparsified methods can retain convergence guarantees comparable to vanilla SGD, provided the sparsification budget k is sufficiently large [25].

3.3. Low-Rank and Sketching Methods

In addition to quantization and sparsification, recent work has explored low-rank approximation techniques to compress model updates. These methods approximate the gradient matrix $\mathbf{G} \in \mathbb{R}^{d\times B}$, where B is the mini-batch size, with a low-rank matrix $\mathbf{U}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{V} \in \mathbb{R}^{B\times r}$ for rank $r \ll \min(d, B)$. The compressed update is transmitted in the form of the factors $\mathbf{U}$ and $\mathbf{V}$, reducing communication to $O(r(d + B))$ [26]. Sketching techniques such as CountSketch or Random Projection further reduce dimensionality by projecting high-dimensional updates into lower-dimensional subspaces using randomized linear maps. Let $\mathbf{S} \in \mathbb{R}^{m\times d}$ be a sketching matrix, where $m \ll d$, then the compressed update is:

$$\tilde{\mathbf{g}} = \mathbf{S}\mathbf{g}, \quad \text{and recovered (approximately) as } \hat{\mathbf{g}} = \mathbf{S}^\dagger\tilde{\mathbf{g}}, \tag{8}$$

where $\mathbf{S}^\dagger$ is the pseudo-inverse of $\mathbf{S}$. While these techniques provide strong compression ratios, they often require centralized coordination or shared randomness between clients and server, complicating decentralized implementations.

3.4. Implications for Convergence and System Design

The use of compression in federated learning introduces a fundamental trade-off between communication efficiency and optimization fidelity. Let $\mathbf{w}^t$ denote the global model at round t, and $\hat{\mathbf{g}}^t$ be the aggregated compressed update. The global model update becomes:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t\hat{\mathbf{g}}^t. \tag{9}$$

When $\hat{\mathbf{g}}^t$ is a biased or noisy approximation of the true gradient $\mathbf{g}^t$, the convergence rate and final accuracy can be adversely affected [27]. Several works have shown that with careful design—e.g., unbiased quantization, error feedback, or adaptive compression levels—it is possible to retain convergence guarantees with bounded error. Let $\mathbb{E}\left[\|\hat{\mathbf{g}}^t - \mathbf{g}^t\|^2\right] \le \delta_t$, then under smoothness assumptions on the objective $F(\mathbf{w})$, we can bound the suboptimality gap after T rounds as:

$$\mathbb{E}\Big[F(\mathbf{w}^T) - F(\mathbf{w}^*)\Big] = O\left(\frac{1}{T} + \frac{1}{T}\sum_{t=1}^{T}\delta_t\right). \tag{10}$$

This result illustrates that as long as the average compression error δ_t remains small, compressed FL methods can converge to a neighborhood of the global optimum [28]. However, in practice, the choice of compression scheme, its implementation efficiency, and its interaction with client heterogeneity must all be carefully calibrated to avoid performance degradation.

3.5. Summary

Model compression and quantization constitute a powerful class of techniques to mitigate the communication bottleneck in federated learning [29]. Through quantized updates, sparsified gradients, low-rank factorization, and random projections, these methods aim to reduce the size of transmitted messages without sacrificing model performance. However, compression inevitably introduces approximation errors, necessitating the use of error correction mechanisms such as residual accumulation and adaptive encoding. While the theoretical underpinnings of these methods are increasingly well-understood, practical deployment remains challenging due to system heterogeneity, variable

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

client bandwidths, and compatibility with secure aggregation protocols [30]. In the next section, we examine adaptive communication strategies that dynamically adjust communication frequency and client participation to further enhance the efficiency of large-scale federated learning systems.

4. Adaptive Communication Protocols

In large-scale federated learning, the communication overhead not only depends on the size of the transmitted model updates but also on the frequency and timing of communication rounds between the clients and the central server [31]. Adaptive communication protocols seek to optimize the trade-off between communication efficiency and convergence speed by dynamically adjusting when and how clients exchange information. This section delves into the mathematical frameworks, algorithmic strategies, and theoretical guarantees underlying adaptive communication in federated optimization.

4.1. Communication Frequency Adaptation

A fundamental aspect of adaptive communication is controlling the communication interval, i.e., how many local computation steps a client performs before communicating with the server. Standard FedAvg uses a fixed number of local epochs E per communication round, but recent research suggests that dynamically tuning E based on system or optimization metrics can significantly improve efficiency. Consider the local update at client k after E_k^t local epochs during round t:

$$\mathbf{w}_k^{t+1} = \mathbf{w}^t - \eta \sum_{e=1}^{E_k^t} \nabla F_k(\mathbf{w}_k^{t,e}), \tag{11}$$

where E_k^t is now client- and round-dependent. Increasing E_k^t reduces communication frequency but may lead to local model drift, especially in heterogeneous data environments, which can slow down convergence or degrade final accuracy [32]. Conversely, smaller E_k^t values increase communication overhead but ensure more frequent global synchronization. Adaptive strategies model this trade-off explicitly. For example, *adaptive local update control* approaches estimate an optimal E_k^t by minimizing an objective that balances local computation cost C_{comp} and communication cost C_{comm}, while constraining convergence error. Formally, one may solve:

$$\min_{E_k^t \in \mathbb{Z}_+} \quad \alpha C_{\text{comp}}(E_k^t) + \beta C_{\text{comm}}(E_k^t) \quad \text{s.t.} \quad \epsilon(E_k^t) \leq \epsilon_{\max}, \tag{12}$$

where $\alpha, \beta > 0$ are weighting factors and $\epsilon(E_k^t)$ is the expected convergence error as a function of local update length. Analytical approximations of $\epsilon(E_k^t)$ depend on problem smoothness and heterogeneity and can be used to derive principled adaptation policies.

4.2. Event-Triggered Communication

Rather than communicating at fixed intervals, *event-triggered communication* protocols transmit updates only when local model changes exceed a specified threshold. This approach leverages the observation that not all local updates contribute equally to global optimization progress. Formally, let $\mathbf{w}_k^{t+1}$ be the local model after local computation, and $\mathbf{w}_k^{t,\text{last}}$ be the model last communicated by client k. The client transmits the update only if:

$$\|\mathbf{w}_k^{t+1} - \mathbf{w}_k^{t,\text{last}}\|_p \geq \tau_k^t, \tag{13}$$

where $\|\cdot\|_p$ denotes the p-norm (typically $p = 2$ or $p = \infty$) and $\tau_k^t > 0$ is a dynamically adjusted threshold [33]. Choosing τ_k^t involves a trade-off: larger thresholds reduce communication but may delay convergence, while smaller thresholds increase communication load [34]. Some protocols adapt thresholds τ_k^t based on network conditions, client importance, or model dynamics [35]. Event-triggered

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

schemes can be combined with compression techniques to amplify communication savings while preserving accuracy.

4.3. Adaptive Client Participation

Another dimension of adaptivity addresses which clients participate in each communication round [36]. Large-scale federated systems cannot afford to involve all clients simultaneously due to bandwidth constraints and client availability variability. Adaptive client selection policies aim to identify subsets $\mathcal{S}^t \subseteq \mathcal{K}$ that maximize training utility while minimizing communication. Let $\mathbf{p}^t = [p_1^t, \ldots, p_K^t]$ be the vector of client selection probabilities at round t. Adaptive methods adjust $\mathbf{p}^t$ based on criteria such as:

- **Client reliability:** Favoring clients with stable connectivity and timely responses to reduce straggler effects [37].
- **Data representativeness:** Selecting clients whose data distributions complement existing updates to reduce bias and improve generalization.
- **Update magnitude:** Prioritizing clients with larger local model changes to accelerate learning progress.

Formally, selection probabilities can be optimized by solving:

$$\max_{\mathbf{p}^t \in \Delta^K} \quad U(\mathbf{p}^t) - \lambda R(\mathbf{p}^t), \tag{14}$$

where Δ^K is the probability simplex, $U(\mathbf{p}^t)$ measures expected utility of the selected clients, $R(\mathbf{p}^t)$ quantifies communication or fairness costs, and λ balances the trade-off. Heuristic algorithms such as importance sampling or multi-armed bandits have been proposed to solve this problem efficiently.

4.4. Theoretical Guarantees

Adaptive communication protocols often introduce non-uniform communication intervals and variable client participation, complicating convergence analysis. Nonetheless, under assumptions such as bounded gradient variance and smoothness of local objectives, recent theoretical work has established convergence guarantees for adaptive FL methods [38]. For example, when clients perform E_k^t local SGD steps before communication, the expected suboptimality after T rounds satisfies:

$$\mathbb{E}[F(\mathbf{w}^T) - F(\mathbf{w}^*)] \leq O\left(\frac{1}{\sum_{t=1}^T \sum_{k\in\mathcal{S}^t} n_k E_k^t} + \frac{\sum_{t=1}^T \sum_{k\in\mathcal{S}^t} \sigma_k^2}{\left(\sum_{t=1}^T \sum_{k\in\mathcal{S}^t} n_k E_k^t\right)^2}\right), \tag{15}$$

where σ_k^2 denotes the variance of local gradients on client k. This bound highlights the importance of balancing communication frequency and local computation to optimize convergence speed [39].

4.5. Practical Considerations and Challenges

Adaptive communication protocols require mechanisms to monitor client states, estimate update significance, and coordinate participation, all of which introduce overhead and system complexity. Further, fairness concerns arise since clients with poor connectivity or slower updates may be excluded more frequently, potentially biasing the model and reducing inclusivity. Designing adaptive schemes that are robust, privacy-preserving, and scalable remains an open challenge. Combining adaptive communication with compression, asynchronous updates, and secure aggregation protocols presents promising avenues for research and development.

4.6. Summary

Adaptive communication protocols dynamically tune communication frequency, event triggers, and client participation to mitigate bandwidth constraints in large-scale federated learning. These approaches leverage system and optimization feedback to optimize the communication-computation

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

trade-off, leading to improved efficiency without sacrificing convergence guarantees [40]. However, practical deployment necessitates addressing challenges related to client heterogeneity, fairness, and implementation overhead. The next section explores client selection strategies in greater detail, a critical component of adaptive communication frameworks.

5. Client Selection and Scheduling

In federated learning systems encompassing potentially millions of devices, selecting which subset of clients participates in each communication round is a crucial factor affecting both system efficiency and model performance [41]. Client selection and scheduling techniques aim to optimize the use of limited communication resources, reduce latency caused by slow or unavailable clients (stragglers), and improve convergence rates by judiciously leveraging data diversity and client capabilities [42]. This section provides an in-depth examination of the mathematical formulations, algorithmic strategies, and theoretical insights underpinning client selection in large-scale federated learning [43].

5.1. Problem Formulation

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the entire set of clients, where K can be very large [44]. At each communication round t, a subset $\mathcal{S}^t \subseteq \mathcal{K}$ of size $S \ll K$ is selected to participate. The selection process can be modeled as sampling from a probability distribution $\mathbf{p}^t = (p_1^t, p_2^t, \ldots, p_K^t)$ over clients, with constraints:

$$\sum_{k=1}^{K} p_k^t = 1, \quad p_k^t \geq 0, \quad \forall k, t [45].$$

Each client k possesses local data distribution $\mathcal{D}_k$ and may have associated costs or capabilities such as communication bandwidth b_k, computational speed c_k, and availability indicator $a_k^t \in \{0, 1\}$ [46]. The goal of client selection is to maximize a utility function $U(\mathcal{S}^t)$ that quantifies the contribution of selected clients towards global model improvement, subject to resource constraints and fairness criteria. A common optimization formulation is:

$$\max_{\mathcal{S}^t \subseteq \mathcal{K}, |\mathcal{S}^t| = S} \quad \mathbb{E}_{\mathbf{p}^t}[U(\mathcal{S}^t)] \quad \text{s.t.} \quad \sum_{k \in \mathcal{S}^t} C_k \leq C_{\max}, \quad \text{and fairness constraints,} \tag{16}$$

where C_k denotes cost metrics (e.g., communication or computation costs) and $C_{\max}$ is a system budget.

5.2. Importance Sampling for Variance Reduction

One theoretically grounded client selection strategy is *importance sampling*, which adjusts the selection probabilities p_k^t based on the expected informativeness or gradient norm magnitude from each client. The intuition is to prioritize clients whose updates have higher variance or contribute more significantly to reducing the global objective [47]. Formally, let $\mathbf{g}_k^t$ be the stochastic gradient computed by client k at round t. The variance of the aggregated gradient estimator

$$\hat{\mathbf{g}}^t = \frac{1}{S} \sum_{k \in \mathcal{S}^t} \frac{\mathbf{g}_k^t}{p_k^t}$$

can be minimized by setting

$$p_k^t \propto \|\mathbf{g}_k^t\|_2,$$

which allocates higher sampling probability to clients with larger gradient magnitudes, thereby reducing estimator variance. However, computing exact gradient norms for all clients prior to selection is impractical, so approximations or historical statistics are often used [48].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

5.3. Straggler Mitigation via Deadline-Aware Scheduling

Straggler clients — those with slow computation or poor network conditions — can substantially delay synchronous federated learning rounds. Scheduling techniques that mitigate straggler effects improve training throughput and stability [49]. One approach is *deadline-aware scheduling*, where the server imposes a time constraint $T_{\max}$ per round [50]. Clients whose expected runtime r_k^t exceeds $T_{\max}$ are excluded from participation:

$$\mathcal{S}^t = \{k \in \mathcal{K} : r_k^t \leq T_{\max},\ a_k^t = 1\},$$

and the server aggregates updates from this subset [51]. This heuristic prioritizes faster clients, ensuring timely model aggregation at the cost of potentially discarding updates from slower clients with useful data [52]. Balancing deadline selection $T_{\max}$ is critical to avoid biasing the model towards fast clients only.

5.4. Fairness and Diversity Considerations

Unbalanced client selection risks marginalizing clients with limited connectivity or less frequent availability, exacerbating bias and degrading model generalization. Incorporating fairness into scheduling ensures equitable representation across clients and data distributions. Mathematically, fairness can be enforced by constraining selection probabilities to satisfy:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{P}(k \in \mathcal{S}^t) \geq \phi_k,$$

where ϕ_k is the minimum participation frequency for client k. Alternatively, fairness-aware objective functions incorporate penalty terms or regularizers that promote selection diversity:

$$\max_{\mathbf{p}^t} U(\mathbf{p}^t) - \lambda \sum_{k=1}^{K}\left(p_k^t - \frac{1}{K}\right)^2,$$

where $\lambda > 0$ controls the trade-off between utility and uniform client selection. Diverse client sampling can also mitigate the impact of non-IID data distributions, enhancing convergence and robustness.

5.5. Scheduling Algorithms

Several algorithmic frameworks have been proposed for client scheduling:

- **Greedy Selection:** Select clients maximizing incremental utility until resource constraints are met. This approach is simple but can be myopic.
- **Multi-Armed Bandits (MAB):** Model client selection as an MAB problem where the server learns client utility over time, balancing exploration and exploitation [53].
- **Optimization-Based Selection:** Solve a constrained optimization problem using techniques such as projected gradient descent or integer programming to find selection probabilities [54].

5.6. Convergence Analysis with Partial Client Participation

Partial client participation complicates convergence guarantees, especially under heterogeneous data distributions. Recent theoretical results establish that under bounded gradient variance and smoothness assumptions, federated averaging with random partial participation converges at a rate of:

$$\mathbb{E}[F(\mathbf{w}^T) - F(\mathbf{w}^*)] = O\left(\frac{1}{\sqrt{ST}} + \frac{\Delta}{T}\right),$$

where S is the number of clients sampled per round, T is the total number of rounds, and Δ quantifies data heterogeneity. These results highlight that increasing S improves convergence but raises communication cost, underscoring the need for optimal client selection.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

5.7. Summary

Client selection and scheduling play pivotal roles in managing communication constraints, addressing system heterogeneity, and enhancing model quality in federated learning. By leveraging importance sampling, deadline-aware heuristics, and fairness-aware algorithms, federated systems can balance efficiency, robustness, and equity. Future research directions include scalable scheduling for ultra-large client pools, integration with secure aggregation, and adaptive strategies responsive to dynamic client behavior.

6. Asynchronous Federated Learning

Traditional federated learning algorithms, such as FedAvg, assume synchronous communication rounds where the server waits for all selected clients to complete their local computations and send updates before performing aggregation. However, in large-scale deployments, synchronous operation is often impractical due to client heterogeneity in computation power, communication bandwidth, and availability [55]. Asynchronous federated learning (AFL) relaxes this synchronization constraint, allowing the server to incorporate client updates as they arrive, thereby improving system efficiency and reducing idle time. This section thoroughly examines the mathematical foundations, algorithmic designs, and convergence properties of asynchronous federated learning.

6.1. Modeling Asynchrony

In AFL, each client k independently computes local updates based on potentially stale global models, and the server continuously integrates these updates without waiting for all clients. Denote by $\mathbf{w}^t$ the global model at the server at iteration t, and by $\mathbf{w}_k^{t-\tau_k}$ the version of the global model that client k received at some past iteration $t-\tau_k$, where $\tau_k \geq 0$ is the staleness or delay associated with client k's update. Client k performs local computation starting from $\mathbf{w}_k^{t-\tau_k}$, producing an update $\Delta\mathbf{w}_k^t$, which is sent asynchronously to the server [56]. The server then updates the global model as follows:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \sum_{k \in \mathcal{A}^t} \Delta\mathbf{w}_k^t, \tag{17}$$

where $\mathcal{A}^t$ is the set of client updates received at iteration t, and η_t is the learning rate schedule. The asynchronous nature means that $\Delta\mathbf{w}_k^t$ may be computed using stale parameters, introducing challenges in convergence and stability.

6.2. Impact of Staleness

The delay τ_k induces a discrepancy between the model parameters used to compute local updates and the current global model. This discrepancy can be formally analyzed by decomposing the update as:

$$\Delta\mathbf{w}_k^t = -\eta \nabla F_k(\mathbf{w}^{t-\tau_k}) + \text{,}_k^t, \tag{18}$$

where $\nabla F_k(\mathbf{w}^{t-\tau_k})$ is the gradient of the local loss function F_k evaluated at the stale global model and ,_k^t is a stochastic noise term due to local sampling. Because $\mathbf{w}^{t-\tau_k}$ may differ significantly from $\mathbf{w}^t$, the aggregated update may no longer be a descent direction for the current global objective, potentially slowing convergence or causing divergence [57]. Under standard assumptions of L-smoothness and μ-strong convexity of the global objective $F(\mathbf{w}) = \sum_{k=1}^K p_k F_k(\mathbf{w})$, the convergence rate of AFL algorithms can be characterized by bounding the error introduced by staleness. For example, the expected optimization error satisfies:

$$\mathbb{E}[F(\mathbf{w}^t) - F(\mathbf{w}^*)] \leq \mathcal{O}\left(\frac{1}{t}\right) + \mathcal{O}\left(\eta L^2 \max_k \tau_k^2\right), \tag{19}$$

indicating that the staleness term $\max_k \tau_k$ critically affects convergence. Careful design of learning rates η_t and staleness-aware aggregation strategies is necessary to mitigate these effects.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

6.3. Algorithmic Strategies for AFL

Several algorithms have been proposed to accommodate asynchronous updates while ensuring convergence and system efficiency:

- **Staleness-weighted aggregation:** Assign weights to client updates inversely proportional to their staleness, e.g.,

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \sum_{k \in \mathcal{A}^t} \frac{1}{1+\tau_k} \Delta \mathbf{w}_k^t,$$

  which downweights outdated updates to reduce their negative impact.
- **Adaptive learning rates:** Employ time-decaying or delay-aware learning rates $\eta_t(\tau_k)$ that reduce step sizes for stale updates, balancing convergence speed and stability [58].
- **Bounded staleness protocols:** Enforce a maximum allowable delay $\tau_{\max}$ by discarding excessively stale updates, trading off system responsiveness with convergence guarantees [59].
- **Asynchronous variance reduction:** Integrate variance reduction techniques such as SVRG or SAGA adapted to asynchronous settings to reduce gradient noise amplified by staleness [60].

6.4. Theoretical Convergence Guarantees

The convergence analysis of AFL typically relies on bounded delay assumptions and Lipschitz continuity of gradients. Under these assumptions, it can be shown that asynchronous federated algorithms converge to stationary points with rates similar to their synchronous counterparts, up to additional error terms that depend on staleness and system heterogeneity. Formally, assuming bounded gradient variance σ^2, bounded delay $\tau_{\max}$, and step size $\eta_t = \frac{\eta_0}{\sqrt{t}}$, the expected convergence rate satisfies:

$$\min_{t \in [T]} \mathbb{E}\left[\|\nabla F(\mathbf{w}^t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{\tau_{\max}^2}{T}\right), \tag{20}$$

demonstrating that as the number of iterations T increases, the negative impact of staleness diminishes [61].

6.5. Practical Challenges and System Considerations

Deploying AFL in real-world systems requires addressing issues such as:

- **Client heterogeneity:** Diverse computational speeds and network conditions yield variable staleness distributions, complicating system design [62].
- **Model consistency:** Ensuring that asynchronous updates do not cause model parameter inconsistencies or conflicts, especially when using non-convex models.
- **Privacy preservation:** Integrating asynchronous updates with privacy-preserving mechanisms like secure aggregation or differential privacy is non-trivial due to the dynamic and out-of-sync nature of updates [63].
- **Fault tolerance:** Handling client dropouts and unreliable connections without stalling the global training process.

Strategies combining asynchronous communication with robust aggregation, adaptive staleness control, and secure protocols are active research areas.

6.6. Summary

Asynchronous federated learning offers a promising paradigm to overcome the scalability and latency challenges inherent in synchronous protocols, particularly in large-scale, heterogeneous environments. By incorporating updates as they arrive and mitigating staleness-induced errors, AFL enhances resource utilization and reduces training time [64]. Nevertheless, careful algorithmic and system design is essential to preserve convergence guarantees, maintain fairness, and uphold privacy.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

The following section will explore advanced model compression techniques that can be integrated with asynchronous and synchronous FL to further alleviate communication bottlenecks [65].

7. Model Compression and Quantization Techniques

Communication overhead is one of the fundamental bottlenecks in large-scale federated learning systems, especially as models grow in size and the number of participating clients increases dramatically. Since federated learning requires frequent transmission of model updates or gradients between clients and the central server, reducing the communication payload without compromising model accuracy is critical [66]. Model compression and quantization techniques have thus emerged as indispensable tools to alleviate bandwidth constraints, lower energy consumption on edge devices, and accelerate convergence. This section delves into the mathematical principles, algorithmic frameworks, and trade-offs involved in model compression and quantization within federated learning [67].

7.1. Problem Setup and Objectives

Consider a global model parameter vector $\mathbf{w} \in \mathbb{R}^d$ of dimensionality d, typically in the order of millions or more for modern deep learning architectures. Each client k computes a local update $\Delta\mathbf{w}_k$, which must be communicated to the server. Direct transmission of $\Delta\mathbf{w}_k$ incurs a communication cost proportional to d, which becomes prohibitive for resource-constrained clients or limited bandwidth networks. The goal of compression is to find a function $Q : \mathbb{R}^d \to \mathcal{C}$ mapping the update vector to a compressed representation in a code space $\mathcal{C}$ with reduced bit-length [68]. Formally, the compression function Q aims to minimize communication bits while controlling the introduced approximation error:

$$\min_{Q} \quad \mathbb{E}\Big[\|Q(\Delta\mathbf{w}_k) - \Delta\mathbf{w}_k\|^2\Big], \quad \text{s.t.} \quad \text{bit-length}(Q(\Delta\mathbf{w}_k)) \leq B,$$

where $B \ll d \times$ bit-width of the original update, and the expectation is over randomness in the compression operator (if randomized).

7.2. Quantization Methods

Quantization discretizes continuous-valued parameters into a finite set of levels, thereby enabling compact encoding [69].

Uniform Quantization

The simplest quantization scheme partitions the value range into q equally spaced levels. For each coordinate $[\Delta\mathbf{w}_k]_i$, the quantized value is:

$$Q_i(\Delta\mathbf{w}_k) = \text{clip}(\Delta\mathbf{w}_k, a, b) \approx \frac{b-a}{q-1} \cdot \left\lfloor \frac{(\Delta\mathbf{w}_k)_i - a}{(b-a)/(q-1)} + \xi_i \right\rfloor + a,$$

where a and b are bounds (e.g., min and max), and $\xi_i \sim \text{Uniform}(0,1)$ is a random dithering variable to ensure unbiasedness [70]. Uniform quantization achieves a compression ratio of approximately $\frac{32}{\log_2 q}$ per coordinate assuming 32-bit floats originally.

Stochastic Quantization

To maintain unbiasedness, stochastic quantization selects adjacent quantization levels probabilistically such that

$$\mathbb{E}[Q_i(\Delta\mathbf{w}_k)] = (\Delta\mathbf{w}_k)_i,$$

which reduces bias-induced convergence degradation. Formally,

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

$$Q_i(\Delta \mathbf{w}_k) = \begin{cases} q_j, & \text{with probability } p = \frac{q_{j+1} - (\Delta \mathbf{w}_k)_i}{q_{j+1} - q_j}, \\ q_{j+1}, & \text{with probability } 1 - p, \end{cases}$$

where q_j and q_{j+1} are the two closest quantization levels surrounding $(\Delta \mathbf{w}_k)_i$ [71].

7.3. Sparsification Techniques

Sparsification reduces communication by sending only a small subset of update coordinates with the largest magnitudes, zeroing out others. Formally, define a sparsification operator S_k such that:

$$[S_k(\Delta \mathbf{w}_k)]_i = \begin{cases} (\Delta \mathbf{w}_k)_i, & i \in \mathcal{I}_k, \\ 0, & \text{otherwise}, \end{cases}$$

where $\mathcal{I}_k \subseteq \{1, \ldots, d\}$ is a subset with $|\mathcal{I}_k| = s \ll d$. The selection can be deterministic (top-s largest coordinates) or randomized with sampling probabilities proportional to magnitude. Sparsification introduces bias if the zeroed coordinates are consistently ignored; hence, error-feedback mechanisms store residuals locally to be added in subsequent rounds:

$$\mathbf{r}_k^{t+1} = \mathbf{r}_k^t + \Delta \mathbf{w}_k^t - S_k(\Delta \mathbf{w}_k^t),$$

and compress $S_k(\mathbf{r}_k^{t+1})$ in the next iteration [72]. This technique provably preserves convergence guarantees despite aggressive sparsification.

7.4. Low-Rank and Structured Compression

Another approach exploits redundancy in parameter updates by approximating $\Delta \mathbf{w}_k$ with low-rank or structured representations [73]. For example, matrix factorization techniques approximate updates as:

$$\Delta \mathbf{W}_k \approx \mathbf{U}_k \mathbf{V}_k^\top,$$

where $\Delta \mathbf{W}_k \in \mathbb{R}^{m \times n}$ reshapes the update vector, and $\mathbf{U}_k \in \mathbb{R}^{m \times r}$, $\mathbf{V}_k \in \mathbb{R}^{n \times r}$ with rank $r \ll \min(m, n)$. Communicating $\mathbf{U}_k$ and $\mathbf{V}_k$ reduces bits substantially, though computational overhead for decomposition must be accounted for. Other structural compression includes quantizing convolutional filters or pruning network weights dynamically during training to reduce communication [74,75].

7.5. Theoretical Trade-Offs

Compression inherently introduces noise or bias, which affects convergence [76]. Let the compressed update be $Q(\Delta \mathbf{w}_k) = \Delta \mathbf{w}_k + \mathbf{e}_k$, where $\mathbf{e}_k$ is the compression error with zero mean and bounded variance:

$$\mathbb{E}[\mathbf{e}_k] = 0, \quad \mathbb{E}[\|\mathbf{e}_k\|^2] \leq \delta \|\Delta \mathbf{w}_k\|^2,$$

for some $\delta \in (0, 1)$. Under such assumptions, the convergence rate of federated averaging with compressed updates satisfies:

$$\mathbb{E}[F(\mathbf{w}^T) - F(\mathbf{w}^*)] = \mathcal{O}\left(\frac{1}{\sqrt{T}} + \delta\right),$$

indicating a trade-off between compression ratio and final accuracy [77].

7.6. Integration with Federated Learning Protocols

Compression techniques can be applied both on client-to-server uplink updates and server-to-client downlink broadcasts. In practice, uplink compression is prioritized due to constrained client bandwidth [78]. Adaptive compression strategies, which dynamically adjust quantization levels or

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

sparsity ratios based on network conditions or model training stage, further enhance performance. Moreover, compression must be carefully designed to maintain privacy guarantees. For instance, combining quantization with secure aggregation protocols requires compatible encoding schemes to allow aggregation of compressed values without revealing individual updates [79].

7.7. Summary

Model compression and quantization techniques are vital for scalable federated learning by significantly reducing communication overhead. The choice among uniform or stochastic quantization, sparsification with error feedback, and low-rank approximation depends on system constraints and model characteristics [80]. Understanding the interplay between compression-induced noise and convergence behavior guides the design of efficient federated algorithms. Future directions include learning adaptive compression policies and combining compression with privacy and robustness mechanisms [81].

8. Privacy and Security Challenges in Large-Scale Federated Learning

Federated learning's fundamental promise is to enable collaborative model training without directly sharing raw user data, thereby enhancing privacy [82]. However, the distributed nature of FL introduces unique privacy and security vulnerabilities that are exacerbated in large-scale deployments with thousands or millions of heterogeneous clients. This section explores the mathematical foundations and practical implications of privacy-preserving techniques, threat models, and defenses in federated learning, emphasizing the complex interplay between privacy guarantees, communication efficiency, and learning utility.

8.1. Threat Models and Privacy Risks

Despite never transmitting raw data, FL is vulnerable to a range of attacks that exploit model updates. Let D_k denote the private dataset of client k, and let $\mathbf{w}^t$ be the global model at round t [83]. Clients compute local updates $\Delta\mathbf{w}_k^t$ based on D_k and send them to the server. An adversary can be either the server or other clients aiming to infer sensitive information about D_k from $\Delta\mathbf{w}_k^t$.

Inference Attacks

Model updates leak information through gradients or weight changes. Gradient inversion attacks reconstruct input data $x \in D_k$ by optimizing an auxiliary input $\hat{x}$ to minimize

$$\min_{\hat{x}} \left\| \nabla_{\mathbf{w}} \ell(\mathbf{w}^t; \hat{x}) - \Delta\mathbf{w}_k^t \right\|^2,$$

where ℓ is the loss function. Such attacks exploit the functional dependency of gradients on specific training samples.

Membership Inference Attacks

Adversaries infer whether a particular data point x^* was part of D_k by analyzing statistical differences in updates [84]. Formally, given access to updates $\Delta\mathbf{w}_k^t$, the attacker learns a classifier

$$\mathcal{A} : \Delta\mathbf{w}_k^t \mapsto \{0, 1\},$$

indicating presence or absence of x^* in the local dataset, compromising membership privacy [85].

Poisoning and Backdoor Attacks

Malicious clients can send carefully crafted updates $\Delta\mathbf{w}_m^t$ to corrupt the global model or insert backdoors [86]. The objective is to maximize an attack loss $L_{\text{attack}}(\mathbf{w}^{t+1})$ by manipulating the aggregation step:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \left(\sum_{k \neq m} \Delta \mathbf{w}_k^t + \Delta \mathbf{w}_m^t \right) [87].$$

Such adversarial behavior threatens model integrity and reliability [88].

8.2. Differential Privacy in Federated Learning

Differential privacy (DP) offers rigorous quantification of privacy leakage by ensuring the output of a mechanism is statistically indistinguishable when any single data point is added or removed. Formally, a randomized mechanism $\mathcal{M}$ satisfies (ε, δ)-DP if for all neighboring datasets D and D' differing in one record,

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta, \quad \forall S \subseteq \text{Range}(\mathcal{M}) [89].$$

Applying DP to FL typically involves perturbing local updates before transmission. Each client adds noise $\mathbf{z}_k^t$ sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathbf{I})$:

$$\widetilde{\Delta \mathbf{w}}_k^t = \Delta \mathbf{w}_k^t + \mathbf{z}_k^t,$$

where the noise scale σ is calibrated to achieve a desired privacy budget (ε, δ). The global model update becomes

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \sum_{k=1}^{K} \widetilde{\Delta \mathbf{w}}_k^t.$$

While DP provides provable privacy guarantees, it also injects bias and variance into the optimization process, degrading convergence speed and model accuracy. Balancing privacy parameters with utility is thus a core challenge [90].

8.3. Secure Aggregation Protocols

To prevent the server from inspecting individual updates, secure aggregation protocols enable the server to only observe the aggregate sum of encrypted client updates. Clients employ cryptographic primitives such as additive homomorphic encryption or secret sharing to encode updates:

$$\text{Enc}(\Delta \mathbf{w}_k^t) \xrightarrow[\text{aggregation}]{} \text{Enc}\left(\sum_{k=1}^{K} \Delta \mathbf{w}_k^t \right),$$

which the server decrypts to update the global model without learning any individual contribution. Such protocols incur communication and computational overhead, especially in large-scale FL with many clients [91]. Designing scalable secure aggregation methods that tolerate client dropout and asynchronous participation remains an open problem [92].

8.4. Robust Aggregation Methods

Robust aggregation algorithms aim to defend against adversarial updates by mitigating the influence of outliers or malicious clients. Techniques include:

- **Median and Trimmed Mean Aggregation:** Replacing the average with coordinate-wise median or trimmed mean reduces sensitivity to extreme values.
- **Krum and Multi-Krum:** Selecting client updates closest to the majority by minimizing the sum of distances to other updates to exclude outliers.
- **Norm Bounding and Clipping:** Limiting the L_2 norm of updates to control the impact of large malicious updates [93].

These defenses improve resilience but often rely on assumptions about the fraction of adversarial clients and can degrade performance in highly heterogeneous or non-IID settings [94].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

8.5. Trade-Offs and Open Challenges

Ensuring privacy and security in large-scale FL involves fundamental trade-offs:

- **Privacy vs. Utility:** Adding noise for DP or restricting updates reduces model accuracy.
- **Security vs. Scalability:** Cryptographic protocols add overhead and complexity, challenging deployment at scale.
- **Robustness vs [95]. Fairness:** Filtering or down-weighting suspicious updates may unintentionally marginalize honest but statistically distinct clients.

Future research directions include designing privacy-preserving algorithms that adaptively balance these trade-offs, developing scalable cryptographic schemes tailored for heterogeneous client populations, and integrating anomaly detection with robust aggregation for dynamic adversary mitigation.

8.6. Summary

Privacy and security are paramount in federated learning but remain challenging to guarantee simultaneously at scale. Differential privacy and secure aggregation provide formal protections against information leakage, while robust aggregation methods defend against poisoning attacks. However, the large scale, heterogeneity, and asynchronous participation in federated learning exacerbate these challenges, necessitating novel approaches that holistically address privacy, security, efficiency, and learning performance.

9. System Heterogeneity and Scalability Challenges

Large-scale federated learning systems operate across a vast and diverse array of client devices, each characterized by distinct computational capabilities, network conditions, power constraints, and data distributions [96]. This heterogeneity introduces significant challenges to both the efficiency and the effectiveness of federated training algorithms. Scalability concerns arise not only from the sheer number of clients but also from the complex interactions between system-level variability and algorithmic convergence [97]. This section rigorously analyzes the modeling of system heterogeneity, its impact on federated learning, and algorithmic strategies to mitigate scalability bottlenecks [98].

9.1. Modeling Device and Network Heterogeneity

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the set of participating clients, where K can range from thousands to millions [99]. Each client $k \in \mathcal{K}$ is associated with a set of system parameters:

$$\Theta_k = \{\tau_k, c_k, b_k, \nu_k\},$$

where:

- τ_k is the local computation speed (e.g., FLOPS),
- c_k is the communication bandwidth available to client k,
- b_k is the battery or power budget,
- ν_k quantifies data heterogeneity or non-IID degree in client k's local dataset D_k [100].

These parameters evolve over time and directly influence the client's participation frequency, update latency, and reliability [101]. The variation in τ_k leads to stragglers—clients that require significantly more time to perform local computation—causing synchronization delays in synchronous federated protocols.

9.2. Impact on Federated Optimization

Consider the standard Federated Averaging (FedAvg) update at round t:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t \sum_{k \in \mathcal{S}^t} \frac{n_k}{n_{\mathcal{S}^t}} \Delta \mathbf{w}_k^t,$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

where $\mathcal{S}^t \subseteq \mathcal{K}$ is the subset of clients selected at round t, $n_k = |D_k|$, and $n_{\mathcal{S}^t} = \sum_{k\in\mathcal{S}^t} n_k$ [102]. Due to heterogeneous resources, the latency L_k^t for client k can be approximated by:

$$L_k^t = \underbrace{\frac{C_{\text{comp}}}{\tau_k}}_{\text{Computation time}} + \underbrace{\frac{S_{\text{comm}}}{c_k}}_{\text{Communication time}} + \delta_k^t,$$

where C_{comp} is the number of local computation operations, S_{comm} the size of communicated data, and δ_k^t captures stochastic delays (e.g., network jitter). In synchronous aggregation, the server waits for all clients in $\mathcal{S}^t$ to finish, resulting in overall latency:

$$L_{\text{sync}}^t = \max_{k\in\mathcal{S}^t} L_k^t,$$

which can be dominated by stragglers. Conversely, asynchronous federated learning avoids waiting but complicates the convergence analysis due to stale updates.

9.3. Client Sampling and Scheduling Strategies

Scalability to massive K necessitates partial client participation via stochastic sampling [103]. Let p_k^t denote the probability that client k is selected at round t [104]. Optimizing p_k^t to balance training speed, fairness, and resource constraints is nontrivial. Adaptive sampling strategies weight clients by resource availability or data quality metrics. For example, sampling proportional to

$$p_k^t \propto \frac{\tau_k c_k}{\nu_k + \epsilon},$$

prioritizes clients with faster computation and communication while penalizing highly heterogeneous data, thus accelerating convergence [105].

9.4. Mitigating Stragglers and Asynchrony

Several algorithmic solutions address straggler effects:

Partial Participation with Deadlines

Set a maximum wait time $T_{\max}$ for client responses. Updates arriving after $T_{\max}$ are discarded, reducing iteration latency at the expense of fewer updates per round.

Asynchronous Federated Optimization

Clients communicate updates independently and the server maintains a global model updated incrementally [106]. Formally, for asynchronous update j received at time t_j from client k_j:

$$\mathbf{w}^{t_j} = \mathbf{w}^{t_j-1} + \eta_{t_j} \Delta \mathbf{w}_{k_j}^{t_j-\tau_{k_j}},$$

where τ_{k_j} accounts for staleness [107]. Convergence proofs require bounding staleness and controlling update variance [108].

9.5. Energy Efficiency and Power Constraints

Battery limitations b_k impose additional constraints, often requiring clients to adjust participation frequency or local computation intensity dynamically [109]. Modeling energy consumption E_k^t as

$$E_k^t = \alpha C_{\text{comp}} / \tau_k + \beta S_{\text{comm}} / c_k,$$

with constants α, β representing power costs per computation and communication unit, clients may adopt sleep schedules or early stopping to conserve energy, affecting global training dynamics.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

9.6. Scalability in Model and Data Heterogeneity

Heterogeneous data distributions across clients induce statistical challenges impacting convergence rates. Define the degree of non-IID-ness ν_k as

$$\nu_k = \|\nabla F_k(\mathbf{w}) - \nabla F(\mathbf{w})\|,$$

where F_k is client k's local objective, and F is the global objective [110]. High variance in ν_k leads to slower convergence and potential divergence in federated averaging. Techniques such as control variates and personalized federated learning frameworks mitigate data heterogeneity effects by correcting client drift or learning client-specific models [111].

9.7. Summary

System heterogeneity, encompassing computational, communication, power, and data diversity, presents formidable obstacles to scalable federated learning. Modeling client-specific parameters allows for informed scheduling and adaptive algorithms that mitigate stragglers and balance fairness with efficiency [112]. Asynchronous methods, deadline-based partial participation, and energy-aware protocols form key approaches to address latency and resource constraints [113]. The confluence of system and statistical heterogeneity necessitates holistic optimization to realize the promise of federated learning at scale.

10. Communication Efficiency and Compression Techniques

Communication bottlenecks constitute a critical challenge in large-scale federated learning (FL) due to limited bandwidth, high latency, and the enormous scale of participating devices [114]. Since each communication round involves transmitting model updates—often high-dimensional parameter vectors—efficient communication protocols and compression schemes are essential to reduce overhead without significantly degrading model convergence or accuracy [115]. This section presents a rigorous analysis of communication costs, quantization, sparsification, and adaptive compression strategies, including their theoretical underpinnings and practical trade-offs.

10.1. Communication Cost Formulation

Consider the global model parameter vector at round t as $\mathbf{w}^t \in \mathbb{R}^d$, where d is the model dimension, typically ranging from millions to billions in deep learning models. Each client k computes a local update $\Delta\mathbf{w}_k^t \in \mathbb{R}^d$, which must be communicated to the server [116]. The raw communication cost per round is

$$C_{\text{raw}} = K \times d \times b,$$

where K is the number of participating clients and b is the number of bits per parameter (e.g., 32 bits for full-precision floating-point) [117]. At large scales, C_{raw} becomes prohibitively expensive.

10.2. Quantization Techniques

Quantization reduces b by representing model updates with fewer bits. Let $\mathcal{Q} : \mathbb{R}^d \to \mathcal{C}^d$ be a stochastic quantizer mapping continuous-valued updates to a discrete codebook $\mathcal{C}$. A popular approach is stochastic s-level quantization:

$$\mathcal{Q}(\Delta w_i) = \|\Delta\mathbf{w}\|_2 \cdot \text{sign}(\Delta w_i) \cdot \xi_i,$$

where

$$\xi_i = \begin{cases} \frac{l}{s}, & \text{with probability } 1-p, \\ \frac{l+1}{s}, & \text{with probability } p, \end{cases} \quad l = \lfloor s|\Delta w_i|/\|\Delta\mathbf{w}\|_2 \rfloor, \quad p = s|\Delta w_i|/\|\Delta\mathbf{w}\|_2 - l.$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

This unbiased quantizer satisfies

$$\mathbb{E}[\mathcal{Q}(\Delta \mathbf{w})] = \Delta \mathbf{w},$$

and has bounded variance proportional to $\frac{d}{s^2}$, trading off compression ratio against gradient noise.

10.3. Sparsification Methods

Gradient sparsification reduces communication by transmitting only a subset of significant update coordinates. Define the sparsification operator $\mathcal{S}_k^t : \mathbb{R}^d \to \mathbb{R}^d$ that retains the top-m coordinates (in magnitude) of $\Delta \mathbf{w}_k^t$ and zeros out the rest:

$$\mathcal{S}_k^t(\Delta \mathbf{w}_k^t) = \Delta \mathbf{w}_k^t \odot \mathbf{m}_k^t,$$

where $\mathbf{m}_k^t \in \{0,1\}^d$ is a mask vector with exactly m ones corresponding to the selected coordinates [118]. To ensure unbiasedness and reduce error accumulation, error-feedback mechanisms maintain a local residual vector $\mathbf{r}_k^t$ updated as:

$$\mathbf{r}_k^{t+1} = \mathbf{r}_k^t + \Delta \mathbf{w}_k^t - \mathcal{S}_k^t(\Delta \mathbf{w}_k^t),$$

which is incorporated into the next round's update. Error compensation improves convergence guarantees [119].

10.4. Adaptive Compression and Communication Frequency

Adaptive compression techniques dynamically adjust quantization levels or sparsification ratios based on model training progress or network conditions. For instance, early training stages tolerate higher compression noise, allowing aggressive reduction, whereas later stages reduce compression to refine convergence. Communication frequency control strategies skip communication rounds or perform multiple local updates before aggregation. Let E be the number of local epochs between communications; then communication rounds occur at intervals of length E, reducing communication cost by approximately a factor of E:

$$C_{\text{comm}} \approx \frac{K \times d \times b}{E} text[120].$$

This FedAvg-style local updating trades off communication for increased local computation, but large E can cause client drift and degrade convergence in non-IID data settings.

10.5. Theoretical Trade-offs and Convergence Bounds

Combining compression and local updates introduces noise and bias [121]. Let the compression operator be $\mathcal{C}$ with contraction parameter δ satisfying

$$\mathbb{E}\|\mathcal{C}(x) - x\|^2 \leq (1-\delta)\|x\|^2,$$

for all $x \in \mathbb{R}^d$. Then, under smoothness and convexity assumptions, the expected optimization error after T rounds obeys bounds of the form:

$$\mathbb{E}[F(\mathbf{w}^T) - F(\mathbf{w}^*)] \leq \mathcal{O}\left(\frac{1}{\eta T} + \frac{\sigma^2}{K} + \frac{1-\delta}{\delta}\right),$$

where σ^2 is gradient variance and η is learning rate. Higher compression (smaller δ) increases the optimization error floor, necessitating a careful balance.

10.6. Practical Considerations

Communication compression schemes must also consider:

- **Encoding and Decoding Overhead:** Computational cost on resource-constrained devices.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

- **Robustness to Packet Loss and Asynchrony:** Ensuring stability in unreliable networks [122].
- **Compatibility with Privacy Mechanisms:** Noise added by differential privacy can interact nontrivially with compression noise [123].

10.7. Summary

Communication efficiency is a critical enabler for scaling federated learning. Quantization and sparsification substantially reduce transmitted data sizes, while adaptive compression and communication frequency adjustment provide flexible trade-offs between communication cost and convergence speed [124]. However, these techniques introduce additional noise and bias, requiring rigorous theoretical analysis and practical tuning to ensure robust, efficient, and accurate federated training at scale.

11. Personalization in Large-Scale Federated Learning

Traditional federated learning aims to learn a single global model $\mathbf{w}^*$ that performs well on the aggregated data distribution across all clients [125]. However, in large-scale scenarios, client data distributions are typically heterogeneous and non-identically distributed (non-IID), leading to suboptimal performance of a single global model on individual clients. Personalization techniques seek to tailor models to each client's unique data characteristics while leveraging shared knowledge from the federation. This section investigates mathematical formulations of personalization, algorithmic approaches, and challenges in balancing global generalization and local specialization.

11.1. Problem Formulation

Let $F_k(\mathbf{w})$ denote the local objective function of client k defined as

$$F_k(\mathbf{w}) = \mathbb{E}_{\mathbf{x}\sim D_k}[\ell(\mathbf{w};\mathbf{x})],$$

where $\ell(\mathbf{w};\mathbf{x})$ is the loss of model parameters $\mathbf{w}$ on data point $\mathbf{x}$ [126]. The classical federated learning objective is the weighted average:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}),$$

with $n_k = |D_k|$ and $n = \sum_{k=1}^{K} n_k$. In personalization, the goal is to learn client-specific models $\mathbf{w}_k$ by solving

$$\min_{\{\mathbf{w}_k\}_{k=1}^{K}} \sum_{k=1}^{K} \frac{n_k}{n} (F_k(\mathbf{w}_k) + \lambda \mathcal{R}(\mathbf{w}_k, \mathbf{w}_G)),$$

where $\mathbf{w}_G$ is a global reference model, $\lambda > 0$ is a regularization parameter, and $\mathcal{R}$ is a proximity function encouraging similarity between personalized and global models, typically

$$\mathcal{R}(\mathbf{w}_k, \mathbf{w}_G) = \frac{1}{2}\|\mathbf{w}_k - \mathbf{w}_G\|^2 \text{ [127]}.$$

This formulation balances local fitting and global knowledge transfer [128].

11.2. Meta-Learning Approaches

Meta-learning optimizes the global model such that it can be efficiently adapted to any client with a few gradient steps. A prototypical meta-learning formulation is:

$$\min_{\mathbf{w}} \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w} - \alpha \nabla F_k(\mathbf{w})),$$

where α is the inner learning rate [129]. This bilevel optimization enables fast adaptation and has been instantiated in algorithms like Model-Agnostic Meta-Learning (MAML) [130].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

11.3. Multi-Task and Clustered Federated Learning

When clients form latent clusters with similar data distributions, multi-task FL frameworks learn a set of models $\{\mathbf{w}_c\}_{c=1}^{C}$ indexed by clusters:

$$\min_{\{\mathbf{w}_c\}} \sum_{k=1}^{K} \sum_{c=1}^{C} \pi_{k,c} F_k(\mathbf{w}_c),$$

where $\pi_{k,c} \in [0,1]$ indicates the membership probability of client k in cluster c [131]. Cluster assignments may be learned jointly with model parameters to exploit statistical similarity.

11.4. Personalized Model Architectures

Personalization can also be achieved via model architectures decomposing parameters into shared and private components:

$$\mathbf{w}_k = \mathbf{w}_G + \mathbf{v}_k,$$

where $\mathbf{w}_G$ is the global shared model and $\mathbf{v}_k$ is the client-specific perturbation. Training optimizes jointly over $\mathbf{w}_G$ and $\{\mathbf{v}_k\}$, often with regularization on $\mathbf{v}_k$ to control complexity.

11.5. Challenges and Open Questions

Key challenges in personalization include:

- **Communication Overhead:** Personalized models require additional communication and storage, especially with large $\mathbf{v}_k$ vectors.
- **Privacy Risks:** Personalization may leak client-specific information; privacy-preserving personalization remains an active research area.
- **Fairness and Robustness:** Balancing performance across diverse clients to avoid disadvantaging minority groups.
- **Theoretical Guarantees:** Establishing convergence and generalization bounds for personalized federated algorithms under heterogeneous data.

11.6. Summary

Personalization enriches federated learning by addressing client heterogeneity through local adaptation, meta-learning, clustering, and architectural decomposition. These methods improve client-level performance and user satisfaction but introduce new trade-offs in communication, privacy, and algorithmic complexity. Designing scalable, privacy-aware, and theoretically sound personalization algorithms remains an essential direction for advancing federated learning at scale.

12. Incentive Mechanisms and Client Participation

In large-scale federated learning (FL) systems, the participation of heterogeneous clients—often voluntary and resource-constrained—is a critical factor affecting both training efficiency and model quality [132]. Designing effective incentive mechanisms is essential to motivate clients to contribute their data and computational resources while ensuring fairness and sustainability of the learning ecosystem. This section provides a comprehensive survey of incentive models, participation strategies, and their theoretical foundations within federated learning frameworks [133].

12.1. Rationale and Challenges

Unlike centralized training, FL relies on distributed devices owned by independent agents (e.g., users, organizations), each with private costs and utilities. Participation incurs costs such as energy consumption, bandwidth usage, and privacy risks. Therefore, without proper incentives, clients may behave selfishly, reduce participation frequency, or supply low-quality updates, leading to degraded system performance. Key challenges include:

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

- **Asymmetric Information:** The server often lacks knowledge of individual client costs and data quality.
- **Strategic Behavior:** Clients may misreport resources or withhold updates to maximize personal utility.
- **Fairness:** Incentives must balance rewarding contribution and preventing exploitation of disadvantaged clients.
- **Scalability:** Mechanisms must be computationally and communication efficient for massive client populations.

12.2. Mathematical Modeling of Client Utilities

Let client k's utility at round t be defined as

$$U_k^t = R_k^t - C_k^t,$$

where

- R_k^t is the reward (monetary, reputation, or other incentives) offered by the server,
- C_k^t represents the client's cost, encompassing computational, communication, and privacy expenses.

Clients participate if $U_k^t \geq 0$ [134]. The server aims to design reward schemes R_k^t that maximize global objectives (e.g., model accuracy, fairness) subject to budget constraints

$$\sum_{k=1}^{K} R_k^t \leq B_t,$$

where B_t is the total available incentive budget at round t.

12.3. Game-Theoretic Incentive Mechanisms

Federated learning with self-interested clients can be modeled as a game $\mathcal{G} = (\mathcal{K}, \{A_k\}, \{U_k\})$, where:

- $\mathcal{K}$ is the set of clients,
- A_k is the action space for client k, typically participation level or quality of update,
- U_k is the utility function as defined above.

The server seeks a mechanism that induces a Nash equilibrium maximizing collective welfare [135]. Mechanisms include:

Auction-Based Approaches

Clients bid their costs and the server selects winners maximizing model improvement per unit cost [136]. Vickrey–Clarke–Groves (VCG) auctions and reverse auctions guarantee truthfulness and efficiency.

Contract Theory

The server offers contracts (R_k, θ_k) specifying rewards R_k contingent on observed quality θ_k. Clients self-select contracts matching their type, ensuring incentive compatibility and individual rationality [137].

12.4. Reputation and Trust Systems

Reputation scores r_k^t quantify client reliability and contribution history. The server uses r_k^t to modulate rewards or participation priority, promoting long-term engagement and discouraging malicious behavior. Formally, let the updated reward be

$$R_k^t = f(r_k^t, q_k^t),$$

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

where q_k^t measures update quality. Reputation evolves as

$$r_k^{t+1} = \gamma r_k^t + (1-\gamma) q_k^t,$$

with forgetting factor $\gamma \in (0,1)$.

12.5. Participation and Scheduling Policies

To manage limited communication budgets and system heterogeneity, the server implements client selection policies $\pi_t : \mathcal{K} \to \{0,1\}$ indicating client participation at round t. Optimizing π_t balances:

$$\max_{\pi_t} \mathbb{E}\big[\Delta F(\mathbf{w}^t)\big] - \eta \sum_{k=1}^{K} \pi_t(k) C_k^t,$$

where $\Delta F(\mathbf{w}^t)$ is the expected model improvement and η trades off accuracy and cost [138]. Scheduling may prioritize clients with:

- High-quality data,
- Low latency and resource costs,
- Good reputation scores,
- Diverse data to reduce bias [139].

12.6. Privacy-Aware Incentives

Clients face privacy risks quantified by a leakage function $\mathcal{L}_k^t$ [140]. Privacy-preserving mechanisms (e.g., differential privacy) degrade update utility, increasing C_k^t [141]. Incentive schemes must compensate privacy costs or adjust noise parameters to balance privacy and participation.

12.7. Summary

Incentive mechanisms are pivotal in sustaining large-scale federated learning ecosystems by aligning client motivations with global learning objectives. Game-theoretic, contract-based, and reputation-driven designs provide theoretical frameworks to encourage truthful and sustained participation while addressing privacy, fairness, and scalability [142]. Effective participation policies and budget-aware rewards ensure efficient resource utilization and robust model convergence across diverse client populations.

13. Conclusion

In this survey, we have provided an extensive overview of the critical challenges and recent advances in large-scale federated learning (FL). As a paradigm that enables collaborative model training across massive numbers of decentralized clients without sharing raw data, FL presents unique opportunities and formidable obstacles in scalability, communication efficiency, system heterogeneity, personalization, incentive design, and privacy preservation.

We began by outlining the foundational principles of federated learning and highlighting the scalability challenges posed by the sheer volume of participating devices and the high dimensionality of modern models. We then explored optimization algorithms tailored to FL settings, emphasizing the trade-offs between local computation and global communication. The necessity of communication-efficient strategies was examined through various compression and quantization techniques that reduce bandwidth consumption while maintaining convergence guarantees.

Addressing client heterogeneity, we surveyed personalization methods that adapt global models to individual clients' data distributions via meta-learning, clustering, and architectural decomposition. These approaches enhance model accuracy and user experience but introduce additional complexities in privacy, communication, and theoretical analysis.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

Recognizing that client participation in FL is inherently voluntary and resource-sensitive, we discussed incentive mechanisms grounded in game theory, contract theory, and reputation systems. These mechanisms play a pivotal role in encouraging truthful and sustained client engagement, balancing fairness, privacy, and system efficiency.

Throughout the survey, we underscored the interplay between theoretical guarantees and practical system considerations, advocating for holistic approaches that integrate algorithmic innovation with system design and economic incentives. Despite substantial progress, numerous open problems remain, including robust handling of extreme data heterogeneity, scalable and privacy-preserving personalization, dynamic incentive models under evolving network conditions, and securing FL systems against adversarial threats.

In conclusion, advancing large-scale federated learning demands multidisciplinary efforts spanning machine learning, optimization, communication theory, economics, and security. By addressing the intertwined challenges outlined herein, future research will unlock FL's full potential to enable privacy-aware, efficient, and personalized intelligence across ubiquitous edge devices, fundamentally transforming the landscape of collaborative artificial intelligence.

References

1. Sattler, F.; Marban, A.; Rischke, R.; Samek, W. Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Transactions on Network Science and Engineering* **2021**.
2. Shi, W.; Zhou, S.; Niu, Z.; Jiang, M.; Geng, L. Joint device scheduling and resource allocation for latency constrained wireless federated learning. *IEEE Transactions on Wireless Communications* **2020**, *20*, 453–467.
3. Sergeev, A.; Del Balso, M. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799* **2018**.
4. Gray, R.M.; Neuhoff, D.L. Quantization. *IEEE transactions on information theory* **1998**, *44*, 2325–2383.
5. Rieke, N.; Hancox, J.; Li, W.; Milletari, F.; Roth, H.R.; Albarqouni, S.; Bakas, S.; Galtier, M.N.; Landman, B.A.; Maier-Hein, K.; et al. The future of digital health with federated learning. *NPJ digital medicine* **2020**, *3*, 1–7.
6. Chen, H.; Huang, S.; Zhang, D.; Xiao, M.; Skoglund, M.; Poor, H.V. Federated learning over wireless IoT networks with optimized communication and resources. *IEEE Internet of Things Journal* **2022**.
7. Donoho, D.L. Compressed sensing. *IEEE Transactions on information theory* **2006**, *52*, 1289–1306.
8. Dai, X.; Yan, X.; Zhou, K.; Yang, H.; Ng, K.K.; Cheng, J.; Fan, Y. Hyper-sphere quantization: Communication-efficient sgd for federated learning. *arXiv preprint arXiv:1911.04655* **2019**.
9. Chen, S.; Shen, C.; Zhang, L.; Tang, Y. Dynamic aggregation for heterogeneous quantization in federated learning. *IEEE Transactions on Wireless Communications* **2021**, *20*, 6804–6819.
10. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Convergence time optimization for federated learning over wireless networks. *IEEE Transactions on Wireless Communications* **2020**, *20*, 2457–2471.
11. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2020, pp. 2021–2031.
12. Sery, T.; Cohen, K. On analog gradient descent learning over multiple access fading channels. *IEEE Transactions on Signal Processing* **2020**, *68*, 2897–2911.
13. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems* **2020**, *31*, 3400–3413.
14. Zhu, Z.; Hong, J.; Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 12878–12889.
15. Jiang, D.; Shan, C.; Zhang, Z. Federated learning algorithm based on knowledge distillation. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE). IEEE, 2020, pp. 163–167.
16. Li, C.; Li, G.; Varshney, P.K. Communication-efficient federated learning based on compressed sensing. *IEEE Internet of Things Journal* **2021**, *8*, 15531–15541.
17. Zeng, Q.; Du, Y.; Huang, K.; Leung, K.K. Energy-efficient radio resource allocation for federated edge learning. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020, pp. 1–6.
18. Philip, J.M.; Durga, S.; Esther, D. Deep learning application in IOT health care: a survey. In *Intelligence in Big Data Technologies—Beyond the Hype*; Springer, 2021; pp. 199–208.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

19. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* **2021**, *14*, 1–210.
20. Lyu, L.; Yu, H.; Yang, Q. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* **2020**.
21. Ozfatura, E.; Ozfatura, K.; Gündüz, D. Time-correlated sparsification for communication-efficient federated learning. In Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT). IEEE, 2021, pp. 461–466.
22. Ahn, J.H.; Simeone, O.; Kang, J. Cooperative learning via federated distillation over fading channels. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 8856–8860.
23. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal* **2020**, *7*, 7751–7763.
24. Xiao, Y.; Shi, G.; Krunz, M. Towards ubiquitous AI in 6G with federated learning. *arXiv preprint arXiv:2004.13563* **2020**.
25. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* **2017**.
26. Xia, S.; Zhu, J.; Yang, Y.; Zhou, Y.; Shi, Y.; Chen, W. Fast convergence algorithm for analog federated learning. In Proceedings of the ICC 2021-IEEE International Conference on Communications. IEEE, 2021, pp. 1–6.
27. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering* **2021**.
28. Guo, Y.; Sun, Y.; Hu, R.; Gong, Y. Hybrid Local SGD for Federated Learning with Heterogeneous Communications. In Proceedings of the International Conference on Learning Representations, 2022.
29. Zamir, R.; Feder, M. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory* **1992**, *38*, 428–436.
30. Zhao, Z.; Luo, M.; Ding, W. Deep Leakage from Model in Federated Learning. 2022.
31. Bernstein, J.; Wang, Y.X.; Azizzadenesheli, K.; Anandkumar, A. signSGD: Compressed optimisation for non-convex problems. In Proceedings of the International Conference on Machine Learning. PMLR, 2018, pp. 560–569.
32. Zhang, L.; Wu, D.; Yuan, X. FedZKT: Zero-Shot Knowledge Transfer towards Resource-Constrained Federated Learning with Heterogeneous On-Device Models. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). IEEE, 2022, pp. 928–938.
33. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019-2019 IEEE international conference on communications (ICC). IEEE, 2019, pp. 1–7.
34. Yu, C.; Tang, H.; Renggli, C.; Kassing, S.; Singla, A.; Alistarh, D.; Zhang, C.; Liu, J. Distributed learning over unreliable networks. In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 7202–7212.
35. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Sparse Binary Compression: Towards Distributed Deep Learning with minimal Communication. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.
36. Rothchild, D.; Panda, A.; Ullah, E.; Ivkin, N.; Stoica, I.; Braverman, V.; Gonzalez, J.; Arora, R. FetchSGD: Communication-Efficient Federated Learning with Sketching. In Proceedings of the Proceedings of the 37th International Conference on Machine Learning; III, H.D.; Singh, A., Eds. PMLR, 13–18 Jul 2020, Vol. 119, *Proceedings of Machine Learning Research*, pp. 8253–8265.
37. Hyeon-Woo, N.; Ye-Bin, M.; Oh, T.H. FedPara: Low-Rank Hadamard Product for Communication-Efficient Federated Learning. *arXiv preprint arXiv:2108.06098* **2021**.
38. *Study on enablers for network automation for the 5G System (5GS)*; 3GPP TR 23.700-91, 2020.
39. Salehi, M.; Hossain, E. Federated learning in unreliable and resource-constrained cellular wireless networks. *IEEE Transactions on Communications* **2021**, *69*, 5136–5151.
40. Abdi, A.; Fekri, F. Quantized compressive sampling of stochastic gradients for efficient communication in distributed deep learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 3105–3112.
41. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Generation Computer Systems* **2021**, *115*, 619–640.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

42. Ye, H.; Liang, L.; Li, G.Y. Decentralized federated learning with unreliable communications. *IEEE Journal of Selected Topics in Signal Processing* **2022**, *16*, 487–500.
43. Cho, Y.J.; Manoel, A.; Joshi, G.; Sim, R.; Dimitriadis, D. Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning. *arXiv preprint arXiv:2204.12703* **2022**.
44. He, Y.; Zenk, M.; Fritz, M. CosSGD: Nonlinear Quantization for Communication-efficient Federated Learning. *CoRR* **2020**, *abs/2012.08241*, [2012.08241].
45. Mao, Y.; Zhao, Z.; Yang, M.; Liang, L.; Liu, Y.; Ding, W.; Lan, T.; Zhang, X.P. SAFARI: Sparsity enabled Federated Learning with Limited and Unreliable Communications. *arXiv preprint arXiv:2204.02321* **2022**.
46. Lin, Z.; Liu, H.; Zhang, Y.J.A. Relay-assisted cooperative federated learning. *IEEE Transactions on Wireless Communications* **2022**.
47. Shi, H.; Zhang, Y.; Shen, Z.; Tang, S.; Li, Y.; Guo, Y.; Zhuang, Y. Towards Communication-Efficient and Privacy-Preserving Federated Representation Learning. *arXiv preprint arXiv:2109.14611* **2021**.
48. Jia, X.; Song, S.; He, W.; Wang, Y.; Rong, H.; Zhou, F.; Xie, L.; Guo, Z.; Yang, Y.; Yu, L.; et al. Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. *arXiv preprint arXiv:1807.11205* **2018**.
49. Oh, S.; Park, J.; Jeong, E.; Kim, H.; Bennis, M.; Kim, S.L. Mix2FLD: Downlink federated learning after uplink federated distillation with two-way mixup. *IEEE Communications Letters* **2020**, *24*, 2211–2215.
50. Wu, C.; Wu, F.; Liu, R.; Lyu, L.; Huang, Y.; Xie, X. Fedkd: Communication efficient federated learning via knowledge distillation. *arXiv preprint arXiv:2108.13323* **2021**.
51. Yao, D.; Pan, W.; Dai, Y.; Wan, Y.; Ding, X.; Jin, H.; Xu, Z.; Sun, L. Local-Global Knowledge Distillation in Heterogeneous Federated Learning with Non-IID Data. *arXiv preprint arXiv:2107.00051* **2021**.
52. Aji, A.F.; Heafield, K. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021* **2017**.
53. Nori, M.K.; Yun, S.; Kim, I.M. Fast federated learning by balancing communication trade-offs. *IEEE Transactions on Communications* **2021**, *69*, 5168–5182.
54. Shahid, O.; Pouriyeh, S.; Parizi, R.M.; Sheng, Q.Z.; Srivastava, G.; Zhao, L. Communication efficiency in federated learning: Achievements and challenges. *arXiv preprint arXiv:2107.10996* **2021**.
55. Amiri, M.M.; Gunduz, D.; Kulkarni, S.R.; Poor, H.V. Federated learning with quantized global model updates. *arXiv preprint arXiv:2006.10672* **2020**.
56. Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; Vojnovic, M. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems* **2017**, *30*.
57. Li, Y.; Yu, M.; Li, S.; Avestimehr, S.; Kim, N.S.; Schwing, A. Pipe-SGD: A decentralized pipelined SGD framework for distributed deep net training. *Advances in Neural Information Processing Systems* **2018**, *31*.
58. Basat, R.B.; Vargaftik, S.; Portnoy, A.; Einziger, G.; Ben-Itzhak, Y.; Mitzenmacher, M. QUICK-FL: Quick Unbiased Compression for Federated Learning. *arXiv preprint arXiv:2205.13341* **2022**.
59. Lin, Y.; Han, S.; Mao, H.; Wang, Y.; Dally, W.J. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887* **2017**.
60. Arandjelovic, O.; Shakhnarovich, G.; Fisher, J.; Cipolla, R.; Darrell, T. Face recognition with image sets using manifold density divergence. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, Vol. 1, pp. 581–588.
61. Zhu, Z.; Hong, J.; Drew, S.; Zhou, J. Resilient and Communication Efficient Learning for Heterogeneous Federated Systems. In Proceedings of the International Conference on Machine Learning. PMLR, 2022, pp. 27504–27526.
62. Eghlidi, N.F.; Jaggi, M. Sparse communication for training deep networks. *arXiv preprint arXiv:2009.09271* **2020**.
63. Sun, C.; Jiang, T.; Zonouz, S.; Pompili, D. Fed2KD: Heterogeneous Federated Learning for Pandemic Risk Assessment via Two-Way Knowledge Distillation. In Proceedings of the 2022 17th Wireless On-Demand Network Systems and Services Conference (WONS). IEEE, 2022, pp. 1–8.
64. Vargaftik, S.; Basat, R.B.; Portnoy, A.; Mendelson, G.; Itzhak, Y.B.; Mitzenmacher, M. Eden: Communication-efficient and robust distributed mean estimation for federated learning. In Proceedings of the International Conference on Machine Learning. PMLR, 2022, pp. 21984–22014.
65. Shlezinger, N.; Chen, M.; Eldar, Y.C.; Poor, H.V.; Cui, S. Federated learning with quantization constraints. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 8851–8855.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

66. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M. Energy efficient federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications* **2020**, *20*, 1935–1949.
67. Jin, R.; Huang, Y.; He, X.; Dai, H.; Wu, T. Stochastic-sign SGD for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940* **2020**.
68. Zhu, G.; Wang, Y.; Huang, K. Broadband analog aggregation for low-latency federated edge learning. *IEEE Transactions on Wireless Communications* **2019**, *19*, 491–506.
69. Muhammad, K.; Ullah, A.; Lloret, J.; Del Ser, J.; de Albuquerque, V.H.C. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems* **2020**, *22*, 4316–4336.
70. Shi, S.; Tang, Z.; Wang, Q.; Zhao, K.; Chu, X. Layer-wise adaptive gradient sparsification for distributed deep learning with convergence guarantees. *arXiv preprint arXiv:1911.08727* **2019**.
71. Guo, H.; Liu, A.; Lau, V.K. Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis. *IEEE Internet of Things Journal* **2020**, *8*, 197–210.
72. Xu, H.; Ho, C.Y.; Abdelmoniem, A.M.; Dutta, A.; Bergou, E.H.; Karatsenidis, K.; Canini, M.; Kalnis, P. Compressed communication for distributed deep learning: Survey and quantitative evaluation. Technical report, 2020.
73. Zhang, N.; Tao, M. Gradient statistics aware power control for over-the-air federated learning in fading channels. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020, pp. 1–6.
74. Ryabinin, M.; Gorbunov, E.; Plokhotnyuk, V.; Pekhimenko, G. Moshpit sgd: Communication-efficient decentralized training on heterogeneous unreliable devices. *Advances in Neural Information Processing Systems* **2021**, *34*, 18195–18211.
75. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
76. Shi, S.; Wang, Q.; Zhao, K.; Tang, Z.; Wang, Y.; Huang, X.; Chu, X. A Distributed Synchronous SGD Algorithm with Global Top-k Sparsification for Low Bandwidth Networks. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 2238–2247.
77. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*; Springer, 2010; pp. 177–186.
78. Feynman, R.; Vernon Jr., F. The theory of a general quantum system interacting with a linear dissipative system. *Annals of Physics* **1963**, *24*, 118–173. https://doi.org/10.1016/0003-4916(63)90068-X.
79. Sun, J.; Chen, T.; Giannakis, G.B.; Yang, Q.; Yang, Z. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**.
80. You, Y.; Buluç, A.; Demmel, J. Scaling deep learning on gpu and knights landing clusters. In Proceedings of the Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, pp. 1–12.
81. Chen, M.; Mao, B.; Ma, T. FedSA: A staleness-aware asynchronous Federated Learning algorithm with non-IID data. *Future Generation Computer Systems* **2021**, *120*, 1–12.
82. Li, T.; Sanjabi, M.; Beirami, A.; Smith, V. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497* **2019**.
83. Harlap, A.; Narayanan, D.; Phanishayee, A.; Seshadri, V.; Devanur, N.; Ganger, G.; Gibbons, P. Pipedream: Fast and efficient pipeline parallel dnn training. *arXiv preprint arXiv:1806.03377* **2018**.
84. Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* **2018**.
85. Yu, C.; Shen, S.; Zhang, K.; Zhao, H.; Shi, Y. Energy-Aware Device Scheduling for Joint Federated Learning in Edge-assisted Internet of Agriculture Things. In Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022, pp. 1140–1145.
86. Xu, H.; Kostopoulou, K.; Dutta, A.; Li, X.; Ntoulas, A.; Kalnis, P. DeepReduce: A Sparse-tensor Communication Framework for Federated Deep Learning. In Proceedings of the Advances in Neural Information Processing Systems; Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; Vaughan, J.W., Eds. Curran Associates, Inc., 2021, Vol. 34, pp. 21150–21163.
87. Sahoo, A.K.; Pradhan, C.; Barik, R.K.; Dubey, H. DeepReco: deep learning based health recommender system using collaborative filtering. *Computation* **2019**, *7*, 25.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

88. Imteaj, A.; Amini, M.H. Fedar: Activity and resource-aware federated learning model for distributed mobile robots. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020, pp. 1153–1160.
89. Yang, Z.; Chen, M.; Wong, K.K.; Poor, H.V.; Cui, S. Federated learning for 6G: Applications, challenges, and opportunities. *Engineering* **2021**.
90. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks. *IEEE Transactions on Industrial Informatics* **2020**, *17*, 5098–5107.
91. Shi, S.; Chu, X.; Li, B. MG-WFBP: Efficient data communication for distributed synchronous SGD algorithms. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications. IEEE, 2019, pp. 172–180.
92. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* **2020**, *37*, 362–386.
93. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security* **2019**, *15*, 911–926.
94. Hellström, H.; Fodor, V.; Fischione, C. Over-the-Air Federated Learning with Retransmissions. In Proceedings of the 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2021, pp. 291–295.
95. Shi, S.; Wang, Q.; Chu, X.; Li, B.; Qin, Y.; Liu, R.; Zhao, X. Communication-Efficient Distributed Deep Learning with Merged Gradient Sparsification on GPUs. In Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 406–415.
96. Mishchenko, K.; Wang, B.; Kovalev, D.; Richtárik, P. IntSGD: Adaptive Floatless Compression of Stochastic Gradients. In Proceedings of the International Conference on Learning Representations, 2022.
97. Lin, T.; Kong, L.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems* **2020**, *33*, 2351–2363.
98. Kairouz, P.; Oh, S.; Viswanath, P. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems* **2014**, *27*.
99. Mahmoudi, A.; Ghadikolaei, H.S.; Júnior, J.M.B.D.S.; Fischione, C. FedCau: A Proactive Stop Policy for Communication and Computation Efficient Federated Learning. *arXiv preprint arXiv:2204.07773* **2022**.
100. Yang, H.; Qiu, P.; Liu, J.; Yener, A. Over-the-Air Federated Learning with Joint Adaptive Computation and Power Control. *arXiv preprint arXiv:2205.05867* **2022**.
101. Itahara, S.; Nishio, T.; Koda, Y.; Morikura, M.; Yamamoto, K. Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training with Non-IID Private Data. *IEEE Transactions on Mobile Computing* **2021**.
102. Wang, H.; Sievert, S.; Liu, S.; Charles, Z.; Papailiopoulos, D.; Wright, S. Atomo: Communication-efficient learning via atomic sparsification. *Advances in Neural Information Processing Systems* **2018**, *31*.
103. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial intelligence and statistics. PMLR, 2017, pp. 1273–1282.
104. Du, W.; Atallah, M.J. Secure multi-party computation problems and their applications: a review and open problems. In Proceedings of the Proceedings of the 2001 workshop on New security paradigms, 2001, pp. 13–22.
105. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* **2017**, *105*, 2295–2329.
106. Wu, C.; Zhu, S.; Mitra, P. Federated Unlearning with Knowledge Distillation. *arXiv preprint arXiv:2201.09441* **2022**.
107. Sun, L.; Lyu, L. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537* **2020**.
108. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M.; Poor, H.V.; Cui, S. Delay minimization for federated learning over wireless communication networks. *arXiv preprint arXiv:2007.03462* **2020**.
109. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* **2019**, *37*, 1205–1221.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

110. Mashhadi, M.B.; Shlezinger, N.; Eldar, Y.C.; Gündüz, D. Fedrec: Federated learning of universal receivers over fading channels. In Proceedings of the 2021 IEEE Statistical Signal Processing Workshop (SSP). IEEE, 2021, pp. 576–580.
111. Zhang, X.; Hong, M.; Dhople, S.; Yin, W.; Liu, Y. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418* **2020**.
112. Philippenko, C.; Dieuleveut, A. Bidirectional compression in heterogeneous settings for distributed or federated learning with partial participation: tight convergence guarantees. *arXiv preprint arXiv:2006.14591* **2020**.
113. Amiri, M.M.; Gündüz, D.; Kulkarni, S.R.; Poor, H.V. Convergence of federated learning over a noisy downlink. *IEEE Transactions on Wireless Communications* **2021**, *21*, 1422–1437.
114. Liu, L.; Zhang, J.; Song, S.; Letaief, K.B. Communication-Efficient Federated Distillation with Active Data Sampling. *arXiv preprint arXiv:2203.06900* **2022**.
115. Sattler, F.; Korjakow, T.; Rischke, R.; Samek, W. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems* **2021**.
116. Li, D.; Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* **2019**.
117. Ozkara, K.; Singh, N.; Data, D.; Diggavi, S. QuPeD: Quantized Personalization via Distillation with Applications to Federated Learning. *Advances in Neural Information Processing Systems* **2021**, *34*, 3622–3634.
118. Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.J.; Zhang, W.; Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems* **2017**, *30*.
119. Sahu, A.; Dutta, A.; M Abdelmoniem, A.; Banerjee, T.; Canini, M.; Kalnis, P. Rethinking gradient sparsification as total error minimization. *Advances in Neural Information Processing Systems* **2021**, *34*, 8133–8146.
120. Liu, Y.; Yuan, X.; Xiong, Z.; Kang, J.; Wang, X.; Niyato, D. Federated learning for 6G communications: Challenges, methods, and future directions. *China Communications* **2020**, *17*, 105–118.
121. Zhang, L.; Shen, L.; Ding, L.; Tao, D.; Duan, L.Y. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10174–10183.
122. Li, S.; Qi, Q.; Wang, J.; Sun, H.; Li, Y.; Yu, F.R. GGS: General Gradient Sparsification for Federated Learning in Edge Computing. In Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–7.
123. Fan, X.; Wang, Y.; Huo, Y.; Tian, Z. Communication-efficient federated learning through 1-bit compressive sensing and analog aggregation. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2021, pp. 1–6.
124. Chen, M.; Shlezinger, N.; Poor, H.V.; Eldar, Y.C.; Cui, S. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2024789118.
125. Das, R.; Acharya, A.; Hashemi, A.; Sanghavi, S.; Dhillon, I.S.; Topcu, U. Faster non-convex federated learning via global and local momentum. *arXiv preprint arXiv:2012.04061* **2020**.
126. Cha, H.; Park, J.; Kim, H.; Kim, S.L.; Bennis, M. Federated reinforcement distillation with proxy experience memory. *arXiv preprint arXiv:1907.06536* **2019**.
127. Stich, S.U.; Cordonnier, J.B.; Jaggi, M. Sparsified SGD with memory. *Advances in Neural Information Processing Systems* **2018**, *31*.
128. Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics* **2018**, *19*, 1236–1246.
129. Wu, W.; He, L.; Lin, W.; Mao, R.; Maple, C.; Jarvis, S. SAFA: A semi-asynchronous protocol for fast federated learning with low overhead. *IEEE Transactions on Computers* **2020**, *70*, 655–668.
130. Zhang, X.; Zhu, X.; Wang, J.; Yan, H.; Chen, H.; Bao, W. Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks. *Information Sciences* **2020**, *540*, 242–262.
131. Shi, S.; Chu, X.; Cheung, K.C.; See, S. Understanding top-k sparsification in distributed deep learning. *arXiv preprint arXiv:1911.08772* **2019**.
132. Marfoq, O.; Xu, C.; Neglia, G.; Vidal, R. Throughput-optimal topology design for cross-silo federated learning. *Advances in Neural Information Processing Systems* **2020**, *33*, 19478–19487.
133. Malekijoo, A.; Fadaeieslam, M.J.; Malekijou, H.; Homayounfar, M.; Alizadeh-Shabdiz, F.; Rawassizadeh, R. Fedzip: A compression framework for communication-efficient federated learning. *arXiv preprint arXiv:2102.01593* **2021**.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2025 doi:10.20944/preprints202507.1037.v1

134. Han, P.; Wang, S.; Leung, K.K. Adaptive Gradient Sparsification for Efficient Federated Learning: An Online Learning Approach. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), 2020, pp. 300–310.
135. Ahn, J.H.; Simeone, O.; Kang, J. Wireless federated distillation for distributed edge learning with heterogeneous data. In Proceedings of the 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). IEEE, 2019, pp. 1–6.
136. Al-Qizwini, M.; Barjasteh, I.; Al-Qassab, H.; Radha, H. Deep learning algorithm for autonomous driving using googlenet. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017, pp. 89–96.
137. Zaw, C.W.; Pandey, S.R.; Kim, K.; Hong, C.S. Energy-aware resource management for federated learning in multi-access edge computing systems. *IEEE Access* **2021**, *9*, 34938–34950.
138. Amiri, M.M.; Gündüz, D. Federated learning over wireless fading channels. *IEEE Transactions on Wireless Communications* **2020**, *19*, 3546–3557.
139. Cha, H.; Park, J.; Kim, H.; Bennis, M.; Kim, S.L. Proxy experience replay: Federated distillation for distributed reinforcement learning. *IEEE Intelligent Systems* **2020**, *35*, 94–101.
140. Chetlur, S.; Woolley, C.; Vandermersch, P.; Cohen, J.; Tran, J.; Catanzaro, B.; Shelhamer, E. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759* **2014**.
141. Shlezinger, N.; Chen, M.; Eldar, Y.C.; Poor, H.V.; Cui, S. UVeQFed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing* **2020**, *69*, 500–514.
142. Wangni, J.; Wang, J.; Liu, J.; Zhang, T. Gradient s for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems* **2018**, *31*.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.