CSIAM Trans. Appl. Math.
doi:

Vol. **x**, No. 1, pp. 1-46
February 202x

Review of Mathematical Optimization in Federated Learning

Shusen Yang[1,2], Fangyuan Zhao[2], Zihao Zhou[1], Liang Shi[2], Xuebin Ren[2], and Zongben Xu [1,2]

[1] *School of Mathematics and Statistics, Xi'an Jiaotong University, China.*
[2] *National Engineering Laboratory for Big Data Analytics, Xi'an Jiaotong University, China.*

Abstract. Federated Learning (FL) has been becoming a popular interdisciplinary research area in both applied mathematics and information sciences. Mathematically, FL aims to collaboratively optimize aggregate objective functions over distributed datasets while satisfying a variety of privacy and system constraints. Different from conventional distributed optimization methods, FL needs to address several specific issues (e.g., non-i.i.d. data distributions and differential private noises), which pose a set of new challenges in the problem formulation, algorithm design, and convergence analysis. In this paper, we will systematically review existing FL optimization research including their assumptions, formulations, methods, and theoretical results. Potential future directions are also discussed.

AMS subject classifications: 90C26, 90C31, 68W15, 68T05

Key words: Federated Learning, distributed optimization, convergence analysis, error bounds.

1 Introduction

With the increasingly stringent privacy regulations [1, 2], data isolation has been becoming the key bottleneck of data sciences and artificial intelligence. To address this issue, Federated learning (FL) emerges as a popular privacy-preserving distributed Machine Learning (ML) paradigm, which enables multiple data owners to jointly train ML models without sharing the raw data [3–5]. It has gained extensive interests from both academia and industry, and demonstrated great success across multiple domains, including medicine [6], finance [7], and industry [8], etc.

Mathematically, FL training tasks are essentially distributed optimization problems, which aim to minimize aggregate global objectives (e.g., the mean empirical loss), across a set of distributed data owners by exchanging model parameters trained on their local datasets [9–11]. Despite being similar in essence, FL optimization has many distinct characteristics from traditional distributed optimization. Their main difference lies in the

communication environment. Specifically, traditional distributed optimization, mainly in the form of distributed ML [12–14] (some used in distributed resource allocation [15–17]), is often used for high-throughput ML speedup in data centers. Here, multiple homogeneous computing nodes with uniformly distributed data partitions are connected by reliable networks with Gigabytes of bandwidth. However, FL is often applied to achieve collaborative and privacy-preserving ML in wide-area networks, where geographically distributed clients with naturally generated data collaborate to train ML models over bandwidth-constrained communication channels.

Due to the drastic differences, FL needs to address a variety of specific and complicated issues in optimization. For example, naturally generated data at the FL clients are commonly heterogeneous, i.e., non-balanced and non-i.i.d. distributed, which leads to biased model aggregation. Considering limited communication bandwidth, FL often performs multiple local updates before the global aggregation, further amplifying the model bias caused by data heterogeneity. Furthermore, model dissemination and aggregation give rise to concerns of private information leakage and falsification. The decentralized communication architecture also leads to partially local approximations of global objective functions. These issues pose new challenges in optimization in FL, including problem formulation, algorithm design, and convergence analysis. In particular, typical challenges are summarized as follows:

- **Biased local objectives from non-i.i.d. datasets.** The potentially unknown and non-i.i.d. data distributions among distributed nodes* could result in biased local objectives [5, 18]. Thus, the global objective function in FL optimization cannot be decomposed into trainable local objectives without bias. This means that the gradients of local models may largely deviate from the steepest descent direction of the global objective, leading to significant degradation of convergence speed and model accuracy [19].
- **Perturbed gradients with DP noises.** The Differential Privacy (DP) mechanisms [20] are commonly adopted in the FL optimization process to protect exchanged parameters or gradients, by introducing statistically unbiased noises (e.g., Gaussian [21] or Laplacian [22] noises). Despite the statistical unbiasedness, random noises may overwhelm the useful gradient information, causing unstable and slow convergence towards sub-optimal solutions [23–25].
- **Partially approximated objectives under decentralized topologies.** In FL systems with decentralized typologies, the model exchange has to rely on limited peer-to-peer neighborhood communication. The global objective for each data owner is thus partially approximated by consolidating the local objectives of their adjacent neighbors [26]. Consequently, the aggregated gradient in each step is also a partial approximation to that of the global objective. The approximation error can lead to significantly slower convergence [3, 27], especially in non-i.i.d. scenarios.

*The terms "node", "client" and "data owner" are used interchangeably within this survey.

- **Obsolete solutions in online settings.** Online FL optimization aims to learn a series of global functions with minimized cumulative losses from distributed sequential data [28–31]. However, the coming data may have a time-evolving distribution, resulting in inevitable generalization errors and instability of models on new data. Meanwhile, the inherent time-varying and cumulative constraints, e.g. communication, computing, and memory capabilities, may also complicate the problem modeling and solving [32].

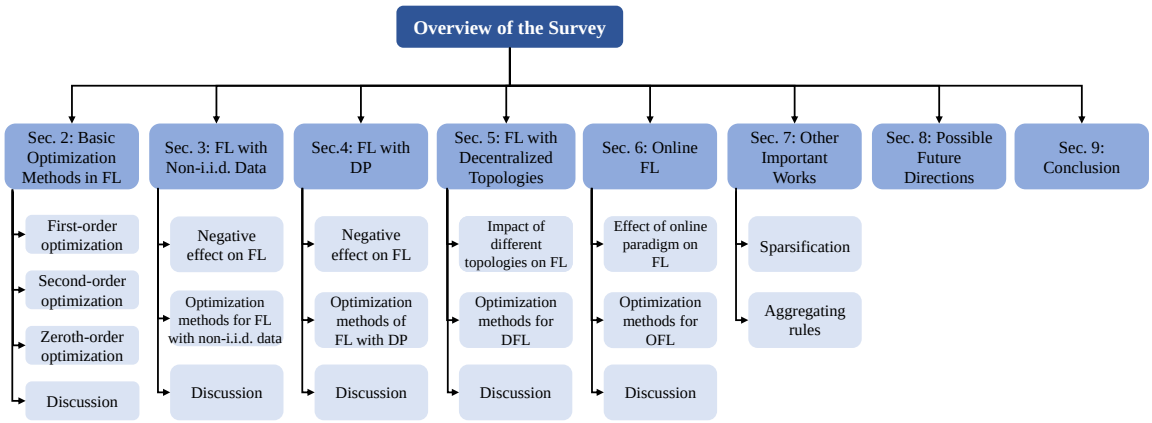


Figure 1: The overview of the survey.

This paper presents a systematic survey of the mathematical optimization in FL, summarizing the assumptions, problem formulations, optimization methods, and theoretical results. It is worth noting that although there exist several FL surveys [5, 33–37], all of them are presented from the perspective of information sciences, including the FL architectures [33], algorithms [34], and applications [38], rather than mathematical optimizations like our survey. Fig. 1 illustrates the overview of this survey. We first review typical optimization methods in FL in Section 2. The optimization challenges and solutions when dealing with non-i.i.d. data, DP noises, decentralized network topologies, and online FL optimizations are presented in Sections 3-6, respectively. We then summarize other important works in Section 7 and discuss possible future directions in Section 8. Finally, we conclude the survey in Section 9.

2 Basic optimization methods in Federated Learning

The common FL optimization is mostly studied in the context of horizontal FL (HFL) where participants possess the same feature space but different samples [5, 39]. We will

Algorithm 1 Algorithmic procedures of FL optimization

Input: Client number N, objective function $f(x)=\frac{1}{N}\sum_{i=1}^{N} f_i(x)$, initialized x_0, total communication rounds T, local learning rate η_l, global learning rate η_g, number of local updates E in one round, distributed dataset $\{\xi_1,\ldots,\xi_N\}$.

Output: x^{T+1}.

1: **for** $t\in\{1,\ldots,T\}$ **do**
2: Server samples a subset S^t of K clients and server sends x^t to these clients;
3: **for** each client $i\in S^t$ **do**
4: x_i^{t+1}, ∇_i^t=**LocalOPT**(x^t,η_l,E,ξ_i);
5: **end for**
6: Server aggregates $\nabla^t=\frac{1}{|S^t|}\sum_{i\in S^t}\nabla_i^t$;
7: Update global model $x^{t+1}=x^t+\eta_g\nabla^t$.
8: **end for**

use HFL as the default FL setting † to introduce the general workflow of FL optimization.

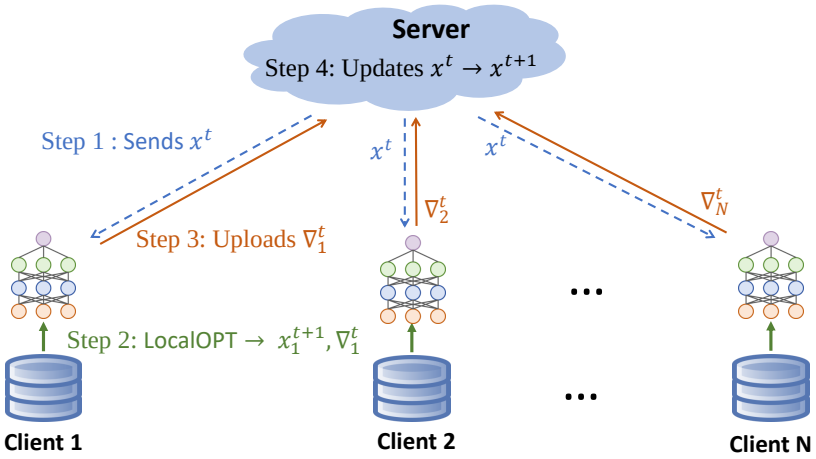


Figure 2: General framework and workflow of FL.

As shown in Algorithm 1 and Fig. 2, a typical FL system consists of multiple distributed clients and one central server. At each iteration round t, the server randomly samples a subset S^t from N clients and broadcasts the current global model $x^t\in\mathbb{R}^d$ to these clients. Each client indexed by i in S^t then executes LocalOPT upon receiving x^t, which involves applying an optimization method to minimize their respective local objectives, subsequently obtaining a local model x_i^{t+1} and uploading the local gradient ∇_i^t to the server. Once all gradients from S^t are received, the server aggregates all local

†Another important category of FL is vertical FL (VFL), where participants share the same sample space but different features. More details can be referred to in [40–42].

gradients to obtain the global gradient ∇^t and accordingly updates the global model as $x^{t+1}=x^t+\eta_g\nabla^t$, where η_g is the global learning rate.

Given its pivotal role in local data information extraction, the choice of LocalOPT would significantly affect the performance of FL training process. Since LocalOPT essentially executes machine learning (ML) tasks in each round on the client side, its candidates encompass all ML optimization methods, including the first-order, second-order, and zeroth-order ones. Each type of these optimization methods exhibits unique convergence characteristics when being adapted to FL settings, which are summarized in Table 1. The following are common assumptions used in the convergence analysis of FL optimization algorithms [11, 43, 44].

- **Lipschitz Objective Function (LOF):** $f(x)$ is β-Lipschitz continuous if there exists $\beta\geq 0$ such that for all $x_1,x_2\in\mathbb{R}^d$,

$$|f(x_1)-f(x_2)|\leq\beta\|x_1-x_2\|. \tag{2.1}$$

- **Smooth Objective Function (SOF):** $f(x)$ is L-smooth if $f(x)$ has L-Lipschitz continuous gradient, i.e., for all $x_1,x_2\in\mathbb{R}^d$,

$$\|\nabla f(x_1)-\nabla f(x_2)\|\leq L\|x_1-x_2\|. \tag{2.2}$$

  When the second-order gradients of $f(x)$ is Lipschitz continuous, we call the condition **Lipschitz Hessian (LH)**.

- **Strongly Convex Objective Function (SCOF):** $f(x)$ is μ-strongly convex if there exists $\mu\geq 0$ such that for all $x_1,x_2\in\mathbb{R}^d$,

$$f(x_1)\geq f(x_2)+(x_1-x_2)^T\nabla f(x_2)+\frac{\mu}{2}\|x_1-x_2\|_2^2. \tag{2.3}$$

- **Convex Objective Function (COF):** $f(x)$ is convex if for all $x_1,x_2\in\mathbb{R}^d$, it holds that

$$f(x_1)\geq f(x_2)+(x_1-x_2)^T\nabla f(x_2). \tag{2.4}$$

- **Coercive Function (CF):** $f(x)$ is coercive if $\lim_{\|x\|\to\infty}f(x)\to\infty$.

- **Bounded Gradient (BG):** The gradient of $f(x)$ is G-bounded if there exists $G\geq 0$ such that for all $x\in\mathbb{R}^d$, $\|\nabla f(x)\|\leq G$.

- **Bounded Variance (BV):** The variance of each stochastic gradient $\nabla f_i(x;\xi)$ is bounded if there exists $\sigma\in\mathbb{R}$, such that

$$\mathbb{E}_\xi\|\nabla f_i(x;\xi)-\nabla f_i(x)\|^2\leq\sigma^2, \tag{2.5}$$

  where $f_i(\cdot)$ denotes the local objective function of the i-th client, x is the current model parameter and ξ is the data sampled in the current round of local training.

- **Bounded Gradient Dissimilarity (BGD):** Local gradients $\{\nabla f_i(x), \forall i\}$ satisfy (G,B)-bounded gradient dissimilarity if there exist constants $G \geq 0$ and $B \geq 1$ such that

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(x)\|^2 \leq G^2 + B^2\|\nabla f(x)\|^2, \quad \forall x \in \mathbb{R}^d. \tag{2.6}$$

In above assumptions, LOF, SOF, and LH describe the smoothness of the objective function. SCOF and COF characterize the convexity of objective functions. CF ensures that the objective function has a global minimum. BG, BV, and BGD capture the properties of gradients.

Table 1: Comparison of basic optimization methods in FL.

Categories	Methods	Assumptions	Convergence rates
First-order Methods	SGD (e.g.,[45, 46])	SOF, BV, BG	$\mathcal{O}(\frac{\sigma}{\sqrt{KT}})$
	Momentum-based SGD (e.g., [47, 48])	SOF, BV	$\mathcal{O}(\frac{\sigma}{\sqrt{KT}})$
	Adam (e.g., [44, 49])	SOF, BV, BG	$\mathcal{O}(\frac{\sigma}{\sqrt{KT}})$
	ADMM (e.g., [50–52])	SOF, CF	$\mathcal{O}(\frac{1}{T})$
Second-order Methods	Newton (e.g., [53–56])	SOF, LH, SCOF	$\mathcal{O}(\gamma^{-2^T})(\gamma > 1)$
Zeroth-order Methods	ZO-SGD (e.g., [57–59])	LOF	$\mathcal{O}(\sigma\sqrt{\frac{d}{KT}})$

2.1 First-order optimization methods

First-order optimization methods [60] rely only on first-order gradients for the model updating in LocalOPT. They are widely adopted in FL due to their lower computational requirements for gradient estimation, especially in settings with pronounced computational heterogeneity. Here, we introduce several representative first-order optimization methods in FL.

The most common choice for the first-order method is Stochastic Gradient Descent (SGD) [19, 27, 39, 61–63]. SGD works by iteratively updating the model as

$$x^{t+1} = x^t - \eta \nabla f(x^t; \xi^t), \tag{2.7}$$

starting from an arbitrary point x^0, where $\nabla f(x^t;\xi^t)$ is the stochastic gradient at x^t estimated on ξ^t. Here, ξ^t can be a single data, or a minibatch of data uniformly sampled from

the whole training data at random. FedAvg [39] is an extensively employed adaptation of SGD to FL settings. In FedAvg, the LocalOPT in Algorithm 1 adopts SGD and the server takes the weighted average of local gradients to update the global model. It has been demonstrated that FedAvg can converge at a sub-linear rate of $\mathcal{O}(\frac{\sigma}{\sqrt{KT}})$ when using a decaying learning rate [45, 46].

Despite the statistically optimal convergence rate, SGD suffers from the problem of converging to saddle points and difficulty of adjusting its learning rate. These drawbacks are also inherited by FedAvg. Momentum-based SGD methods [47, 64–67] can accelerate model convergence and escape saddle points by incorporating historical gradient information into the current gradient. [48] has shown that introducing Nesterov's method into FL leads to the same theoretical convergence rate as FedAvg. However, when the learning rate satisfies mild conditions, momentum-based SGD achieves faster actual convergence speed. Building upon this, a recent study [44] has further integrated typical methods for adaptive learning rate into momentum-based SGD to achieve higher convergence speed. It has been demonstrated that integrating Adam [68] into FL to optimize the global learning rate η_g can achieve the convergence rate of $\mathcal{O}(\frac{\sigma}{\sqrt{KT}})$ in non-i.i.d. scenarios, matching the rate of FedAvg. Variance reduction methods [19, 69–71] can also be integrated into federated frameworks to reduce the gradient variance term σ in FedAvg's convergence rate. Additionally, the Alternating Direction Method of Multipliers (ADMM) algorithm [50–52] can be used to handle constrained federated optimization problems. Research [50] has shown that ADMM achieves a convergence rate of $\mathcal{O}(\frac{1}{T})$ when dealing with topology-constrained federated optimization problems in decentralized FL.

2.2 Second-order optimization

Second-order optimization methods [72, 73] invest additional computational resources to compute the Hessian matrix [74], thereby offering a detailed representation of the local optimization landscape. Despite the increased computation and communication overhead, they can deliver a more precise optimization path and often surpass first-order methods with faster convergence speed. The mainstream second-order algorithm in FL is the Newton's method [53–56], which typically updates the model parameters by combining the Hessian matrix (e.g.,$\nabla^2 f(x^t)$) and the first-order gradient information:

$$x^{t+1} = x^t - \nabla^2 f(x^t)^{-1} \nabla f(x^t). \tag{2.8}$$

Considering the aggregation process in FL, i.e., $\nabla f(x^t) = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x^t)$, the server updates the global model by

$$x^{t+1} = x^t - \Big(\frac{1}{N}\sum_{i=1}^{N} \nabla^2 f_i(x^t)\Big)^{-1} \Big(\frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x^t)\Big). \tag{2.9}$$

However, as mentioned, the adaptation of Newton's method to FL suffers from heavy communication and computation of the Hessian matrix. To relieve the burden, one intu-

itive approach is to use advanced compression methods such as the top-k selection technique [54]. Furthermore, to avoid computing the inverse of the Hessian matrix, the server can leverage ADMM to approximate Hessian inverse-gradient product $\nabla^2 f(x^t)^{-1}\nabla f(x^t)$ (also called global Newton direction) [53]. Some other iterative methods, like Richardson iteration [55], can also be applied to get local Newton direction, i.e., $\left(\nabla^2 f_i(x^t)\right)^{-1}\nabla f_i(x^t)$, which is regarded as a solution of a linear system. It has been proved that FL with these adapted Newton's methods can converge at the quadratic speed as centralized Newton's methods [53, 55].

2.3 Zeroth-order optimization

The gradient (including the first-order and second-order) information may be inaccessible or computationally prohibitive in some scenarios, e.g., black box models [75] and reinforcement learning [76, 77]. Zeroth-order (ZO) optimization (also called derivative-free optimization) methods [78, 79] works for such scenarios by utilizing the zeroth-order information, i.e., the variation of objective function value along some directions over a mini-batch of data samples. They can naturally be adopted in FL and named as federated ZO [57–59]. Particularly, in federated ZO, the gradients can be approximated by the variation of objective function value as

$$\widetilde{\nabla} f_i\left(x_i^t;\{\xi_{i,m}^t\}_{m=1}^{b_1},\{v_{i,l}^t\}_{l=1}^{b_2},\mu\right)=\frac{1}{b_1 b_2}\sum_{m=1}^{b_1}\sum_{l=1}^{b_2}\frac{dv_{i,l}^t}{\mu}(f_i(x_i^t+\mu v_{i,l}^t;\xi_{i,m}^t)-f_i(x_i^t;\xi_{i,m}^t)), \quad (2.10)$$

where $\{\xi_{i,m}^t\}_{m=1}^{b_1}$ is a set of i.i.d. random samples, $\{v_{i,l}^t\}_{l=1}^{b_2}$ is a set of i.i.d. random direction vectors (sampling from the d-dimensional uniform distribution), and μ is a positive step size. Then the local model update in LocalOPT of federated ZO algorithms can be written as

$$x_i^{t+1}=x_i^t-\eta_l\widetilde{\nabla} f_i\left(x_i^t;\{\xi_{i,m}^t\}_{m=1}^{b_1},\{v_{i,l}^t\}_{l=1}^{b_2},\mu\right). \quad (2.11)$$

Although federated ZO algorithms adopt a similar framework to FedAvg, the convergence analysis of federated ZO algorithms is slightly different from that of FedAvg. That is because that the gradient estimator of ZO algorithms does not preserve the unbiasedness of stochastic gradients, i.e., $\widetilde{\nabla} f_i\left(x_i^t;\{\xi_{i,m}^t\}_{m=1}^{b_1},\{v_{i,l}^t\}_{l=1}^{b_2},\mu\right)$ is a biased approximation to the real gradient $\nabla f_i(x_i^t)$ [57, 79]. Therefore, in the analysis of federated ZO algorithms, the gradient estimator is usually decomposed into two components, i.e., the difference between the real gradient and its expected estimator, the divergence between the expected and its ZO-approximated estimator [80]. It has been shown that federated ZO algorithms can achieve sub-linear convergence under the assumptions of non-smooth convex loss functions in both i.i.d. and non-i.i.d. settings [58]. Notably, under the assumption of L-smooth local objective functions, federated ZO algorithms can also achieve sub-linear convergence in the non-convex setting [59].

2.4 Discussion

In federated optimization, one criterion for selecting appropriate optimization methods is the trade-off between iteration complexity and communication cost [81]. A larger volume of communication data, such as the Hessian matrix, can reduce the total number of iterations, leading to faster model convergence. On the other hand, first-order methods with higher iteration complexity, often incur lower communication costs. Zeroth-order methods are primarily employed in specific FL scenarios where gradients are unavailable, showing no significant advantage in communication or iteration complexity.

3 Federated Learning with Non-i.i.d. Data

Table 2: Non-i.i.d. classification based on different feature and label distributions [5].

Non-i.i.d. categories	**Probability distributions**
Feature distribution skew	$\mathcal{P}_i(y\|x)=\mathcal{P}_j(y\|x)$, $\mathcal{P}_i(x)\neq\mathcal{P}_j(x)$.
Label distribution skew	$\mathcal{P}_i(x\|y)=\mathcal{P}_j(x\|y)$, $\mathcal{P}_i(y)\neq\mathcal{P}_j(y)$.
Varying features under one label	$\mathcal{P}_i(y)=\mathcal{P}_j(y)$, $\mathcal{P}_i(x\|y)\neq\mathcal{P}_j(x\|y)$.
Identical features with different labels	$\mathcal{P}_i(x)=\mathcal{P}_j(x)$, $\mathcal{P}_i(y\|x)\neq\mathcal{P}_j(y\|x)$.
Data imbalance	Data amount significantly varies across clients.

In FL, data is generated naturally and stored locally, giving rise to data heterogeneity, i.e., non-independent and identically distributed (non-i.i.d.) data across diverse clients. Formally, assume any data sample (x,y) of the i-th client is drawn from a distribution $\mathcal{P}_i(x,y)$, where x and y denote the feature vector and label respectively. $\mathcal{P}_i(x,y)$ can be rewritten as

$$\mathcal{P}_i(x,y)=\mathcal{P}_i(y|x)\mathcal{P}_i(x)=\mathcal{P}_i(x|y)\mathcal{P}_i(y),\quad i\in[N]. \tag{3.1}$$

Then the non-i.i.d. settings in FL can be classified into five categories according to different feature and label distributions [5, 18], which are summarized in Table 2.

3.1 Negative effect of non-i.i.d. data on FL

In real-world FL tasks, the data heterogeneity comes from the complicated mixture of the above five non-i.i.d. issues. Regardless of the category, data heterogeneity inherently leads to the decomposition of the global gradient into multiple biased local gradients over non-i.i.d datasets. The biases in local gradients then compromise the convergence rate and error bound of the global model through the aggregation process [19, 62]. In the following, we will present some theoretical results as well as an intuitive explanation for the performance differences in model convergence between i.i.d. and non-i.i.d. scenarios.

3.1.1 Theoretical results on FL with non-i.i.d. data

In the following, we highlight the degraded model convergence by comparing the theoretical convergence rates and error bounds in both i.i.d and non-i.i.d. settings.

- **Faster convergence to the first-order stationary point in i.i.d. settings.** Under the assumption of the smooth objective function, the convergence results are as follows:
  - **For strongly convex objectives**, traditional distributed algorithms such as local SGD [82] demonstrate a convergence rate of $\mathcal{O}\left(\frac{1}{T}\right)$ towards the optimal point [82, 83].
  - **For non-convex objectives**, the model may get stuck in an extreme point or a saddle point, and require more iterations to escape these points [84]. Therefore, traditional distributed algorithms converge towards the first-order stationary point at a sub-optimal convergence rate of $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ [85].
- **Slower convergence to a neighborhood of the first-order stationary point in non-i.i.d. settings.** In the theoretical analysis, the non-i.i.d. issue is always characterized by the assumption of (G,B)-bounded gradient dissimilarity [19]. By further imposing the smoothness assumption on the objective function, we can obtain the following analytical convergence results:
  - **For strongly convex objectives**, the error bound of FedAvg is $\mathcal{O}\left(\frac{1}{\mu T}+\frac{G^2}{\mu T}+\frac{B^2}{\mu}\right)$, where μ is a coefficient to describe the convexity [19]. This implies that although FedAvg also achieves a convergence rate of $\mathcal{O}(\frac{1}{T})$, non-i.i.d. data distributions still result in a reduced convergence rate and an increased convergence error due to the existence of terms $\frac{1+G^2}{\mu T}$ and $\frac{B^2}{\mu}$ respectively.
  - **For non-convex objectives**, FedAvg exhibits a sub-linear convergence rate of $\mathcal{O}\left(\frac{1+G^2}{\sqrt{T}}+\frac{B^2}{T}\right)$ [19].

3.1.2 Intuitive explanations

Here we provide an intuitive understanding of the negative impact of non-i.i.d. data on the model convergence by comparing the model updating trajectories in the FL optimization process in both i.i.d. and non-i.i.d. settings.

- **Faster convergence to the first-order stationary point in i.i.d. settings.** In ideal i.i.d. settings, as illustrated in Figure 3 (a), the proximity between client optima and global optimum ensures unbiased model updates and consistent convergence under the identical initial models. Consequently, the global model achieves a fast convergence to the first-order stationary point.

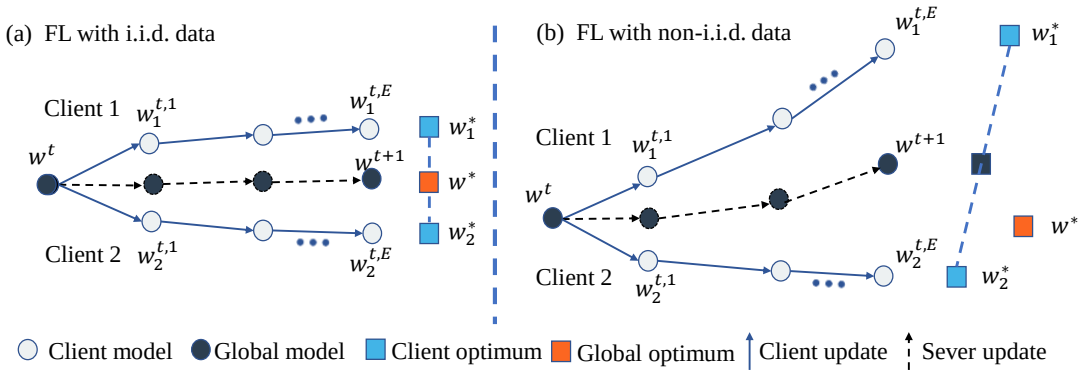


Figure 3: Illustration of model update trajectories under i.i.d. (a) and non-i.i.d. (b) settings in FL for two clients with E local iterations [19].

- **Slower convergence to a neighborhood of the first-order stationary point in non-i.i.d. settings.** In practical non-i.i.d. settings, as illustrated in Figure 3 (b), due to disparities among multiple client optima, FL algorithms such as FedAvg exhibit a decreased convergence rate and increased error bound [19]. On the client side, the local model overfits the local data with biased distribution, causing inevitable biases in gradients [62]. This can be characterized as

$$\mathbb{E}[\nabla f_i(x)] \neq \nabla f(x), \quad i \in [N]. \tag{3.2}$$

  On the server side, the biases cannot be eliminated in the model aggregation and updating process due to the inaccessibility to local data distributions [62]. After multiple training rounds, the biases accumulate, which causes the global model to update along a largely biased direction [86]. Eventually, the non-i.i.d. data significantly degrades the model utility [87].

3.2 Optimization methods for FL with non-i.i.d. data

The above analysis and explanations show the routine along which the biases in data propagate to the model. Existing methodologies [63, 88–90] aim to mitigate the non-i.i.d. issue by correcting the biases in different components along the routine. They can be summarized as follows.

- **Regularization** [63, 89–92] methods manipulate the objective function to constrain the parameter space, ensuring that the optimal parameters across different clients closely resemble one another. Specifically, regularization methods try to constrain the biases by introducing a regularization term related to the distance between the local and global model parameters. The empirical loss function for the i-th client is then represented as

$$\min_x F_i(x, x^{t-1}) := f_i(x) + \alpha l_{\text{reg}}(x, x^{t-1}), \tag{3.3}$$

where α is the regularization parameter and $l_{\text{reg}}(x, x^{t-1})$ is the regularization term, e.g., the L_2-regularization term $\|x-x^{t-1}\|^2$ [63]. The regularization term enforces the local model to remain in a limited neighboring region around the global model and reduces the model deviation in the FL optimization process. It has been proved that the loss reduction in each round is lower bounded by $\rho^t\|\nabla f(x^t)\|^2$ under assumptions of the smooth and non-convex loss function [63], where

$$\rho = \mathcal{O}\left(\frac{1-\gamma}{\alpha} - \frac{\gamma(1+\alpha)}{(\alpha-\theta)\alpha} - \frac{\gamma^2}{(\alpha-\theta)^2}\right), \tag{3.4}$$

γ is a parameter positively correlated with the level of data heterogeneity and θ is a constant. This theoretical result indicates that the convergence accelerates with the increase of α due to the constrained model biases. However, when α exceeds a certain threshold, local models tend to align closely with the received global model and stop learning from local data, thereby slowing down the convergence. By selecting an appropriate α, regularization-based algorithms can achieve a convergence rate of $\mathcal{O}(\frac{1}{T})$ [89].

- **Model interpolation** [91, 93, 94] combines the biased local model and global model to reduce the distances among different local models, improving the trade-off between model personalization and generalization. Specifically, each client trains a composite model $\lambda_i x_i + (1-\lambda_i)x$ in each round, where λ_i is the interpolated weight for the i-th client. Interpolation facilitates the alignment of each client's model towards the global average model, thereby fostering consistency across different local models throughout the convergence process. Theoretical analysis [91] reveals that the interpolation methods can affect the generalization error bound of each local model x_i through a scaling term $\mathcal{O}\left(\sqrt{\frac{\lambda_i^2}{n_i} + \frac{(1-\lambda_i)^2}{n}}\right)$, where n_i and n represent the number of local data and global data, respectively. The generalization error bound scales markedly without interpolation (i.e., λ_i equals 0 or 1). Instead, by setting an appropriate λ, model interpolation may significantly reduce the generalization error bound.

- **Adaptive optimizer** [44] enhances the exploration capability of the optimizer (e.g., SGD in FedAvg) by adaptively adjusting the learning rates of both local and global models, thereby reducing the biases. This is motivated by the theoretical analysis that FedAvg with a fixed learning rate cannot minimize the convergence error incurred by the non-i.i.d. data [44]. The basic idea is to incorporate advanced adaptive optimizers, e.g., Adagrad [95], Yogi [96], and Adam, into the FedAvg framework. Taking Adam as an example, the local and global learning rate is set as $\mathcal{O}(\frac{1}{L\sqrt{T}})$ and $\eta_g = \mathcal{O}(\frac{\sqrt{K}}{\sqrt{v_t}+\frac{G}{L}})$ respectively, and the global updating rule is

$$x^{t+1} = x^t + \eta_g m^t, \tag{3.5}$$

where m^t denotes the momentum and v^t is a time-varying term that combines the historical and current gradient information. Then the convergence rate of the global model in non-i.i.d. settings can reach $\mathcal{O}(\frac{1}{\sqrt{KT}})$, under the assumptions of non-convex and L-smooth objective function and G-bounded gradients. This rate matches the standard convergence rate in i.i.d. settings [97].

- **Variance reduction** [19, 69–71] methods reduce the variance of local gradients among clients, enhancing the consistency among each other. They are motivated by some theoretical results that a larger gradient variance induces decreased convergence rate and increased convergence error [19]. Variance reduction is always accomplished by fusing the unbiased global gradient with each local biased gradient. In particular, in each round of local training, each client combines historical unbiased global gradients and newly calculated local gradients to yield the local update [70, 71]. The local updating rule for the i-th client is:

$$x_i^{t+1} = x_i^t - \eta_l \left(\nabla f_i\left(x^t\right) - \nabla f_i\left(x^{t-1}\right) + \frac{1}{N}\sum_{i=1}^{N} \nabla f_i\left(x^{t-1}\right) \right). \tag{3.6}$$

  Theoretical results reveal that applying variance reduction methods in non-i.i.d. settings can achieve a similar convergence performance to i.i.d. settings. That is, the convergence rate of $\mathcal{O}\left(\frac{1}{T}\right)$ for strongly convex objective functions and a sublinear rate of $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$ for non-convex objective functions [19].

- **Momentum** [64, 98, 99] utilizes both the historical information and current model to correct the update direction, thereby alleviating the bias induced by non-i.i.d. data. Specifically, this optimization method aggregates the last round gradients and the current update by assigning an appropriate weight. Existing research [64, 98] has demonstrated that momentum can alleviate client drift and accelerate the convergence of FL. This is attributed to the fact that the accumulated historical information can enhance the representativeness of data from multiple clients and mitigate the inherent biases stemming from non-i.i.d. data. Specifically, the momentum method can achieve a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{NET}} + \frac{1}{T})$ [98, 99] under the assumptions of L-smooth functions.

- **Discrepancy-aware aggregation** methods [100] allocate aggregation weights that exhibit a negative correlation with local discrepancy levels. These methods are driven by the theoretical analysis of the expected gradient error. It suggests that, to minimize the upper bound of gradient error in FedAvg, aggregation weights should demonstrate a negative correlation with local discrepancy levels while being positively correlated with dataset size. In the implementation, each client first computes its local discrepancy using the local category distribution. Then, the server assigns distinctive aggregation weights based on these discrepancy values.

Table 3: Summary of optimization methods for FL with non-i.i.d. data.

Optimized components	**Methods**	**Advantages**	**Disadvantages**
Objective function	Regularization	Easy deployment	Difficulty in selecting appropriate α
Model	Model interpolation	Balance between generalization and personalization	High computation complexity and memory consumption
	Discrepancy-aware aggregation	Comprehensive theoretical foundation	Difficulty in selecting weight-related hyper-parameters
Optimizer	Adaptive optimizers	Enhanced model exploration capability	High computation complexity and memory consumption
Gradients	Momentum	Enhanced model convergence	High memory consumption
	Variance reduction	Tight analysis in non-i.i.d. settings	High communication cost and memory consumption

3.3 Discussion

As summarized in Table 3, the regularization methods are convenient to deploy since they can be achieved by simply manipulating the objective function. However, it is challenging to select an appropriate regularization parameter α in prior. Model interpolation methods can strike a balance between generalization and personalization, but require additional computational resources to learn the interpolation parameter λ and extra memory to store the global model for each client. Discrepancy-aware aggregation stems from a rigorous theoretical analysis of optimizing error bounds and has a tight theoretical analysis. However, the model performance relies on the properly selected metric for local discrepancy levels and hyper-parameters for discrepancy-aware aggregation weights. Adaptive optimizers enhance model exploration but introduce complexity, risking bias-variance dilemmas and demanding more computational and memory resources. Momentum methods improve the convergence via utilizing the historical information, which however requires additional memory consumption for both clients and the server. Variance reduction methods offer rigorous theoretical guarantees in non-i.i.d. scenarios. However, they necessitate additional communication costs and memory consumption for transmitting and storing historical gradients.

4 Federated Learning with Differential Privacy

Differential Privacy (DP) provides a mathematical framework to formulate and control privacy loss in FL [23, 25]. Despite no raw data exchange, the communicated models in FL may still disclose sensitive information about training data. By incorporating the DP constraint [101–103], the FL optimization can be regularized to converge to a model with limited information disclosure.

Formally, let $\mathcal{M}$ be a differentially private FL (DP-FL) algorithm that takes the distributed dataset $\mathcal{D}=\{\mathcal{D}_1,\ldots,\mathcal{D}_N\}$ as input and outputs the well-trained model x. Then $\mathcal{M}$ satisfies (ε,δ)-differential privacy if it holds that

$$Pr[\mathcal{M}(\mathcal{D})=x]\leq e^{\varepsilon}Pr[\mathcal{M}(\mathcal{D}')=x]+\delta, \tag{4.1}$$

where $\mathcal{D}'$ is a virtual dataset which differs in $\mathcal{D}$ with only one record. According to the different granularity of $\mathcal{D}'$, the privacy guarantee can be classified into *sample-level* (i.e., $\mathcal{D}'$ differs from $\mathcal{D}$ in a single sample [104]) and *client-level* (i.e., $\mathcal{D}'$ differs from $\mathcal{D}$ in a client's dataset [104]).

Besides the granularity of privacy protection, it is also necessary to consider different adversary/threat models. According to whether the server is trusted or not, there are two main DP models, namely centralized DP (CDP) and local DP (LDP) [105–107]. In CDP, the server is assumed trustworthy, while the threat comes from external malicious analysts or clients. These adversaries may have access to global models from the server to infer the sample or client-specific information. To address this, the server introduces noise into the global model to safeguard privacy in FL with CDP. In contrast, in LDP, the server is assumed honest-but-curious, meaning it follows the protocol but may attempt to infer the information of training samples via the received intermediate results from clients. In this case, FL clients have to locally perturb their gradients or models to defend against this threat in FL with LDP [107–110]. Note that, with the same privacy parameter ϵ, LDP provides stronger privacy protection than CDP, but causes larger utility loss. To improve the privacy-utility tradeoff, distributed DP (DDP) [111] integrated with secure shuffling [25] or secure aggregation [112, 113] has also been proposed, in which the local model updates are perturbed with a smaller noise and then shuffled or securely aggregated to achieve a sufficient CDP noise.

We use the sample-level DP as the default setting to illustrate the general workflow of FL with DP. As shown in Fig. 4, each client first conducts local training and then clips the local gradients $\nabla f_i(x^t)$ with a positive constant c:

$$\overline{\nabla f_i(x^t)}=\nabla f_i(x^t)\cdot\min\left\{1,\frac{c}{\|\nabla f_i(x^t)\|_2}\right\}. \tag{4.2}$$

Gradients clipping [21] is a common method to limit the gradients' *sensitivity*, i.e., the change of the gradient norm due to adding or deleting an individual sample. Afterward, a certain level of DP can be achieved by injecting carefully calibrated noises (e.g.,

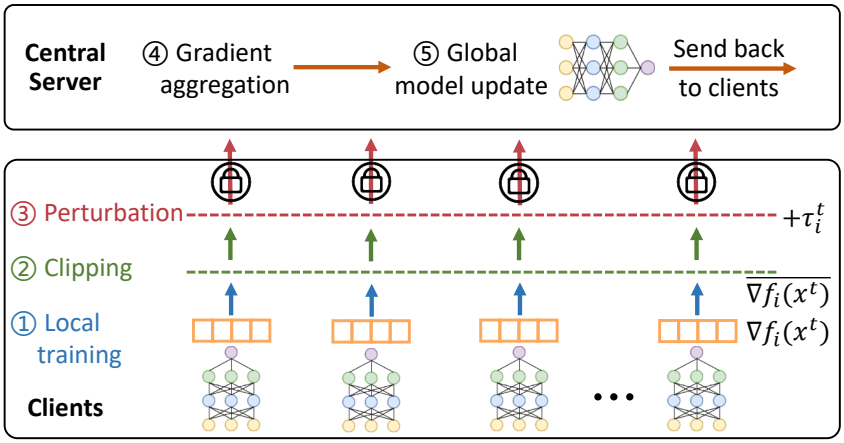


Figure 4: Typical workflow of FL with DP.

Gaussian [21] or Laplacian [22] noises) into the clipped gradients. Taking the most commonly used Gaussian mechanism as an example, the scale of Gaussian noise is specified as $\sigma^2 = \frac{2c^2\log(1.25/\delta)}{\epsilon^2}$ to achieve $(\epsilon,\delta)-$DP [21]. Finally, the perturbed gradient $\overline{\nabla f_i(x^t)}+\tau_i^t$ is sent to the central server for updating the global model [104].

4.1 Negative effect of DP on FL optimization

As discussed, realizing DP in FL involves the operations of gradient clipping and noise addition to the vanilla FL algorithms. While guaranteeing DP, these operations inevitably lead to a reduced convergence rate and increased error bound, reflecting the privacy-utility trade-off [24, 25]. Specifically, their major impacts on the model convergence can be summarized as follows:

- **Decreased convergence rate.** Both gradient clipping and noise addition introduce addition error, thus impacting the convergence of FL algorithms [23–25]. As analyzed in [25], the convergence rate of DP-enhanced FL can be represented as

$$\mathcal{O}\left(\frac{\log(T)\max\{d^{\frac{1}{2}-\frac{1}{p}},1\}}{\sqrt{T}}\sqrt{\frac{cd}{q}\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)}\right), \tag{4.3}$$

  where p represents the norm exponent adopted to clip the gradient, q is the data sampling ratio in a mini-batch, and d is the dimension of model parameters. Specifically, it is This theoretical result indicates that a larger sampling ratio q and privacy budget ϵ can speed up the convergence, while the larger gradient dimension d and clipping bound c can slow down the convergence.

- **Increased error bound.** Besides the convergence rate, DP also harms the final model utility. By achieving DP in FedAvg, an error bound of the trained model [23]

can be derived as follows with a small clipping bound c.

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\alpha^t\left\|\nabla f\left(x^t\right)\right\|^2\right]\leq\mathcal{O}\left(\frac{1}{\sqrt{KET}}\right)+\mathcal{O}(c^2E)+\mathcal{O}(\frac{d\sigma^2}{EK}), \tag{4.4}$$

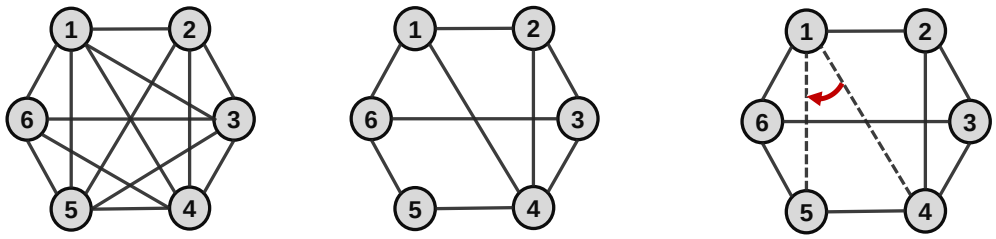
where α is a constant, K is the number of participant clients, and σ^2 and d represent the noise scale and the dimension respectively. The term $\mathcal{O}(c^2E)$ in the right side of Eq. (4.4) captures the impact of clipping, wherein a larger clipping bound c indicates a higher error bound. Similarly, the last term $\mathcal{O}(\frac{d\sigma^2}{EP})$ demonstrates that the error bound increases linearly with the dimension d and noise scale σ^2.

4.2 Optimization methods of FL with DP

Section 4.1 reveals that an additional DP guarantee compromises the model utility in terms of both error bound and convergence rate. Recent studies [114, 115] propose several strategies to improve the model utility in FL without sacrificing the DP guarantee, including the clipping bound calibration to reduce sensitivity [116], sampling for amplifying the privacy preservation [25], and gradient sparsification to enhance the siginal-noise-ratio [115].

- **Sensitivity reduction by fine-grained clipping.** As analyzed in Section 4.1, a small clipping bound can effectively improve the model utility through limiting the added noises. However, an undersized clipping bound c can lead to severe biases in the aggregated results [116, 117], also diminishing the model utility. Therefore, the clipping bound should be calibrated carefully. To this end, recent study [116] empirically observes the reduction of gradient norms with increasing T and accordingly proposes an adaptive norm-aware clipping algorithm. The fine-grained clipping algorithm effectively reduces the aggregation bias and DP noises, achieving a convergence rate of $\mathcal{O}(\frac{1}{T})$ under the assumption of smooth objective function [116].

- **Privacy amplification by sampling.** As a common method, sampling [25, 118], e.g., the client sampling and data sampling in FedAvg, can amplify the privacy protection capability of a randomized FL algorithm by introducing additional randomness. Given ϵ_0 as the privacy budget to protect gradients in each round, the total T rounds of training can be proved to satisfy $(\mathcal{O}(\epsilon_0\sqrt{qT\log(qT)}),\delta)$-DP [25], where q is the sampling ratio. This theoretical result indicates that by setting a smaller sampling ratio, the integration of the sampling method can significantly reduce the DP noises required to perturb the gradients [114], thereby improving the model utility. However, this is conflicted with the analysis that a larger sampling ratio speeds up the convergence, as discussed in Section 4.1. Therefore, we conclude that there exists a trade-off between the model convergence and the sampling ratios [25]. Thus, through appropriate calibration of the sampling ratios, substantial enhancements in model utility can be achieved.



(a) Fully connected topology. (b) Partially connected topology. (c) Time-varying connected topology.

Figure 5: Illustration of network topology for fully connected topology (a), partially connected topology (b), time-varying connected topology (c).

- **Dimension reduction by sparsifying the gradients.** Eq. (4.3) demonstrates that a high dimension d of gradients can degrade the model convergence [119]. A direct solution involves randomly perturbing k out of d dimensions in the model gradients with Gaussian noise while setting the remaining dimensions to zero [120], based on which, the dimension-related terms in the error bound (Eq (4.4)) can then be scaled by $\frac{k}{d}$, thus decreasing the error bound. However, such a random sparsification algorithm introduces significant biases into gradients, thus also degrading the model utility. To this end, recent work [115] first discovers the effect of zeroing out different dimensions of gradients on the model utility, and accordingly proposes to identify and zero out $d-k$ dimensions with the least impact.

4.3 Discussion

The core challenge of FL optimization with DP is to strike a satisfactory balance between privacy protection and model performance (e.g., model errors and convergence rates). Under a certain level of privacy protection, reducing the intensity of DP noises is a core idea to achieve better model performance. Generally, the noise intensity is proportional to both the model sensitivity and the size of communicated data in the whole training process [121, 122]. Given this fact, optimization methods described in Section 4.2 utilize gradient clipping to reduce model sensitivity, and employ gradient sparsification or sampling to reduce the size of communicated data, thereby decreasing the added noise. Additionally, sampling, which introduces extra randomness, further enhances the level of privacy protection. Besides, these methods are complementary and can be combined to achieve a better trade-off between privacy protection and model performance.

5 Federated Learning with Decentralized Topologies

Conventional FL, which is also named server-client FL [123], enables a central server to coordinate the learning task by communicating with multiple distributed clients. However, server-client FL encounters several problems, such as single-point failure [124], communication bottleneck [125], and potential malicious server [126], etc. As an effective alternative, Decentralized FL (DFL) [127–129] can mitigate these problems.

In DFL, each client can only communicate with its neighboring nodes. Formally, the network topology of DFL can be modeled as an adjacent matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, where $\mathbf{A}_{i,j} = 1/0$ indicates the presence/absence of the communication link between nodes i and j. As shown in Figure 5, the network topology can fall into three categories according to different links among nodes, including fully connected topology [130], partially connected topology [129] and time-varying connected topology [17, 131].

Under fully connected topology, each pair of nodes maintains direct link [130], and the adjacent matrix $\mathbf{A} = \mathbf{E}$, where $\mathbf{E}$ is the all-ones matrix. Under partially connected topology, each node only maintains direct links to a subset of nodes [129] and the adjacent matrix is a time-invariant symmetric $0/1$ matrix. Classical partially connected topologies include ring-structured [132] and clique topologies [133, 134], etc. Under time-varying connected topology, links among nodes are randomly or selectively determined based on specific factors [131] such as resource availability. Here, the adjacent matrix is a time-varying symmetric $0/1$ matrix.

In DFL with the above topologies, due to the lack of a central server and limited communication range, it becomes challenging to obtain global gradient information for each client in each round of aggregation. Several DFL methods have been proposed to mitigate this issue for different underlying topologies. Some typical ones are summarized as follows.

- **Decentralized FedAvg** [126] is an adaptation of FedAvg in server-client FL. It enables each node, e.g., the i-th node, to aggregate neighboring gradients to update its own model x_i. Through multiple local aggregations, x_i can gradually fuse the information from the nodes that is not in $\mathcal{N}_i$. Such information propagation in the decentralized network can facilitate the model consistency and convergence.

- **Cyclic learning**, originally proposed in decentralized and consensus optimizations [135, 136], has been also adapted into DFL [137, 138]. It allows different clients to train a single model sequentially and cyclically in the FL system with ring-structured topology. Specifically, in each round, each client trains the model received from the previous client and then passes the trained model to the next client. Such a process proceeds for multiple rounds. However, cyclic learning may suffer from poor model utility due to *catastrophic forgetting* [139–141].

- **Swarm learning** [131, 142] dynamically elects the leader among nodes to aggregate local models based on the consensus mechanism of blockchain. It combines

the strengths of server-client FL and DFL, eliminating the need for a trusted third party while guaranteeing consistency among local models. It can achieve the same model utility as the server-client FL with additional computation and communication costs in the elected leader in each round.

5.1 Impact of different topologies on FL

Decentralized FL is essentially an extension of the well-established field of decentralized optimization [143, 144]. In traditional decentralized optimization, the influence of topologies has been revealed explicitly [145]. And it is apparent that the connectivity of the graph topology impacts the model performance in DFL [127, 146] as well. However, it encounters new challenges in analyzing the impact of decentralized topologies on DFL due to the distinct characteristics of FL scenarios, such as constrained communication, heterogeneous data, privacy concerns, etc. Specifically, the lower node connectivity coupled with non-i.i.d. dataset, even DP noise, will significantly hinder the information propagation, and degrade the model convergence and generalization performance [26, 27, 146].

- **Decreased convergence rate.** It has been proved that static topology can achieve a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}}+\frac{1}{(1-\lambda)^2T^{3/2}})$ [127], where $\lambda\in(0,1)$ is the second largest absolute value of the eigenvalues of the mixing matrix (i.e., maximum-degree matrix and metropolis-hastings matrix [147]) associated with network topology. A smaller λ corresponds the higher graph connectivity of the topology. The convergence rate reveals that a smaller λ can result in a faster convergence [26, 27, 127, 148], meaning that a higher graph connectivity can facilitate the model convergence. A randomly connected topology probably corresponds to a larger λ, thereby decreasing the model convergence rate. Furthermore, as a common issue in FL, the data heterogeneity is also proven to be an important factor to degrade the model convergence in DFL [127, 145, 149].
- **Decreased generalization ability.** It has been proved that the *generalization error* of DFL, i.e., the difference between the true risk of model on the real data distribution and the empirical risk on the training data distribution, is a monotone-increasing function with respect to λ. This implies that the generalization ability of DFL can be degraded by the low connectivity of the underlying topology [26, 27].

5.2 Optimization methods for DFL

Extensive optimization methods for improving the model convergence and generalization ability in DFL have been proposed [127, 146]. We review these methods from the aspects of topology-aware optimization [26], and accelerated optimization [17, 150].

- **Topology-aware optimization** methods [26] attempt to design graph topology with higher connectivity to facilitate model convergence and enhance model generalization ability. However, a highly connected graph inevitably incurs a large overhead

on communications [151, 152]. To speed up model convergence with limited communication resources, existing work proposes a sparse network topology design method based on d-regular expander graphs to optimize the trade-off between connectivity and communication, where d is a pre-specified threshold of graph degree. This method gradually densifies the network topology based on a set of virtual coordinates and recursive queries until the degree of each node reaches d.

Considering the topology of DFL, multiple gossip steps (MGS), which means more frequent communication, can result in improved consensus among the participating clients. In view of this, a balance between the communication cost and generalization ability can be ensured [153].

- **Accelerated optimization** methods [17, 146, 150] adapt advanced optimization techniques to facilitate model convergence in DFL settings. By approximating the global unbiased gradient using neighboring local gradients, the variance reduction method corrects the local gradient for each client [146, 154, 155]. This reduces model inconsistency caused by data heterogeneity and achieves a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{KT}})$. Combining the Nesterov gradient descent method with gradient compression, a contractive compression operator is derived for the time-varying decentralized topology [150]. Theoretical analysis demonstrates the accelerated model convergence. Besides, by modeling the DFL task as a constrained optimization problem:

$$\begin{aligned} &\min_{x_i,\, i\in[N]} \ \frac{1}{N}\sum_{i=1}^{N} f_i(x_i), \\ &s.t. \ \ x_i = x_j, \ \ \forall (i,j)\in\mathcal{E}, \end{aligned} \tag{5.1}$$

  where $\mathcal{E}$ represents the set of node pairs that maintain links, ADMM can be applied to decouple and solve the DFL optimization problem, effectively enhancing model consistency among nodes [17, 156]. Theoretical results demonstrate that the iteration sequence generated by the ADMM algorithm converges to ϵ-suboptimality with a manageable iteration complexity $\mathcal{O}(\frac{1}{\epsilon})$ under time-varying (un)directed network topologies.

  By utilizing sharpness aware minimization (SAM), i.e., introducing a small perturbation to the models, both generalization and robustness are enhanced [153]. Meanwhile, the improved convergence rate $\mathcal{O}(\frac{1}{\sqrt{KT}}+\frac{1}{T}+\frac{1}{K^{1/2}T^{3/2}(1-\lambda)^2})$ can be achieved in the non-convex settings.

5.3 Discussion

Table 4 summarizes the advantages and disadvantages of the two optimization methods. Topology-aware optimization methods improve the model convergence and generalization ability by optimizing the underlying topology directly. However, it is only applicable

Table 4: Summary of optimization methods in DFL.

Methods	**Advantages**	**Disadvantages**	**Network topologies**
Topology-aware optimization	Enhanced model convergence and generalization	Constraints of connectivity among clients	Partially connected topology
Accelerated optimization	Enhanced model convergence	High computation cost	All topologies

to settings where the topology is allowed to change. In practice, the generated topology potentially imposes higher communication costs on neighboring nodes located at substantial geographic distances. Accelerated optimization methods can enhance the model convergence by manipulating the gradients in each round of training. It is applicable to all types of topologies but suffers from relatively higher computation costs to correct the gradients in each round.

6 Online Federated Learning

The above discussions of FL optimization assume that each client possesses a fixed and static dataset, which is also referred to as batch-based FL. However, in numerous real-world scenarios, the training data of clients are often generated in a streaming mode [157]. Consequently, online FL (OFL) [28–31] has been introduced by combining online learning (OL) [158, 159] and FL. OFL performs online optimization in FL over the time-evolving data, and makes a prompt label prediction or decision upon receiving incoming data.

OFL aims to learn a sequence of global models from distributed streaming data at local devices. At timestamp t, each client indexed by $i\in[N]$ receives a new data (u_i^t,v_i^t), where u_i^t is the feature and v_i^t is the label, and the latest global model x^t from the server. The global model is used to predict the label of newly incoming data. Thus, the i-th client has a local loss $\mathcal{L}(f(u_i^t;x^t),v_i^t)$, where $\mathcal{L}(\cdot,\cdot)$ is a loss function that measures the error between true and predicted labels. Leveraging the local loss, the i-th client optimizes its local model x_i^{t+1} and sends it to the server. After receiving all local models, the server updates the global model x^{t+1} by averaging the local models $\{x_i^{t+1}\}_{i=1}^N$. OFL aims to seek a sequence of global models $x^1,x^2,\ldots,x^t$ that minimizes the cumulative regret (i.e., the difference between cumulative loss of the algorithm and that of the static optimal function) over T timestamps:

$$\text{Regret}_T=\sum_{t=1}^{T}\sum_{i=1}^{N}\mathcal{L}(f(u_i^t;x_i^t),v_i^t)-\min_{x\in\mathbb{R}^d}\sum_{t=1}^{T}\sum_{i=1}^{N}\mathcal{L}(f(u_i^t;x),v_i^t). \tag{6.1}$$

6.1 Effect of online paradigm on federated optimization

The online paradigm introduces novel optimization problems and requires new optimization methods for handling streaming data. Here, we discuss the effects of the online paradigm on federated optimization from the following two aspects.

- **Exacerbated issue of obsolete solutions.** In online settings, the model is required to be updated exclusively with newly acquired data and continuously give solutions for dynamic environments [28, 31]. However, general FL methods may be unable to update the model continuously according to new data. This produces obsolete solutions, as the data distribution may have changed, leading to poor model performance (e.g., regret). Furthermore, the constraints of computing and communication in FL aggravate the difficulty of continuously updating the model, which causes a more serious issue of obsolete solutions.

- **Severe catastrophic forgetting.** In the scenario of continuously learning a sequence of tasks based on the online learning paradigm, a model may exhibit degraded performance on old tasks if it only learns new tasks when acquiring new data. This is known as the *catastrophic forgetting* issue [139–141]. More seriously, different FL clients may have different task sequences, which may lead to the mutual interference of task knowledge between different clients [140, 141, 160]. This further aggravates the catastrophic forgetting of clients.

6.2 Optimization methods for OFL

To alleviate the negative effects caused by the online paradigm, many optimization methods for OFL have been proposed [28–31]. We review these methods from the aspects of addressing obsolete solutions and mitigating catastrophic forgetting issues.

- **Kernel-based methods for addressing obsolete solutions**. Current OFL methods [28, 30, 31] mainly rely on online gradient descent method (OGD) [158], an adaptation of SGD for online optimization, to update local models. For better solving non-linear optimization problems (e.g., deep learning tasks) and addressing the issue of obsolete solutions in OFL, some works [28, 30, 161] integrate OGD, FedAvg and multi-kernel learning (MKL). MKL is an advanced method in OL and has exhibited superior performance [162, 163]. The kernel that maps the original input space to a higher-dimensional feature space can significantly improve the generalization capabilities of the model, thereby addressing the issue of obsolete solutions. It is analyzed that the existing OFL methods with multiple kernels can achieve an optimal sub-linear regret $\mathcal{O}(\sqrt{T})$ by setting the learning rate as $\mathcal{O}(\frac{1}{\sqrt{T}})$.

- **Mitigating catastrophic forgetting at the data and model levels.** Existing methods mitigate the catastrophic forgetting issue from both data and model perspectives.

Data-based methods aim to leverage historic training samples or generated samples with the similar distribution for training models with new classes. One way is to simply store and replay old samples at the client [164]. The other is to use generative models to simulate training data with the similar distribution to the historic data [165, 166]. Model-based methods try to balance the model stability on old tasks and generalization on new tasks [139]. For exmaple, weighted averaging of new and old models [164] and decreasing learning rates [167] are beneficial in avoiding catastrophic forgetting for local model updates. The specific mitigating methods can be further divided into the regularization-based and knowledge distillation-based ones. The former usually adds penalty terms to the loss function [164] while the latter applies knowledge distillation to old training samples [165, 166, 168]. Regarding the knowledge distillation-based methods, prototype-based learning approaches [141, 160, 169] are often utilized to collect prototype data of each class and monitor global model performance, which helps to choose the best old global model for knowledge distillation to mitigate the impact of inter-client interference. Also, a recent method of parameter decomposition [164] separates the network parameters into the global and task-specific parameters, enabling clients to selectively learn from each other.

6.3 Discussion

Currently, research on OFL is still in its early stages. The basic framework integrates FedAvg and OGD. Existing works in OFL address obsolete solutions using OL methods like kernel-based approaches [28, 30, 161]. They also tackle the problem of catastrophic forgetting by adopting centralized continual learning methods such as regularization-based approaches [164–166, 168]. However, OFL research has yet to explore the important aspects of heterogeneity and privacy issues in FL from both challenges and optimization methods perspectives.

7 Other important works

In addition to FL optimization studies discussed in the above sections, there are other important research topics, such as sparsification methods [170, 171] and gradient aggregation rules [172, 173]. Sparsification methods aim to optimize the communication complexity of FL systems while maintaining the model performance. Gradient aggregation rules aim to evaluate and aggregate gradients from different local clients to generate a high-quality (e.g., Byzantine robust) global model update.

7.1 Sparsification

Reducing communication complexity is another core challenge in FL optimization [170, 174], especially in the context of massive participants and complex models. Sparsifica-

tion [174–176], as a class of communication-efficient methods, can cut down the communication costs by reducing the number of exchanged parameters, which can also improve the generalization ability of the model [174]. However, these methods have a significant impact on the convergence rate and errors of the model. Thus, the essential problem is to optimize the model convergence while satisfying the communication constraint via calibrating the sparsification parameter. Next, we will review the advanced studies from three perspectives, including client sparsification, temporal sparsification and gradient sparsification.

7.1.1 Client sparsification

Exchanging model updates with abundant participating clients contributes to the communication bottleneck during an FL training round. Client sparsification (also called client sampling) [39, 177, 178], i.e., a random selection of a subset of clients, is a viable solution, but that randomness may result in a lot of missed potential. In most FL implementations, the clients vary in design and capability, a diversity that extends to the quality of communication mediums. Choosing the clients that meet the most favorable communication conditions in each round should help achieve a higher convergence rate [170]. It is analyzed in [171] that a larger selection skew results in faster convergence at the rate $\mathcal{O}(\frac{1}{T\rho})$, where ρ represents the selection skew towards clients with higher local losses, which reveals that biasing client selection towards clients with higher local loss achieves faster convergence.

7.1.2 Temporal sparsification

To leverage all the available data samples on the data owners, standard FL methods generally let the clients synchronize their models through the server in each training iteration. However, this implies many rounds of communication between the clients and the server which results in communication contention over the network. Instead, some works [31, 179] propose that participating clients conduct several local updates and synchronize through the server periodically. Specifically, once clients pull an updated model from the server, they update the model locally by running τ iterations of the SGD method and then send proper information to the server for updating the aggregate model. Under a strongly convex setting, for a total number of iterations $T = K\tau$, where K is a positive integer, the convergence rate is $\mathcal{O}(\frac{\tau}{T}) + \mathcal{O}(\frac{\tau^2}{T^2}) + \mathcal{O}(\frac{(\tau-1)^2}{T}) + \mathcal{O}(\frac{\tau-1}{T^2})$ [179]. In particular, any pick of $\tau = \mathcal{O}(\sqrt{T})$ ensures the convergence of the FL to the global optimal. For smooth non-convex loss functions, the convergence rate is $\mathcal{O}(\frac{1}{\sqrt{T}}) + \mathcal{O}(\frac{\tau-1}{T})$.

7.1.3 Gradient sparsification

Gradient sparsification methods [180, 181] convert a dense gradient into a sparse one by retaining only a subset of significant elements and setting the remaining coordinates to zero. Two commonly used techniques are rand-k sparsification [181] and top-k sparsification [182–184]. In particular, rand-k sparsification randomly selects k elements from

the gradients, whereas top-k sparsification retains the k elements with the highest absolute values. It has been proved [185] that rand-k sparsification is an unbiased compression operator which, however, results in larger compression errors and therefore makes it less effective in practice compared to top-k sparsification when high compression is required [186].

Quantization [31, 179, 187] is another classic sparsification method that reduces the model size by lowering the bit width from 32-bit floating-point to a smaller precision. For now, numerous quantization methods (e.g., stochastic quantization [188], rotation-based quantization [189], etc.) have been proposed to compress model gradients of each client into a discrete set. Theoretical analysis shows that the convergence rate of FL with quantization has the order of $\mathcal{O}(\frac{1}{\sqrt{T}})$, which is the same as that of a classical FL framework for a non-convex loss function [179].

Generally, top-k sparsification methods can reduce communication complexity more efficiently compared to quantization. It has been demonstrated that the top-k sparsification with error feedback can accelerate convergence and accuracy with over 99% gradient elements zeroed out [190].

7.2 Aggregation Rules

The most widely used aggregation rule is FedAvg, which takes weighted average over all local gradients according to local data sizes. However, FedAvg implicitly assumes the equal quality of all gradients, limiting its effectiveness in scenarios like asynchronous FL (e.g., some gradients are stale [191]) and Byzantine attack settings (e.g., some gradients are fake [192]). To overcome these limitations, a variety of aggregation rules [193] have been proposed to optimize the model utility (e.g., model convergence and Byzantine robustness). Advanced aggregation methods can be roughly divided into two categories: weighting-based aggregation and statistic-based aggregation.

7.2.1 Weighting-based aggregation

Weighting-based aggregation rules aim to optimize the model performance (e.g., convergence, fairness, etc.) by assigning differentiated weights to local gradients according to specific statistical indicators (e.g., gradient staleness, model loss. etc.). The aggregation rule is as follows:

$$\nabla f(x) = \frac{\sum_{i=1}^{N} a_i \nabla f(x_i)}{\sum_{i=1}^{N} a_i}, \tag{7.1}$$

where $a_i \geq 0$ represents the weight for local gradient $\nabla f(x_i)$.

In asynchronous FL, with the objective of improving model convergence rate and decreasing errors, the weights are mainly determined by staleness [172, 194] or descent direction [191]. Stale gradients typically exhibit biased descent directions and larger norms than normal gradients. Consequently, simply averaging the stale gradients would result in their dominance during the training process, leading to a sub-linear convergence rate

of $\mathcal{O}(\frac{1}{\sqrt{T}})$ under the assumptions of general convexity [195] or bounded gradient [196]. Motivated by the intuition that low-staleness gradients are more reliable and accurate, it is reasonable to have the server take the weighted average over all gradients according to their staleness to estimate an unbiased gradient. Nonetheless, some high-staleness gradients also exhibit consistent descent directions to unbiased gradients, which can be utilized to accelerate model convergence [191, 197]. Consequently, existing research enables the server to first evaluate the consistency of stale gradients with the estimated unbiased gradient, and accordingly assigns differentiated weights to those gradients to improve the convergence rate [191]. Through increasing the contributions of consistent gradients to the aggregation, such weighted aggregation method can significantly decrease the error in each round. It has been demonstrated that weighted asynchronous FL can achieve a convergence rate of $\mathcal{O}(\frac{1}{T})$ even under the assumption of non-convexity [191, 198].

In fair FL, to achieve performance consistency, most fair algorithms primarily enhance optimization objectives by amplifying the dominance of the large losses, which ultimately influences the aggregation weights in the training process [89, 199, 200]. The fundamental idea is to encourage the global model to demonstrate a favorable inclination towards disadvantaged clients (i.e. clients with smaller losses) by augmenting their weights. A straightforward method is to assign larger weights to gradients with larger loss values [199, 200]. Besides, several research introduces fairness constraints (e.g., the limited difference of predicted label probability distribution under different sensitive attributes [200]) to the FL optimization problem [199, 200]. To solve the constrained problem, the server first assigns different fairness budgets to clients and then adjusts the weights of different gradients according to whether the model performance inconsistency surpasses the assigned fairness budget.

7.2.2 Statistic-based aggregation

Statistic-based aggregation rules [173, 192, 193, 201–204] generate some robust statistics over the local gradients to perform aggregation, aiming to optimize the model utility with the presence of some fake gradients. The typical FedAvg, which takes the arithmetic mean of gradients to update the model, may diverge in the optimization process due to its vulnerability to outliers. In contrast, robust averaging methods [173, 201–203] demonstrate improved robustness, i.e., median/trimmed-mean [201] can achieve order-optimal error rates while maintaining a convergence rate of $O(\frac{\sigma}{\sqrt{KT}})$ under the assumption of non-convexity. However, these methods assume that honest gradients are close to each other and overlook data heterogeneity, making them vulnerable in non-i.i.d. FL settings.

To address the impact of non-i.i.d. data on robustness, recent studies [193, 204] leverages a bucketing step that groups heterogeneous gradients and computes the average within each bucket. This pre-processing step generates more homogeneous gradients, enabling robust aggregation and producing more resilient results. The bucketing-based methods have been shown to converge with a rate of $O(\frac{\sigma}{\sqrt{KT}})$, even under the assumptions of data heterogeneity and non-convexity. Furthermore, a large dimension of model

gradients can also amplify the effects of malicious gradients in non-i.i.d. data settings. To this end, GAS [192] splits high-dimensional gradients into multiple low-dimensional subsets to mitigate these impacts. Robust statistic-based rules can then be applied to each subset, followed by concatenation. This splitting method is proven to be effective in reducing the impact of large gradients dimension d.

Existing studies [205, 206] have demonstrated that the variance of stochastic gradients significantly impacts the Byzantine robustness of FL and a higher variance in stochastic gradients leads to weaker robustness. Hence, some research [205–207] combines existing variance reduction algorithms with robust statistics to mitigate the impact of gradient variances on Byzantine robustness. Specifically, Byra-SAGA [205] proposes the integration of the distributed SAGA algorithm with the geometric median statistics achieving enhanced robustness. Theoretical results suggest that Byra-SAGA can achieve a convergence rate of $O(\frac{1}{T})$. Byz-VR-MARINA [206] introduces the combined gradient compression algorithm VR-MARINA, variance reduction algorithm SARAH/PAGE, and robust aggregation rules, such as the geometric median statistics-based rule, to simultaneously achieve communication compression and robustness enhancement. Theoretical analysis indicates that under general non-convex assumptions, Byz-VR-MARINA can achieve a convergence rate of $O(\frac{1}{T})$.

Additionally, recent research [98, 208–212] has explored the extension of statistic-based aggregation rules to decentralized FL. For example, BRIDGE [209] extends the coordinate-wise trimmed-mean to decentralized FL. Theoretical findings indicate that, in a statistical sense, BRIDGE can converge at a convergence rate of $O(\frac{1}{T})$ to the optimal solution and a first-order stationary point in convex and non-convex settings respectively. IOS [213] provides general guidelines to design Byzantine robust statistics for aggregation in decentralized FL, and proposes an effective method that iteratively discards models that are farthest away from the weighted average of models from neighboring nodes.

8 Possible Future Directions

8.1 Discussions for FL Development

Along with extensive academic research, FL has also shown great potential and even practical applications in several industrial scenarios. For instance, Google has deployed FL on billions of Android systems to enable precise next-word prediction for mobile keyboards [4]. NVIDIA has applied FL in medical image analysis across multiple institutions [214]. As artificial intelligence has been becoming pervasive while privacy consciousness is ever increasing, FL is believed to thrive in the following future application scenarios.

- **FL in autonomous cars and robots.** Both self-driving cars and robots are expected to bloom soon, which will accumulate huge amounts of data in diverse environments. On the other hand, both applications are highly driven by various ML tech-

nologies, which require repeated training over massive data. At the same time, these applications may experience varying communication channels in the real world. Clearly, FL is very promising to be applied for accelerating their model learning while limiting data transfer and privacy disclosure [215, 216].

- **Vertical federated Learning.** Besides these applications based on traditional horizontal FL [35] (where the feature space is identical but the sample space is different), there are also explorations for vertical FL [40] (where the sample space is overlapped but the feature space is orthogonal). For example, Webank has applied vertical FL in financial risk controls by sharing knowledge between banks and insurance companies [35, 217]. Also, many internet companies like Bytedance has adopted vertical FL for intelligent recommendation in e-commence [218].

- **FL for large models (LMs).** Large Models (LMs) have recently demonstrated astonishing AI abilities, gaining massive attention. To exploit the full potential of LMs, it often has to fine-tune them to domain-specific tasks or adapt them with domain-specific knowledge [219]. However, the fine-tuning of LMs not only requires rather powerful computing resources but also relies on a large amount of high-quality domain-specific data [220], which however may be scattered among multiple sites and cannot be centralized. In this case, FL has been becoming a promising technology for achieving privacy-preserving LMs fine-tuning, thus truely grounding the large foundation models [221].

8.2 Future Directions for FL Optimization

Despite the numerous advanced studies and promising applications, there still remains considerable space for the foundational FL optimization in terms of the optimization methods, and the practical system and privacy constraints. In the following, we discuss the possible future directions in FL optimization from three perspectives: optimization, system, and privacy.

- **New theories and methods for black-box optimization.** As we try to optimize the increasingly complicated AI systems, many practical learning tasks are essentially black-box optimization [75, 78, 79], which can hardly give the analytic expression of the loss function or the gradient information. In such cases, besides zeroth-order optimization [78, 79] without the gradient information, we sometimes need more sophisticated optimization theories and methods. For example, hyper-parameter optimization [222–224] and neural architecture search [225, 226] are very promising for optimizing the complex deep learning models. These problems are often modeled as Bi-level optimization where an optimization problem contains another optimization problem as a constraint [90, 227–231] in essence. However, how to adapt these advanced optimizations to the FL setting remains largely unexplored, thus being a promising research direction.

- **Optimization under practical FL system constraints.** FL optimization differs from conventional distributed optimization in many practical system constraints, including the underlying topology [127, 130, 131], parallel mode [38, 232], resource limit [11, 32], and data distribution [5, 18] etc. Existing studies have extensively explored the issue of non-i.i.d. data distribution, but still lack the deep consideration of many system constraints like topology and resource, especially their combinations. As discussed before, the underlying topologies can impact both the convergence rate and stability of FL optimization. In fact, the possible imbalance of resources across computing nodes, can impact not only the choice of the FL parallel mode but also the convergence performance. Therefore, future work may consider how to achieve faster and more accurate FL optimization by simultaneously optimizing the topology and resource allocation among FL nodes.

- **FL optimization aware of privacy-preserving techniques.** FL optimization is proposed as a privacy-aware optimization method by design. Therefore, beyond DP, FL is often incorporated with many other privacy-preserving technologies like multi-party computation [233, 234] and homomorphic encryption [235, 236]. However, similar to the degraded convergence incurred by DP constraints, the additional constraints in these privacy-preserving methods bring new challenges for FL optimization. For example, encryption-based methods often require quantization of exchanged gradient information to reduce the computing complexity [237–239]. This would further result in a degraded performance of FL optimization. Also, the cryptographic primitives significantly burden the computation and communication of FL optimization, which again complicates the trilemma among privacy, utility and efficiency. Therefore, an important question is how to design FL algorithms to optimize the tradeoff among the accuracy, privacy and complexity.

9 Conclusion

As an important interdisciplinary research area in both applied mathematics and information sciences, FL still lacks a summarization of advanced studies in terms of mathematical optimization. To this end, we presented the first systematic survey on the assumptions, formulations, methods, and theoretical results in FL optimization, mainly focusing on the optimization challenges induced by non-i.i.d. data, rigorous privacy guarantee, decentralized topology, and online settings. Besides, we also reviewed some other important works on sparsification methods and aggregation rules, which can also improve FL optimization. Finally, we envisioned the applications of FL in the AI era and discussed several broader future directions from the perspectives of optimization, system, and privacy respectively.

References

[1] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[2] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68, 2018.

[3] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and Hamid R Tizhoosh. Decentralized federated learning through proxy model sharing. *Nature communications*, 14 (1):2899, 2023.

[4] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[5] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[6] Jean Ogier du Terrail, Armand Leopold, Clément Joly, Constance Béguier, Mathieu Andreux, Charles Maussion, Benoît Schmauch, Eric W Tramel, Etienne Bendjebbar, Mikhail Zaslavskiy, et al. Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer. *Nature medicine*, 29(1):135–146, 2023.

[7] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. Ffd: A federated learning based method for credit card fraud detection. In *Proceedings of 8th International Congress on BIGDATA 2019, held as Part of the Services Conference Federation (SCF)*, pages 18–32. Springer, 2019.

[8] Rui Zhang, Hongwei Li, Luoding Tian, Meng Hao, and Yuan Zhang. Vertical federated learning across heterogeneous regions for industry 4.0. *IEEE Transactions on Industrial Informatics*, 2024.

[9] Xinwei Zhang, Mingyi Hong, and Nicola Elia. Understanding a class of decentralized and federated optimization algorithms: A multirate feedback control perspective. *SIAM Journal on Optimization*, 33(2):652–683, 2023.

[10] Behrouz Touri and Bahman Gharesifard. A unified framework for continuous-time unconstrained distributed optimization. *SIAM Journal on Control and Optimization*, 61(4):2004–2020, 2023.

[11] Haoyu Zhao, Konstantin Burlachenko, Zhize Li, and Peter Richtárik. Faster rates for compressed federated learning with client-variance reduction. *SIAM Journal on Mathematics of Data Science*, 6(1):154–175, 2024.

[12] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

[13] Guanghui Lan and Yi Zhou. Random gradient extrapolation for distributed and stochastic optimization. *SIAM Journal on Optimization*, 28(4):2753–2782, 2018.

[14] Konstantin Mishchenko, Franck Iutzeler, and Jérôme Malick. A distributed flexible delay-tolerant proximal gradient algorithm. *SIAM Journal on Optimization*, 30(1):933–959, 2020.

[15] Olivier Beaude, Pascal Benchimol, Stéphane Gaubert, Paulin Jacquot, and Nadia Oudjane. A privacy-preserving method to optimize distributed resource allocation. *SIAM Journal on Optimization*, 30(3):2303–2336, 2020.

[16] Tobias Harks and Julian Schwarz. A unified framework for pricing in nonconvex resource

allocation games. *SIAM Journal on Optimization*, 33(2):1223–1249, 2023.

[17] Necdet Serhat Aybat and Erfan Yazdandoost Hamedani. A distributed admm-like method for resource sharing over time-varying networks. *SIAM Journal on Optimization*, 29(4):3036–3068, 2019.

[18] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning (ICML)*, pages 5132–5143. PMLR, 2020.

[20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006.

[21] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (CCS)*, pages 308–318, 2016.

[22] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 304–317. IEEE, 2020.

[23] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. In *International Conference on Machine Learning (ICML)*, 2022.

[24] Rui Hu, Yuanxiong Guo, and Yanmin Gong. Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy. *IEEE Transactions on Mobile Computing*, 2023.

[25] Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2521–2529. PMLR, 2021.

[26] Yifan Hua, Kevin Miller, Andrea L. Bertozzi, Chen Qian, and Bao Wang. Efficient and reliable overlay networks for decentralized federated learning. *SIAM Journal on Applied Mathematics*, 82(4):1558–1586, 2022.

[27] Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[28] Songnam Hong and Jeongmin Chae. Communication-efficient randomized algorithm for multi-kernel online federated learning. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):9872–9886, 2021.

[29] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 15–24. IEEE, 2020.

[30] Pouya M Ghari and Yanning Shen. Personalized online federated learning with multiple kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 33316–33329, 2022.

[31] Dohyeok Kwon, Jonghwan Park, and Songnam Hong. Tighter regret analysis and optimization of online federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[32] Francois Gauthier, Vinay Chakravarthi Gogineni, Stefan Werner, Yih-Fang Huang, and Anthony Kuh. Resource-aware asynchronous online federated learning for nonlinear regression. In *ICC 2022-IEEE International Conference on Communications (ICC)*, pages 2828–2833.

IEEE, 2022.

[33] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

[34] Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. Federated learning with non-iid data: A survey. *IEEE Internet of Things Journal*, 2024.

[35] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10 (2):1–19, 2019.

[36] Xuebin Ren, Shusen Yang, Cong Zhao, Julie McCann, and Zongben Xu. Belt and braces: When federated learning meets differential privacy. *Communications of the ACM (CACM)*, 67(12):66–77, 2024.

[37] K Naveen Kumar, C Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[38] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37, 2022.

[39] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017.

[40] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[41] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. Cafe: Catastrophic data leakage in vertical federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 994–1006, 2021.

[42] Ganyu Wang, Bin Gu, Qingsong Zhang, Xiang Li, Boyu Wang, and Charles X Ling. A unified solution for privacy and communication efficiency in vertical federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

[43] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[44] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2020.

[45] Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019.

[46] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2(4):7, 2018.

[47] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning (ICML)*, pages 5311–5319. PMLR, 2021.

[48] Zhengjie Yang, Wei Bao, Dong Yuan, Nguyen H Tran, and Albert Y Zomaya. Federated learning with nesterov accelerated gradient. *IEEE Transactions on Parallel and Distributed*

Systems, 33(12):4863–4873, 2022.

[49] Li Ju, Tianru Zhang, Salman Toor, and Andreas Hellander. Accelerating fair federated learning: Adaptive federated adam. *arXiv preprint arXiv:2301.09357*, 2023.

[50] Shenglong Zhou and Geoffrey Ye Li. Federated learning via inexact admm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[51] Beomyeol Jeon, SM Ferdous, Muntasir Raihan Rahman, and Anwar Walid. Privacy-preserving decentralized aggregation for federated learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6. IEEE, 2021.

[52] Shashi Kant, José Mairton B da Silva, Gabor Fodor, Bo Göransson, Mats Bengtsson, and Carlo Fischione. Federated learning using three-operator admm. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):205–221, 2022.

[53] Anis Elgabli, Chaouki Ben Issaid, Amrit Singh Bedi, Ketan Rajawat, Mehdi Bennis, and Vaneet Aggarwal. Fednew: A communication-efficient and privacy-preserving newton-type method for federated learning. In *International Conference on Machine Learning (ICML)*, pages 5861–5877. PMLR, 2022.

[54] Mher Safaryan, Rustem Islamov, Xun Qian, and Peter Richtárik. Fednl: Making newton-type methods applicable to federated learning. *arXiv preprint arXiv:2106.02969*, 2021.

[55] Canh T Dinh, Nguyen H Tran, Tuan Dung Nguyen, Wei Bao, Amir Rezaei Balef, Bing B Zhou, and Albert Y Zomaya. Done: distributed approximate newton-type method for federated edge learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2648–2660, 2022.

[56] C Nagaraju, Mrinmay Sen, and C Krishna Mohan. Fonn: Federated optimization with nys-newton. In *TENCON 2023-2023 IEEE Region 10 Conference (TENCON)*, pages 530–534. IEEE, 2023.

[57] Wenzhi Fang, Ziyi Yu, Yuning Jiang, Yuanming Shi, Colin N Jones, and Yong Zhou. Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70:5058–5073, 2022.

[58] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[59] Yuyang Qiu, Uday Shanbhag, and Farzad Yousefian. Zeroth-order methods for nondifferentiable, nonconvex, and hierarchical federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[60] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[61] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *19th International Conference on Computational StatisticsParis France (COMPSTAT)*, pages 177–186. Springer, 2010.

[62] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations (ICLR)*, 2019.

[63] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine learning and systems (MLSys)*, volume 2, pages 429–450, 2020.

[64] Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 31(8): 1754–1766, 2020.

[65] Zhouyuan Huo, Qian Yang, Bin Gu, Lawrence Carin Huang, et al. Faster on-device training using new federated momentum algorithm. *arXiv preprint arXiv:2002.02090*, 2020.
[66] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.
[67] Jianhui Sun, Xidong Wu, Heng Huang, and Aidong Zhang. On the role of server momentum in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 15164–15172, 2024.
[68] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.
[69] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.
[70] Tomoya Murata and Taiji Suzuki. Bias-variance reduced local sgd for less heterogeneous federated learning. *arXiv preprint arXiv:2102.03198*, 2021.
[71] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10112–10121, 2022.
[72] Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *Journal of Machine Learning Research*, 18(116):1–40, 2017.
[73] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.
[74] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
[75] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.
[76] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.
[77] Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023.
[78] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.
[79] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.
[80] Jun Chen, Hong Chen, Bin Gu, and Hao Deng. Fine-grained theoretical analysis of federated zeroth-order optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.
[81] Xun Qian, Rustem Islamov, Mher Safaryan, and Peter Richtarik. Basis matters: Better communication-efficient second order methods for federated learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 680–720. PMLR, 2022.
[82] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

[83] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4519–4529. PMLR, 2020.

[84] Sijin Chen, Zhize Li, and Yuejie Chi. Escaping saddle points in heterogeneous federated learning via distributed sgd with communication compression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2701–2709. PMLR, 2024.

[85] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 5693–5700, 2019.

[86] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7611–7623, 2020.

[87] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[88] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

[89] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning (ICML)*, pages 6357–6368. PMLR, 2021.

[90] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21394–21405, 2020.

[91] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[92] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International Conference on Machine Learning (ICML)*, pages 32991–33013. PMLR, 2023.

[93] Zhikai Yang, Yaping Liu, Shuo Zhang, and Keshen Zhou. Personalized federated learning with model interpolation among client clusters and its application in smart home. In *World Wide Web (WWW)*, volume 26, pages 2175–2200. Springer, 2023.

[94] Minghui Chen, Meirui Jiang, Qi Dou, Zehua Wang, and Xiaoxiao Li. Fedsoup: Improving generalization and personalization in federated learning via selective model interpolation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 318–328. Springer, 2023.

[95] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *Journal of Machine Learning Research*, 21(219):1–30, 2020.

[96] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[97] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[98] Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. Momentum benefits non-iid federated learning simply and provably. In *International Conference on Learning Representa-*

tions (ICLR), 2024.

[99] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=SkxJ8REYPH`.

[100] Rui Ye, Mingkai Xu, Jianyu Wang, Chenxin Xu, Siheng Chen, and Yanfeng Wang. Feddisco: Federated learning with discrepancy-aware collaboration. In *International Conference on Machine Learning (ICML)*, pages 39879–39902. PMLR, 2023.

[101] Jonathan Ullman. Tight lower bounds for locally differentially private selection. *arXiv preprint arXiv:1802.02638*, 2018.

[102] Nurdan Kuru, S Ilker Birbil, Mert Gurbuzbalaban, and Sinan Yildirim. Differentially private accelerated optimization algorithms. *SIAM Journal on Optimization*, 32(2):795–821, 2022.

[103] Ziqin Chen and Yongqiang Wang. Locally differentially private decentralized stochastic bilevel optimization with guaranteed convergence accuracy. In *International Conference on Machine Learning*, 2024.

[104] Xiaoyong Yuan, Xiyao Ma, Lan Zhang, Yuguang Fang, and Dapeng Wu. Beyond class-level privacy leakage: Breaking record-level privacy in federated learning. *IEEE Internet of Things Journal*, 9(4):2555–2565, 2021.

[105] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.

[106] Mengchu Li, Thomas B Berrett, and Yi Yu. On robustness and local differential privacy. *The Annals of Statistics*, 51(2):717–737, 2023.

[107] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):1–10, 2022.

[108] Baocang Wang, Yange Chen, Hang Jiang, and Zhen Zhao. Ppefl: Privacy-preserving edge federated learning with local differential privacy. *IEEE Internet of Things Journal*, 10(17): 15488–15500, 2023.

[109] Hajira Batool, Adeel Anjum, Abid Khan, Stefano Izzo, Carlo Mazzocca, and Gwanggil Jeon. A secure and privacy preserved infrastructure for vanets based on federated learning with local differential privacy. *Information Sciences*, 652:119717, 2024.

[110] Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated latent dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6283–6290, 2020.

[111] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Proc. EUROCRYPT*, pages 375–403. Springer, 2019.

[112] Wei-Ning Chen, Christopher A Choquette Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. In *International Conference on Machine Learning (ICML)*, pages 3056–3089. PMLR, 2022.

[113] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[114] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS

Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[115] Anda Cheng, Peisong Wang, Xi Sheryl Zhang, and Jian Cheng. Differentially private federated learning with local regularization and sparsification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10122–10131, 2022.

[116] Yanan Li, Shusen Yang, Xuebin Ren, Liang Shi, and Cong Zhao. Multi-stage asynchronous federated learning with adaptive differential privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[117] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13773–13782, 2020.

[118] Yifan Shi, Yingqi Liu, Kang Wei, Li Shen, Xueqian Wang, and Dacheng Tao. Make landscape flatter in differentially private federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24552–24562, 2023.

[119] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In *International Conference on Machine Learning (ICML)*, pages 4587–4604. PMLR, 2022.

[120] Rui Hu, Yanmin Gong, and Yuanxiong Guo. Federated learning with sparsification-amplified privacy and adaptive optimization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[121] Yiwei Li, Shuai Wang, Chong-Yung Chi, and Tony QS Quek. Differentially private federated learning in edge networks: The perspective of noise reduction. *IEEE Network*, 36(5): 167–172, 2022.

[122] Tao Qi, Fangzhao Wu, Chuhan Wu, Liang He, Yongfeng Huang, and Xing Xie. Differentially private knowledge transfer for federated learning. *Nature Communications*, 14(1): 3785, 2023.

[123] Sisi Zhou, Kuanching Li, Yuxiang Chen, Ce Yang, Wei Liang, and Albert Y Zomaya. Trustbcfl: Mitigating data bias in iot through blockchain-enabled federated learning. *IEEE Internet of Things Journal*, 2024.

[124] Yuben Qu, Haipeng Dai, Yan Zhuang, Jiafa Chen, Chao Dong, Fan Wu, and Song Guo. Decentralized federated learning for uav networks: Architecture, challenges, and opportunities. *IEEE Network*, 35(6):156–162, 2021.

[125] Abolfazl Hashemi, Anish Acharya, Rudrajit Das, Haris Vikalo, Sujay Sanghavi, and Inderjit Dhillon. On the benefits of multiple gossip steps in communication-constrained decentralized federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11): 2727–2739, 2021.

[126] Hongchang Gao, My T Thai, and Jie Wu. When decentralized optimization meets federated learning. *IEEE network*, 2023.

[127] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4289–4301, 2023.

[128] Jie Xu, Wei Zhang, and Fei Wang. A(dp)^{2}sgd: Asynchronous decentralized parallel stochastic gradient descent with differential privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8036–8047, 2022.

[129] Pedro Miguel Sánchez Sánchez Sergio López Bernal Gérôme Bovet Manuel Gil Pérez Gregorio Martínez Pérez Enrique Tomás Martínez Beltrán, Mario Quiles Pérez and Alberto Huertas Celdrán. Decentralized federated learning: fundamentals, state of the art,

frameworks, trends, and challenges. *IEEE Communications surveys and tutorials*, 25(4):2983–3013, 2023.

[130] Hao Ye, Le Liang, and Geoffrey Ye Li. Decentralized federated learning with unreliable communications. *IEEE journal of selected topics in signal processing*, 16(3):487–500, 2022.

[131] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.

[132] Zhao Wang, Yifan Hu, Jun Xiao, and Chao Wu. Efficient ring-topology decentralized federated learning with deep generative models for industrial artificial intelligent. *arXiv preprint arXiv:2104.08100*, 2021.

[133] Aurélien Bellet, Anne-Marie Kermarrec, and Erick Lavoie. D-cliques: Compensating for data heterogeneity with topology in decentralized federated learning. In *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, pages 1–11. IEEE, 2022.

[134] Thijs Vogels, Hadrien Hendrikx, and Martin Jaggi. Beyond spectral gap: The role of the topology in decentralized learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 15039–15050, 2022.

[135] Xianghui Mao, Kun Yuan, Yubin Hu, Yuantao Gu, Ali H Sayed, and Wotao Yin. Walkman: A communication-efficient random-walk algorithm for decentralized optimization. *IEEE Transactions on Signal Processing*, 68:2513–2528, 2020.

[136] Xianghui Mao, Yuantao Gu, and Wotao Yin. Walk proximal gradient: An energy-efficient algorithm for consensus optimization. *IEEE Internet of Things Journal*, 6(2):2048–2060, 2018.

[137] Marcos F. Criado, Fernando E. Casado, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro. Non-iid data and continual learning processes in federated learning: A long road ahead. *Information Fusion*, 88:263–280, 2022. ISSN 1566-2535.

[138] Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598, 2020.

[139] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[140] Zhe Wang, Yu Zhang, Xinlei Xu, Zhiling Fu, Hai Yang, and Wenli Du. Federated probability memory recall for federated continual learning. *Information Sciences*, 629:551–565, 2023.

[141] Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[142] Oliver Lester Saldanha, Philip Quirke, Nicholas P West, Jacqueline A James, Maurice B Loughrey, Heike I Grabsch, Manuel Salto-Tellez, Elizabeth Alwers, Didem Cifci, Narmin Ghaffari Laleh, et al. Swarm learning for decentralized artificial intelligence in cancer histopathology. *Nature medicine*, 28(6):1232–1239, 2022.

[143] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[144] Cassio G Lopes and Ali H Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, 2008.

[145] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A

unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning (ICML)*, pages 5381–5393. PMLR, 2020.

[146] Xin Zhang, Minghong Fang, Zhuqing Liu, Haibo Yang, Jia Liu, and Zhengyuan Zhu. Net-fleet: Achieving linear convergence speedup for fully decentralized federated learning with heterogeneous data. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc)*, pages 71–80, 2022.

[147] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 46(4):667–689, 2004.

[148] Jiadong Liang, Yang Peng, and Zhihua Zhang. Gradient tracking for high dimensional federated optimization, 2023. URL `https://arxiv.org/abs/2312.05590`.

[149] Yasaman Esfandiari, Sin Yong Tan, Zhanhong Jiang, Aditya Balu, Ethan Herron, Chinmay Hegde, and Soumik Sarkar. Cross-gradient aggregation for decentralized learning from non-iid data. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3036–3046. PMLR, 18–24 Jul 2021.

[150] Dmitry Kovalev, Egor Shulgin, Peter Richtarik, Alexander V Rogozin, and Alexander Gasnikov. Adom: Accelerated decentralized optimization method for time-varying networks. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 5784–5793. PMLR, 18–24 Jul 2021.

[151] Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 13975–13987, 2021.

[152] Zhuoqing Song, Weijian Li, Kexin Jin, Lei Shi, Ming Yan, Wotao Yin, and Kun Yuan. Communication-efficient topologies for decentralized learning with $o(1)$ consensus rate. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 1073–1085, 2022.

[153] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31269–31291. PMLR, 23–29 Jul 2023.

[154] Jiaojiao Zhang, Huikang Liu, Anthony Man-Cho So, and Qing Ling. Variance-reduced stochastic quasi-newton methods for decentralized learning. *IEEE Transactions on Signal Processing*, 71:311–326, 2023.

[155] Ran Xin, Usman A Khan, and Soummya Kar. Vari ance-reduced decentralized stochastic optimization with accelerated convergence. *IEEE Transactions on Signal Processing*, 68:6255–6271, 2020.

[156] Qunwei Li, Bhavya Kailkhura, Ryan Goldhahn, Priyadip Ray, and Pramod K Varshney. Robust decentralized learning using admm with unreliable agents. *IEEE Transactions on Signal Processing*, 70:2743–2757, 2022.

[157] Arindam Banerjee and Sugato Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, pages 431–436. SIAM, 2007.

[158] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[159] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.
[160] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10164–10173, 2022.
[161] Vinay Chakravarthi Gogineni, Stefan Werner, Yih-Fang Huang, and Anthony Kuh. Communication-efficient online federated learning strategies for kernel regression. *IEEE Internet of Things Journal*, 10(5):4531–4544, 2022.
[162] Ofer Dekel, Shai Shalev-Shwartz, and Yoram Singer. The forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing*, 37(5):1342–1372, 2008.
[163] Steven CH Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. Online multiple kernel classification. *Machine learning*, 90:289–316, 2013.
[164] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning (ICML)*, pages 12073–12086. PMLR, 2021.
[165] Sara Babakniya, Zalan Fabian, Chaoyang He, Mahdi Soltanolkotabi, and Salman Avestimehr. A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.
[166] Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4782–4793, 2023.
[167] Liangqi Yuan, Yunsheng Ma, Lu Su, and Ziran Wang. Peer-to-peer federated continual learning for naturalistic driving action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5249–5258, 2023.
[168] Yuhang Ma, Zhongle Xie, Jue Wang, Ke Chen, and Lidan Shou. Continual federated learning based on knowledge distillation. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2182–2188, 2022.
[169] Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5054–5062, 2023.
[170] Omair Rashed Abdulwareth Almanifi, Chee-Onn Chow, Mau-Luen Tham, Joon Huang Chuah, and Jeevan Kanesan. Communication and computation efficiency in federated learning: A survey. *Internet of Things*, 22:100742, 2023.
[171] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.
[172] Yinbin Miao, Ziteng Liu, Xinghua Li, Meng Li, Hongwei Li, Kim-Kwang Raymond Choo, and Robert H Deng. Robust asynchronous federated learning with time-weighted and stale model aggregation. *IEEE Transactions on Dependable and Secure Computing*, 2023.
[173] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
[174] Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799, 2021.
[175] Zihao Zhao, Yuzhu Mao, Yang Liu, Linqi Song, Ye Ouyang, Xinlei Chen, and Wenbo Ding.

Towards efficient communications in federated learning: A contemporary survey. *Journal of the Franklin Institute*, 360(12):8669–8703, 2023.

[176] Zhifeng Jiang, Wei Wang, Bo Li, and Qiang Yang. Towards efficient synchronous federated training: A survey on system optimization strategies. *IEEE Transactions on Big Data*, 9(2): 437–454, 2022.

[177] Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications (INFOCOM)*, pages 1739–1748. IEEE, 2022.

[178] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning (ICML)*, pages 3407–3416. PMLR, 2021.

[179] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2021–2031. PMLR, 2020.

[180] Pengchao Han, Shiqiang Wang, and Kin K Leung. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 300–310. IEEE, 2020.

[181] Negar Foroutan Eghlidi and Martin Jaggi. Sparse communication for training deep networks. *arXiv preprint arXiv:2009.09271*, 2020.

[182] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[183] Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated sgd under local differential privacy with top-k dimension selection. In *Database Systems for Advanced Applications: 25th International Conference (DASFAA)*, pages 485–501. Springer, 2020.

[184] Shiwei Lu, Ruihu Li, Wenbin Liu, Chaofeng Guan, and Xiaopeng Yang. Top-k sparsification with secure aggregation for privacy-preserving federated learning. *Computers & Security*, 124:102993, 2023.

[185] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[186] Shaohuai Shi, Xiaowen Chu, Ka Chun Cheung, and Simon See. Understanding top-k sparsification in distributed deep learning. *arXiv preprint arXiv:1911.08772*, 2019.

[187] Nir Shlezinger, Mingzhe Chen, Yonina C Eldar, H Vincent Poor, and Shuguang Cui. Uveqfed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, 69:500–514, 2020.

[188] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[189] Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International Conference on Machine Learning (ICML)*, pages 3329–3337. PMLR, 2017.

[190] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021*, 2017.

[191] Zihao Zhou, Yanan Li, Xuebin Ren, and Shusen Yang. Towards efficient and stable k-

asynchronous federated learning with unbounded stale gradients on non-iid data. *IEEE Transactions on Parallel and Distributed Systems*, 33(12):3291–3305, 2022.

[192] Yuchen Liu, Chen Chen, Lingjuan Lyu, Fangzhao Wu, Sai Wu, and Gang Chen. Byzantine-robust learning on heterogeneous data via gradient splitting. In *International Conference on Machine Learning (ICML)*, pages 21404–21425. PMLR, 2023.

[193] Banghua Zhu, Lun Wang, Qi Pang, Shuai Wang, Jiantao Jiao, Dawn Song, and Michael I Jordan. Byzantine-robust federated learning with optimal statistical rates. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 3151–3178. PMLR, 2023.

[194] Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on neural networks and learning systems*, 31(10):4229–4238, 2019.

[195] Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

[196] Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

[197] Ioannis Mitliagkas, Ce Zhang, Stefan Hadjis, and Christopher Ré. Asynchrony begets momentum, with an application to deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 997–1004. IEEE, 2016.

[198] Chen Dun, Mirian Hipolito, Chris Jermaine, Dimitrios Dimitriadis, and Anastasios Kyrillidis. Efficient and light-weight federated learning via asynchronous distributed dropout. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 6630–6660. PMLR, 2023.

[199] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

[200] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 26091–26102, 2021.

[201] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning (ICML)*, pages 5650–5659. PMLR, 2018.

[202] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning (ICML)*, pages 3521–3530. PMLR, 2018.

[203] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning (ICML)*, pages 6246–6283. PMLR, 2022.

[204] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations (ICLR)*, 2021.

[205] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

[206] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. In *The Eleventh International Conference on Learning*

Representations, 2022.
[207] Nikita Fedin and Eduard Gorbunov. Byzantine-robust loopless stochastic variance-reduced gradient. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 39–53. Springer, 2023.
[208] Lili Su and Nitin H Vaidya. Fault-tolerant distributed optimization (part iv): Constrained optimization with arbitrary directed networks. *arXiv preprint arXiv:1511.01821*, 2015.
[209] Cheng Fang, Zhixiong Yang, and Waheed U Bajwa. Bridge: Byzantine-resilient decentralized gradient descent. *IEEE Transactions on Signal and Information Processing over Networks*, 8:610–626, 2022.
[210] Zhixiong Yang and Waheed U Bajwa. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*, 5(4):611–627, 2019.
[211] Jie Peng, Weiyu Li, and Qing Ling. Byzantine-robust decentralized stochastic optimization over static and time-varying networks. *Signal Processing*, 183:108020, 2021.
[212] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust decentralized learning via self-centered clipping. *arXiv preprint arXiv:2202.01545*, 2022.
[213] Zhaoxian Wu, Tianyi Chen, and Qing Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE transactions on signal processing*, 2023.
[214] Holger R Roth, Yan Cheng, Yuhong Wen, Isaac Yang, Ziyue Xu, Yuan-Ting Hsieh, Kristopher Kersten, Ahmed Harouni, Can Zhao, Kevin Lu, et al. Nvidia flare: Federated learning from simulation to real-world. *arXiv preprint arXiv:2210.13291*, 2022.
[215] Jason Posner, Lewis Tseng, Moayad Aloqaily, and Yaser Jararweh. Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network*, 35(2):152–159, 2021.
[216] Stefano Savazzi, Monica Nicoli, Mehdi Bennis, Sanaz Kianoush, and Luca Barbieri. Opportunities of federated learning in connected, cooperative, and automated industrial systems. *IEEE Communications Magazine*, 59(2):16–21, 2021.
[217] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. Privacy preserving vertical federated learning for tree-based models. *arXiv preprint arXiv:2008.06170*, 2020.
[218] Wenjie Li, Zhongren Wang, Jinpeng Wang, Shu-Tao Xia, Jile Zhu, Mingjian Chen, Jiangke Fan, Jia Cheng, and Jun Lei. Refer: Retrieval-enhanced vertical federated recommendation for full set user benefit. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1763–1773, 2024.
[219] Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. Fedbiot: Llm local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3345–3355, 2024.
[220] Yan Kang, Tao Fan, Hanlin Gu, Lixin Fan, and Qiang Yang. Grounding foundation models through federated transfer learning: A general framework. *arXiv preprint arXiv:2311.17431*, 2023.
[221] Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 3, 2021.
[222] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, 2011.
[223] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.
[224] Tong Yu and Hong Zhu. Hyper-parameter optimization: A review of algorithms and ap-

plications. *arXiv preprint arXiv:2003.05689*, 2020.

[225] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 10293–10301, 2021.

[226] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4):1–34, 2021.

[227] Qingsong Zhang, Fengxiang He, Jindong Gu, Bin Gu, Cheng Deng, Heng Huang, and Dacheng Tao. Bambi: Vertical federated bilevel optimization with privacy-preserving and computation efficiency. 2022.

[228] Junyi Li, Feihu Huang, and Heng Huang. Communication-efficient federated bilevel optimization with global and local lower level problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

[229] Minhui Huang, Dewei Zhang, and Kaiyi Ji. Achieving linear speedup in non-iid federated bilevel learning. In *International Conference on Machine Learning (ICML)*, pages 14039–14059. PMLR, 2023.

[230] Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. Fednest: Federated bilevel, minimax, and compositional optimization. In *International Conference on Machine Learning*, pages 21146–21179. PMLR, 2022.

[231] Yifan Yang, Peiyao Xiao, and Kaiyi Ji. Simfbo: Towards simple, flexible and communication-efficient federated bilevel learning. In *Advances in Neural Information Processing Systems*, 2024.

[232] Ji Liu, Tianshi Che, Yang Zhou, Ruoming Jin, Huaiyu Dai, Dejing Dou, and Patrick Valduriez. Aedfl: Efficient asynchronous decentralized federated learning with heterogeneous devices. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 833–841.

[233] Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security (CCS)*, pages 1575–1590, 2020.

[234] Swanand Kadhe, Nived Rajaraman, O Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020.

[235] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51 (4):1–35, 2018.

[236] Jing Ma, Si-Ahmed Naas, Stephan Sigg, and Xixiang Lyu. Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*, 37(9):5880–5901, 2022.

[237] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 USENIX annual technical conference (USENIX ATC)*, pages 493–506, 2020.

[238] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191, 2017.

[239] Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. The poisson binomial mechanism for unbiased federated learning with secure aggregation. In *International Conference on Machine*

Learning, pages 3490–3506. PMLR, 2022.