
WHY FEDERATED OPTIMIZATION FAILS TO ACHIEVE PERFECT FITTING ?

A THEORETICAL PERSPECTIVE ON CLIENT-SIDE OPTIMA

Zhongxiang Lei
Beijing Institute of Technology
zxlei@bit.edu.cn

Qi Yang
Beijing Institute of Technology
qyang@bit.edu.cn

Ping Qiu
Beijing Institute of Technology
qiuping@njupt.edu.cn

Gang Zhang
Beijing Institute of Technology
zhanggang@bit.edu.cn

Yuanchi Ma
Beijing Institute of Technology
yma@bit.edu.cn

Jinyan Liu
Beijing Institute of Technology
jyliu@bit.edu.cn

November 4, 2025

ABSTRACT

Federated optimization is a constrained form of distributed optimization that enables training a global model without directly sharing client data. Although existing algorithms can guarantee convergence in theory and often achieve stable training in practice, the reasons behind performance degradation under data heterogeneity remain unclear. To address this gap, the main contribution of this paper is to provide a theoretical perspective that explains why such degradation occurs. We introduce the assumption that heterogeneous client data lead to distinct local optima, and show that this assumption implies two key consequences: 1) the distance among clients' local optima raises the lower bound of the global objective, making perfect fitting of all client data impossible; and 2) in the final training stage, the global model oscillates within a region instead of converging to a single optimum, limiting its ability to fully fit the data. These results provide a principled explanation for performance degradation in non-iid settings, which we further validate through experiments across multiple tasks and neural network architectures. The framework used in this paper is open-sourced at: <https://github.com/NPCLEI/fedtorch>.

1 Introduction

Federated optimization (FO) [13, 12] research in a distributed system explores how to optimize machine learning models without requiring clients' data to participate in communications, with neural network models being the most common. The objective function of FO is:

$$\min_{x \in \mathbb{R}^d} F(x) \quad \text{where} \quad F(x) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} f(x; D_i). \quad (1)$$

Here, x is the machine learning model, $\mathcal{S}$ is the set of all clients, D_i is data set of client i and f is the loss function of x on D_i ($f(x; D_i)$ as f_i for short). The most classic problem faced by FO is that the distribution of client data D_i cannot represent the overall distribution, leading to a deterioration in the performance of the optimized model, also known as the non-iid problem.

The main contribution of this paper is to provide a theoretical explanation for the performance degradation of federated optimization under non-iid settings. As illustrated in Fig. 1, although most existing federated optimization algorithms can theoretically guarantee convergence and often achieve stable convergence in practice, the models still exhibit significant underfitting in non-iid scenarios (blue curve), consistently performing worse than in the iid case (purple curve). This phenomenon has not yet been convincingly explained in the current literature. Methods such as SCAFFOLD [10]

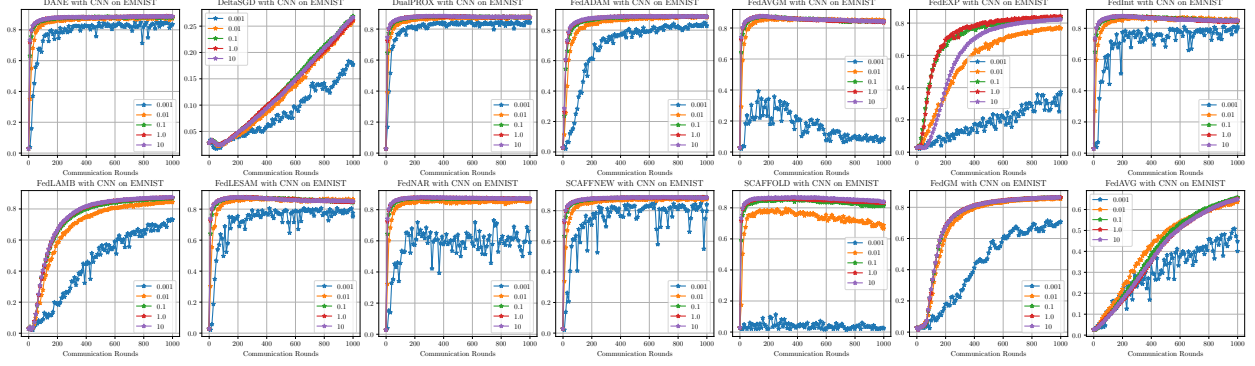


Figure 1: The performance of FedAVG, FedAVGM [1], DeltaSGD [2], FedRed [3], FedEXP [4], FedGM [5], FedInit [6], FedLESAM [7], FedNAR [8], FedPROX [9], SCAFFOLD [10], SCAFFNEW [11], and FedADAM [12] on the EMNIST classification task. These experiments demonstrate a common phenomenon: These algorithms converge to the stationary point, but their final performance deteriorates due to increased heterogeneity. Where α is the Dirichlet distribution parameter, commonly used to simulate heterogeneity.

and FedProx [9] generally attribute the issue to the “drift” of local client updates; however, this perspective fails to fully capture the root cause of the degradation. Even with the improvements proposed by FedAvgM [1], FedRed [3], SCAFFNEW [11] and related approaches, experimental results demonstrate that the performance deterioration under non-iid conditions remains persistent.

1.1 Theoretical conclusions

Based on experimental observations, we find that training data across different clients can drive the model to converge to distinct local optima. Motivated by this phenomenon, this paper introduces a novel perspective for **analyzing the convergence behavior of federated optimization within a single communication round t** . This theoretical framework allows us to rigorously establish the following key conclusions:

- **Lower Bound from Local Optima Heterogeneity (Theorem 3.1):** Due to heterogeneity among local optima, a significant lower bound exists for Equation 1, indicating that the global model cannot perfectly fit the data. This phenomenon may lead to underfitting in non-i.i.d. scenarios. The conclusion is validated experimentally on multiple complex neural networks, including GRU, ResNet-18, ViT, and Deepseek.
- **Oscillatory Region (Theorem 4.1):** During the final stage of convergence, an oscillatory region emerges. Once the global model x_t enters this region, its trajectory exhibits pronounced oscillations, making convergence difficult without gradually decaying the local learning rate to zero. Visual experiments in Fig. 1 and several convex function cases corroborate this finding. The theory also elucidates the impact of variable local update rounds, client weights, and participation rates on convergence.
- **Trajectory Correction Condition (Theorem 4.3):** Consistent with prior studies [1, 3], the effectiveness of the correction term h_t depends on two factors: its norm must not exceed the distance between the global model and the local optimum, and its direction must align with the trajectory.
- **Momentum and Adaptive Learning (Theorem 4.5):** Our theory offers a fresh perspective on the roles of momentum and learning rate in adaptive methods. Momentum increases the descent distance of a single update, accelerating overall convergence, while adaptive learning rates, though having a decaying effect on distance, can also enhance descent magnitude.

1.2 Research Setting for a Cluster of Federated Optimization Methods

Federated optimization has made significant progress, and the theory proposed in this paper can be shared by abstracting it as much as possible into Algorithm 1. Specifically, based on whether there are correction terms and whether there are adaptive optimization methods, the following three categories of work can be summarized:

Algorithm 1 A cluster of federated optimization methods

Input: Global step η for adaptive optimization to update x_t ; Local learning rate η_l^t to optimize local model; Communication rounds T ; Local update rounds K_i .

Output: Global model parameter x .

```

1: Initialization: Global model parameters  $x_0$ 
2: for  $t = 0, \dots, T$  do
3:   The server connects any number of active clients to form the subset  $\mathcal{S}_t$  and sends  $x_t$  to clients.
4:   for  $i \in \mathcal{S}_t$  in parallel do
5:     Starting from the point  $x_t$ , client  $i$  updates the model  $K_i$  times and obtain the updated state  $x_{i,K}^t$  with learning
       rate  $\eta_l^i$  and optional correction direction  $h_t$ .
6:   end for
7:    $x_{t+1} = x_t - \eta_t \sum_{i \in \mathcal{S}_t} \rho_i^t (x_t - x_{i,K}^t)$ .
8:   s.t.  $\sum_{i \in \mathcal{S}_t} \rho_i^t = 1, \rho_i^t \geq 0$ 
9: end for
10: return  $x_T$ 

```

1.2.1 Local adaptive optimization methods (LA)

When the algorithm does not have local correction terms, i.e., $\eta_t = 1$ and $h_t = 0$ in Algorithm 1, these algorithms optimize the model locally using some optimization algorithms, such as gradient clipping [14] and various adaptive optimization methods [15, 16]. The most classic method is FedAVG [13], which uses SGD training on the client side and aggregates samples as weights on the server side.

1.2.2 Drift correction methods (DC)

Some methods assume that the training trajectories on the client side exhibit “drift,” so they introduce a drift correction term to correct the trajectories. This paper simplifies this drift correction term to h_t . Note that regularization methods can also be viewed as correction terms, such as $f_i + \|x - x_t\|^2$, where the correction term can be treated as a correction term when computing the gradient. Representative works include SCAFFNEW, FedAVGM, FedRed, and SCAFFOLD [11, 1, 3, 10], which correspond to the case where $h_t \neq 0$ (with η being arbitrary) in Algorithm 1.

1.2.3 Server Adaptive optimization methods (SA)

Some studies have introduced highly successful adaptive optimization methods for neural networks into federated learning. For example, FedGM [5] uses the Heavy ball and Nesterov methods to accelerate training, while FedOPT, FedExp, Fed-EF, and FedAMS [12, 4, 17, 18] employ adaptive learning rate methods. They all utilize the pseudo-gradient $\mathbb{G}_{\mathcal{S}_t} = \sum_{i \in \mathcal{S}_t} \rho_i^t (x_t - x_{i,K}^t)$ to replace the true gradient in these methods for optimization.

It is worth noting that there are many variations of these methods. This article analyzes the common elements of the above methods and may not cover all versions.

2 Basic Assumptions of Heterogeneity of Local Optimal Points

Our goal is to propose a theoretical analysis that focuses solely on the parameters of federated learning, such as heterogeneity, local training rounds, client participation rate, and aggregation weights, etc. The challenge is how to formulate an assumption that covers all settings for analysis, given that the data distribution on clients and the function properties are unknown. A key observation is that: **the client’s data D_i differs from other clients**, which implies that for clients i and j , the parameters that minimize their respective target functions f_i and f_j are distinct; in other words, most their local optimal points satisfy $x_i^* \neq x_j^*$. We conduct extensive experiments to verify this conjecture. In Fig. 2(left), we plotted the loss landscape on the client side and the relative positions of the approximated optimal points of the neural network model at different rounds (Fig. 2(right)). The experiments demonstrated that there are multiple local optima on the client side. Based on this, we formally propose the following assumption:

Assumption 2.1. (*Heterogeneity of Local Optimal Points*) Let that $x_i^* = \arg_x \min f(x; x_t, \mathcal{D}_i)$ is the local optimal point of f_i achieved by client i start optimization at x_t . The average distance of local optimal points satisfies the

¹In these experiments, we optimize the local loss function f_i to 1^{-2} or the point where the loss does not change anymore as an approximation to the local optimum x_i^* .

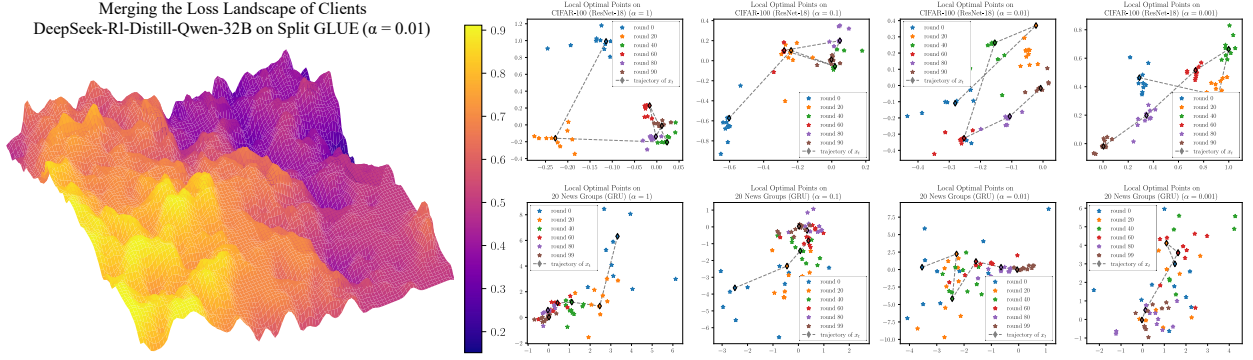


Figure 2: The two figures on the left and right support the view that the local optimal points of the clients is heterogeneous. The left figure merges the loss landscapes of all f_i into a single plot using the aggregation method $g(x) = \min(f_1(x), f_2(x), f_{\dots}(x))$. The right figure demonstrates the changes in the relative positions of local optimal points (approximation¹) under different tasks and rounds.

following relationship:

$$\mathcal{H}(\{x^*\}_{i \in \mathcal{S}}) = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \|x_i^* - x_{\mathcal{S}}^*\|^2 \geq 0, \quad (2)$$

where $x_{\mathcal{S}}^* = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} x_i^*$ is center of local optimal points. The equality holds if and only if all client data are i.i.d. It should be noted that in non-convex settings, x_i^* may change as x_t changes. Unlike convergence analysis, the conclusions in this paper focus only on the behavior at time t , so the change of x_i^* does not affect the subsequent results.

Assumption 2.2. (Locally Efficient Descent Condition) Define the initial vector pointing to the local optimum as $\delta_{t,i} = x^t - x_i^*$. Let $x_{i,K}^t$ represent the updated model after client-side updates. The updated vector pointing to the local optimum is then defined as $\delta_{i,K}^t = x_{i,K}^t - x_i^*$. A client i is considered effectively descending if it satisfies the condition:

$$\|\delta_{i,K}^t\| = \sigma_i^t(K, \eta_i) \|\delta_{t,i}\|, 0 \leq \sigma_i^t(K, \eta_i) \leq 1,$$

where $\sigma_i^t(K, \eta_i)$ (σ_i^t for short) comprehensively incorporates properties of f_i and some properties related to the optimization method used by client i .

Assumption 2.2 effectively bypasses assumptions about the properties of function f_i and its gradient in $\sigma_i^t(K, \eta_i)$, allowing us to focus more on the parameters of FL rather than the specific properties of the function. For example, σ_i^t usually has a relationship where the larger K_i is, the closer σ_i^t is to 0. For methods using adaptive optimization neural networks, σ_i^t is related to the descent accuracy. For methods that use adaptive optimization methods to optimize neural networks, σ_i^t is related to the descent precision. For example, when using adaptive methods such as Adam [19, 20] and Adagrad [21], $\sigma_i^t = \mathcal{O}(K^{-1/2})$ is below this order of magnitude. And ordinary gradient methods can guarantee that $\|\delta_{i,K}^t\| \leq (\frac{Q_f-1}{Q_f+1})^K \|\delta_{t,i}\|$ where Q_f is related to the properties of the function (strong convexity μ , smoothness L , etc.). Nesterov's acceleration method [22, 23] can guarantee that $\sigma_i^t = \mathcal{O}(K^{-2})$. For general smooth cases, often $\sigma_i^t = \mathcal{O}(K^{-1/2})$, which can all be found in Nesterov's Lectures [24].

3 Lower Bound from Local Optima Heterogeneity

Although neural networks are inherently non-convex, their loss landscape near optimal points can be approximated as convex. Theoretical support comes from two sources: 1) Under certain conditions [25, 26], neural networks are universal function approximators, enabling clients to reach local optima $f(x_i^*)$; and 2) Visualization studies [27] reveal that near these optima, ResNets exhibit convex-like loss landscapes characterized by positive Hessian eigenvalues.

Theorem 3.1. (Lower Bound of Objective Function) If $f(x; D_i)$ can be approximated as a convex function within the neighborhood $U_i(x_i^*)$ around x_i^* , then $\nabla f_i = 0$ and $\nabla^2 f_i$ is positive definite. For all $x \in U_0 \cap U_1 \cap \dots$, the lower bound of Eq. 1 is:

$$F(x) \geq \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left(f(x_i^*) + \frac{\lambda_{\min}^i}{2} \left(\|x_i^* - x_{\mathcal{S}}^*\| - \|x - x_{\mathcal{S}}^*\| \right)^2 \right) \quad (3)$$

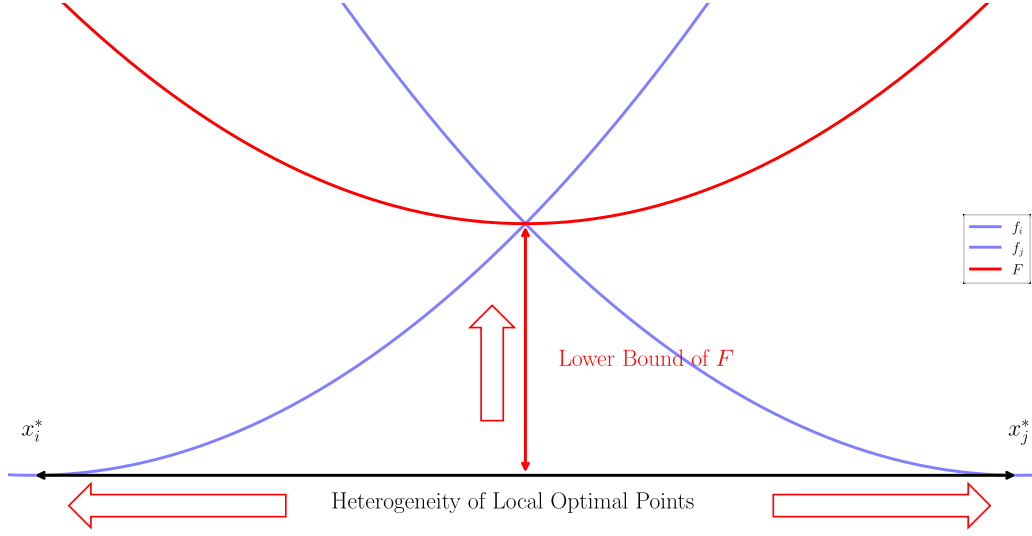


Figure 3: The lower bound of Eq. 1(F) will be pulled up by distant of local optimal points.

where $\lambda_{\min}^i > 0$ is the smallest eigenvalue of $\nabla^2 f_i(x_i^* + t(x - x_i^*))$, $t \in (0, 1)$.

Proof. Apply the function by Taylor's theory [28] to f_i :

$$\begin{aligned}
 f_i(x_i^* + (x - x_i^*)) &= f_i(x_i^*) + (x - x_i^*)^\top \nabla f_i(x_i^*) + \frac{1}{2}(x - x_i^*)^\top \nabla^2 f_i(z_i)(x - x_i^*) \\
 &= f_i(x_i^*) + \frac{1}{2}(x - x_i^*)^\top \nabla^2 f_i(z_i)(x - x_i^*) \\
 &= f_i(x_i^*) + \frac{1}{2}((x - x_S^*) - (x_i^* - x_S^*))^\top \nabla^2 f_i(z_i)((x - x_S^*) - (x_i^* - x_S^*)) \\
 &\geq f_i(x_i^*) + \frac{\lambda_{\min}^i}{2} \|(x - x_S^*) - (x_i^* - x_S^*)\|^2 \\
 &\geq f_i(x_i^*) + \frac{\lambda_{\min}^i}{2} \left| \|(x - x_S^*)\| - \|(x_i^* - x_S^*)\| \right|^2
 \end{aligned}$$

where $\lambda_{\min}^i > 0$ is the smallest eigenvalue of the positive definite matrix $\nabla^2 f_i(x_i^* + t(x - x_i^*))$, $t \in (0, 1)$. $\square$

3.1 Results Analysis and Experimental Validation

α	GRU on 20 NewsGroups			Resnet 18 on CIFAR100			Vit-large on ImageNet 1K			DeepSeek-R1-32B on GLUE		
	F^*	acc.	$\mathcal{H}$	F^*	acc.	$\mathcal{H}$	F^*	acc.	$\mathcal{H}$	F^*	acc.	$\mathcal{H}$
0.001	0.79	44.19 %	0.63	1.18	47.86 %	1.93	0.59	77.70 %	0.24	1.62	60.45 %	1.57
0.01	0.35	64.67 %	0.39	0.54	52.57 %	1.08	0.44	79.87 %	0.19	1.73	70.86 %	1.16
0.1	0.04	74.70 %	0.21	0.50	54.16 %	0.97	0.33	80.34 %	0.17	0.79	78.10 %	0.82
1	0.03	75.16 %	0.20	0.22	57.41 %	1.04	0.33	80.71 %	0.14	0.48	81.24 %	0.73
10	0.03	74.83 %	0.17	0.21	56.97 %	1.10	0.33	80.72 %	0.15	0.42	82.86 %	0.71
$Per(*, \mathcal{H})$	0.997	-0.990	-	0.895	-0.838	-	0.962	-0.971	-	0.888	-0.997	-

Table 1: Approximate value of $F(x^*)$ (F^*), test accuracy (acc.), $\mathcal{H}(\{x^*\}_{i \in \mathcal{S}})$ ($\mathcal{H}$) and the parameter α of the Dirichlet distribution exhibit correlations across various neural network tasks. $Per(*, \mathcal{H})$ represents the Pearson correlation coefficient between the test accuracy column and the $\mathcal{H}$ column, or between the F^* column and the $\mathcal{H}$ column.

The lower bound of the objective function Eq. 1 of FL is bounded by two inherent limitations: the first is the degree to which the local function $f(x_i^*)$ is fitted, and the second is the heterogeneity of the local optimal points, as defined in

Assumption 2.1. Even if a client fully fits its local data set such that $f(x_i^*; D_i) = 0$, the lower bound of the objective function will still be constrained by the distance between the local optima, preventing Eq 1 from perfectly fitting all data $\{D\}_{i \in \mathcal{S}}$. As long as Assumption 2.1 is not equal to zero, Eq. 1 is unlikely to be zero.

To verify whether this theory is effective for neural networks, we designed an experiment that was conducted on a variety of complex tasks and neural network structures (including shallow RNNs, Resnet-18, large Vision Transformers [29], and the recently popular large network Deepseek-R1-32B [30]). The experimental results presented in Table 1 demonstrate a correlation among the objective function value $F(x^*)$, the distant of local optimal points $\mathcal{H}(\{x^*\}_{i \in \mathcal{S}})$, test accuracy (acc.), and α , thereby providing empirical validation for Theorem 3.1. This lower bound also explains why FL can sometimes prevent overfitting, but at other times it can lead to underfitting.

4 Heterogeneity Theorems of Local Optimal Points

This section is divided into three theories, which are presented in a step-by-step manner. Theorem 4.1 analyzes the single-step convergence conclusions when arbitrary optimization methods are used on the client side and the FedAVG algorithm is used on the server side, and provides separate analyses of trajectory analysis, the role of the local update times K , the role of weights, and the role of participation rates. Theorem 4.3 examines the effect of the correction term h_t on single-step convergence when it is present on the client side. Theorem 4.5 analyzes the roles of momentum and adaptive learning rate in SA for single-step updates.

4.1 Theorem of LA-FedAVG: Trajectory and Role of Factors Analysis

The advantage of Assumption 2.2 is that it allows analysis when participating clients optimize models using different optimization methods, because different optimization methods can ultimately calculate different $\sigma_i^t(K, \eta)$. Then, when clients optimize using any optimization method and there is no correction term (i.e., h_t), the Local adaptive optimization methods (LA) mentioned in Chapter 2 have the following theoretical analysis:

Theorem 4.1. (LA-FedAVG) Assuming that local optimal points satisfy Assumption 2.1 and all clients $i \in \mathcal{S}_t$ satisfy the effective descent condition 2.2. The single update distance from x_{t+1} of LA-FedAVG to the weighted-sampled local optimal point $x_{\mathcal{S}_t}^* = \sum_{i \in \mathcal{S}_t} \rho_i x_i^*$ is:

$$\|x_{t+1} - x_{\mathcal{S}_t}^*\|^2 = \|x_t - x_{\mathcal{S}_t}^*\|^2 - \frac{1}{|\mathcal{S}_t|^2} (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t})^\top A (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t}), \quad (4)$$

where $\mathcal{P}_{\mathcal{S}_t} = [\rho_i^t]_{i \in \mathcal{S}_t}$ is weight vector, $X_{\mathcal{S}_t} = [\delta_{t,i}]_{i \in \mathcal{S}_t}$, and the element of matrix A is $A_{i,j} = \cos \langle \delta_{t,i}, \delta_{t,j} \rangle - \sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle$.

4.1.1 Trajectory Analysis

To analyze whether x_{t+1} moves closer to $x_{\mathcal{S}_t}^*$ than x_t , the key is to analyze when the quadratic form is positive definite: $\Delta_{t+1}(X_{\mathcal{S}_t}) = \frac{1}{|\mathcal{S}_t|^2} (\mathcal{P}_{\mathcal{S}_t} X_{\mathcal{S}_t})^\top A (\mathcal{P}_{\mathcal{S}_t} X_{\mathcal{S}_t}) > 0$. Analyzing this matrix as a whole is extremely complex, particularly due to the interdependence of the vector angles and the fact that research [31] considers the expectation of quadratic forms to be chaotic, making it difficult to establish precise conditions for A being positive definite. However, we can still observe the characteristics of matrix elements to analyze trajectory information.

We observe that $\mathcal{P}_{\mathcal{S}_t} X_{\mathcal{S}_t} \geq 0$, $A_{i,i} = 1 - \sigma_{i,t}^2 \geq 0$ and A is the Hessian matrix of quadratic form. Hence A directly determines whether the surface of the quadratic function is convex or non-convex (in the top right corner of Fig.4). In particular, if the quadratic form is everywhere convex, then A is positive semi-definite; If the quadratic form exhibits non-convexity, it may nevertheless remain nonnegative in regions close to the coordinate axes: $\lim_{X_{\mathcal{S}_t}[i] \rightarrow 0} \Delta_{t+1} =$

$A_{j,j} (\mathcal{P}_{\mathcal{S}_t}[j] \cdot X_{\mathcal{S}_t}[j])^2 \geq 0$. From a numerical standpoint, whenever the following inequality holds: $\sum_{i \in \mathcal{S}_t} \rho_i (1 - \sigma_{i,t}^2) \|\delta_{t,i}\|^2 \geq \sum_{j \in \mathcal{S}_t} \sum_{i \in \mathcal{S}_t, i \neq j} \rho_i \rho_j (\sigma_{i,t} \sigma_{j,t} \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle) \|\delta_{t,i}\| \|\delta_{t,j}\|$, The quadratic form is positive semi-definite. Because of $A_{i,i} \geq 0$, we can only conclude that A is likely positive definite when "most" of its off-diagonal elements satisfy $A_{i,j} \geq 0 \Rightarrow \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \geq \sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle$. The above information is enough for us to analyze the optimization trajectory of x_t . Ignoring σ_i^t and σ_j^t temporarily, if the local optimal points do not change with variations in x_t , then we can deduce that:

Stage 1: When x_t is far away from both x_i^* and x_j^* , we can obtain $\cos \langle \delta_{t,i}, \delta_{t,j} \rangle \approx 1 \Rightarrow A_{i,j} \geq 0$. In this case, A is semi-positive definite and x_{t+1} approaches $x_{\mathcal{S}_t}^*$.

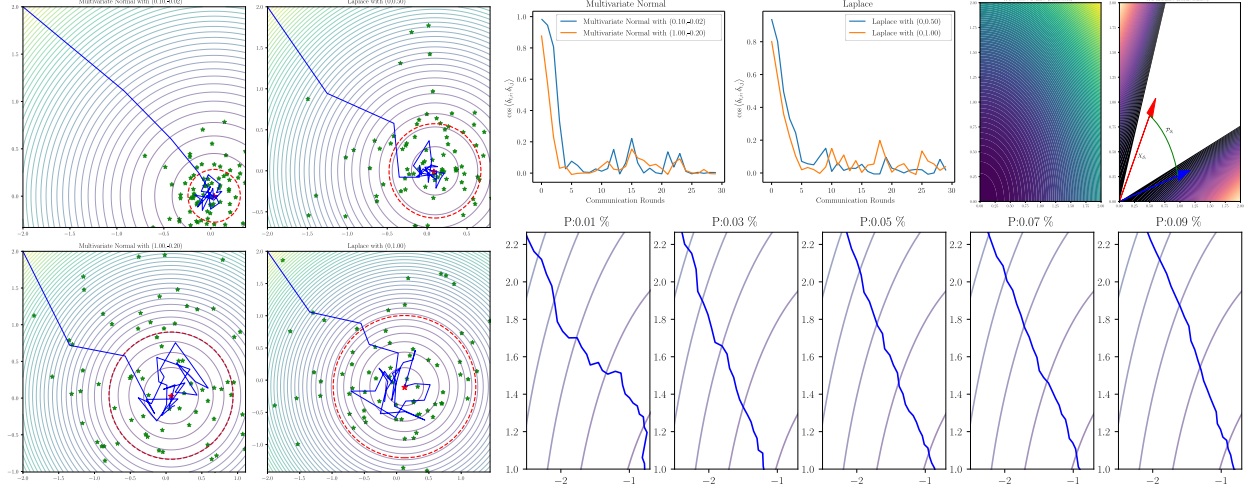


Figure 4: The left four diagrams represent the contour plots of paraboloid surface f , where the green points indicate the local optimal point of f_i obtained by sampling using both multivariate Gaussian and Laplace distributions. The blue line shows the optimization trajectory of x_t , while the red dashed lines outline the oscillatory region. In the upper right corner, the left plot shows the communication rounds on the horizontal axis and the average value of $\cos \langle \delta_{t,i}, \delta_{t,j} \rangle$ on the vertical axis. The two contour plots in the upper right corner explain the effect of the positive definiteness of A on the range of quadratic forms. The white areas indicate regions where the values are less than zero, the red vectors represent X_{S_t} before being deflected by $\mathcal{P}_{S_t}$, and the blue vectors represent the deflected ones. The lower right section presents the trajectories for P ranging from 0.01% to 0.09%.

Stage 2: As x_t gradually moves closer to x_i^* and x_j^* , with the angle between them increasing, $A_{i,j} > 0$ no longer holds for most elements, making matrix A indefinite. Consequently, x_{t+1} begins to move away from $x_{S_t}^*$.

Stage 3: Once x_t moves far away again from x_i and x_j , the angle between them decreases, and x_{t+1} starts to approach $x_{S_t}^*$ once more. This indicates that x_t 's trajectory begins to **oscillate** around $x_{S_t}^*$.

4.1.2 Oscillatory region and Heterogeneity

When the mean vector angle between x_t and x_i^*, x_j^* is greater than or equal to 90 degrees, it forms a spherical region with $\|x_{t,i}^* - x_{t,j}^*\|$ as its diameter, we call this region is **oscillatory region**:

$$\|x_t - x_{S_t}^*\| = \frac{1}{2} \|x_j^* - x_i^*\| \text{ if } \cos \langle \delta_{t,i}, \delta_{t,j} \rangle = 0.$$

As shown by the red dashed line (centered at $x_{S_t}^*$ with radius $\frac{1}{2|S|(|S|-1)} \sum_{i \in S} \sum_{j \in S, j \neq i} \|x_j^* - x_i^*\|$) on the left side of Fig. 4, the experimental results on the paraboloid fully support our analysis of the trajectories. Once x_t enters this region of the sphere, σ_i^t and σ_j^t become small enough to fail (refer to The Role of K_i). At this point, even if the client updates faster, it becomes ineffective, causing the trajectory of x_t to begin oscillating. In the worst case, this can lead to A becoming indefinite, resulting in the trajectory diverging outward. This also explains why gradient heterogeneity theories suggest that the local learning rate $\eta_l = \mathcal{O}(\frac{1}{T})$ is required.

4.1.3 The Role of Local Update Rounds $K_i(\sigma_i^t)$

One of the original intentions behind FL [13] design is to increase the local training times K_i of clients to save communication times and costs. If the local optimum point $x_{i,t}^*$ is fixed (e.g. f_i is strong convex) and unchanged, the smaller σ_i^K (the larger K) before x_t enters the oscillatory region, the better. However, in the case of neural networks, the local optimum point will change with the change of x_t , and a larger K may make the algorithm prematurely fall into the poor-performance optimal point.

4.1.4 The Role of Weights

There are two places where weights influence Eq. 4. The first is in the calculation of $x_{S_t}^*$, which computes the weight center of the local optimal points in S_t and affects the descent direction of x_{t+1} . The second is $\mathcal{P}_{S_t} \cdot X_{S_t}$, where since the sum of the weight vector is 1, its role is to deflect the angle of the vector X_{S_t} by a certain degree Fig. 4 (top right corner). This determines whether the descent distance of x_{t+1} is more biased towards which client $:\lim_{\substack{\mathcal{P}_{S_t}[i] \rightarrow 1 \\ i \neq j}} \Delta_{t+1} = A_{j,j} (X_{S_t}[j])^2 \geq 0$.

4.1.5 The Role of Participation Rate

In Theorem 4.1, the main term that participates in this participation rate is $x_{S_t}^*$. This term represents an estimate of the weighted center x_S^* of all local optima. As shown in the lower right corner of Fig. 4, if the participation rate is too low, it can lead to a certain bias in each estimation, which may result in oscillations in the trajectory. A high participation rate, on the other hand, does not affect the estimate of the weighted center.

4.2 Theorem of Drift Correction Methods

When the client side uses the SGD method to optimize the model and there is a correction term $h_t \neq 0$, the DC method in Chapter 2 has the following theoretical analysis when the server side uses SGD to update the global model.

Lemma 4.2. *Pseudo-gradient $\mathbb{G}_{S_t} = \sum_{i \in S_t} \rho_i^t (x_t - x_{i,K}^t)$ can be regarded as the update direction of the LA-FedAVG algorithm: $x_{t+1}^{LA-FedAVG} = x_t - \sum_{i \in S_t} \rho_i^t (x_t - x_{i,K}^t) \Rightarrow x_{t+1}^{LA-FedAVG} - x_t = -\mathbb{G}_{S_t}$. If the matrix A is positive definite, then the pseudo-gradient direction points in the direction that takes x_{t+1} away from x_{S_t} .*

Theorem 4.3. (DC-FedSGD) *Assuming the clients satisfy Assumptions 2.1 and 2.2, and the updating model takes the form of $x_{i,K}^{h,t} = x_t - \eta_l^i \sum_{i=0}^{K_i-1} \nabla f_i + \eta_l^i K_i h_t = x_{i,K}^t + h_t^i$. Let $\delta_{S_t}^K = x_{t+1} - x_{S_t}^*$ and $\delta_{S_t} = x_t - x_{S_t}^*$, the single-round update distance is:*

$$\|x_{t+1} - x_{S_t}^*\|^2 = (1 - \eta_t \wp) \|x_t - x_{S_t}^*\|^2 - \eta_t (\eta_t \Delta_{t+1} + \bar{\wp} \|\mathbb{H}_{S_t}\|), \quad (5)$$

where $\wp = 2(1 - \eta_t)(1 - \cos\langle \delta_{S_t}^K, \delta_{S_t} \rangle \sigma_\Delta)$, $\bar{\wp} = [2h \|\delta_{S_t}\| - \eta_t \|\mathbb{H}_{S_t}\|]$, $h = -\eta_t \sigma_\Delta \cos\langle \delta_{S_t}^K, \mathbb{H}_{S_t} \rangle + (\eta_t - 1) \cos\langle \delta_{S_t}, \mathbb{H}_{S_t} \rangle$ and $\sigma_{\Delta_{t+1}} = \sqrt{\|\delta_{S_t}\| - \Delta_{t+1} / \|\delta_{S_t}\|}$, $\mathbb{H}_{S_t} = \sum_{i \in S_t} \rho_i h_t^i$.

Corollary 4.4. (Effective Condition of $\mathbb{H}_{S_t}$) *Whether the correction direction $\mathbb{H}_{S_t}$ is absolutely effective depends on if $\bar{\wp} > 0 \Rightarrow \|\mathbb{H}_{S_t}\| < 2 \frac{h}{\eta_t} \|\delta_{S_t}\|$ and $\frac{1-\eta_t}{\eta_t \sigma_\Delta} \cos\langle \delta_{S_t}, \mathbb{H}_{S_t} \rangle < \cos\langle \delta_{S_t}^K, \mathbb{H}_{S_t} \rangle$. This conclusion intuitively shows that the norm of H_{S_t} should not exceed $\|\delta_{S_t}\|$, and it is optimal when its direction is opposite to δ_{S_t} . As far as we know, this theorem is the first to provide the effective condition of $\mathbb{H}_{S_t}$, and this result is suitable for papers [1, 11, 10] with a correct term.*

Use ‘Implicit Correct-Term’ to expand the analysis: Theorem 4.3 has strong extensibility and can be used to analyze other methods and properties by replacing correct-direction h_t . For example, training neural networks typically employs a batch manner, where sampling from the data set inevitably leads to inaccurate gradient estimation. For instance, let $\xi_B \sim D_i$, then we have: $\|\nabla f(\mathbf{x}_{t,i}^k; \xi_B) - \nabla f(\mathbf{x}_{t,i}^k; D_i)\|^2 = \varsigma_{t,i}^k$, where $\varsigma_{t,i}^k$ represents the noise introduced due to sampling imprecision. If we regard this noise as h_t^i (although its role differs from ‘correction direction’), the analysis under Theorem 4.3 still holds, and Corollary 4.4 remains applicable.

Proof. Proof idea : Although we have summarized the commonly used federated optimization methods in Algorithm 1, it is impossible for us to analyze all methods in one article. We will use a case study to demonstrate how the conclusions of Theorem 4.1 can be used to simply analyze methods not discussed in this paper and future theoretical approaches.

Through Lemma 4.2, we can know that the pseudo-gradient and the LA-FedAVG algorithm vector are consistent. For the starting operation of LA-FedSGD-Corr (referred to as LAFC for short), it involves substituting the pseudo-gradient using $x_{t+1}^{LA-FedAVG} - x_t = -\mathbb{G}_{S_t}$ to establish a connection with Theorem 4.1. Assume client update as follows: $x_{i,K}^{h,t} = x_t - \eta_l^i \sum_{i=0}^{K_i-1} \nabla f_i + \eta_l^i K_i h_t = x_{i,K}^t + h_t^i$. The update x_{t+1}^{LAFC} of Algorithm 1 is $x_{t+1}^{LAFC} - x_t = -\eta_t [\sum_{i \in S_t} \rho_i (x_t - x_{i,K}^t - h_t^i)] = -\eta_t (\mathbb{G}_{S_t} - \mathbb{H}_{S_t})$ where $\mathbb{G}_{S_t} = \sum_{i \in S_t} \rho_i (x_t - x_{i,K}^t)$ and $\mathbb{H}_{S_t} = \sum_{i \in S_t} \rho_i h_t^i$.

$$x_{t+1}^{LAFC} - x_{S_t}^* = \eta_t (x_{t+1}^{LA-FedAVG} - x_{S_t}^*) + (1 - \eta_t)(x_t - x_{S_t}^*) + \eta_t \mathbb{H}_{S_t}$$

For the equation above, taking the 2-norm on both sides:

$$\begin{aligned}
\|x_{t+1}^{\text{LAFC}} - x_{\mathcal{S}_t}^*\|^2 &= \|\eta_t (x_{t+1}^{\text{LA-FedAVG}} - x_{\mathcal{S}_t}^*) + (1 - \eta_t)(x_t - x_{\mathcal{S}_t}^*) + \eta_t \mathbb{H}_{\mathcal{S}_t}\|^2 \\
&= (2\eta_t^2 - 2\eta_t + 1 + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta) \|\delta_{\mathcal{S}_t}\|^2 - \eta_t^2 \Delta + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + [2\eta_t^2 \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + \\
&\quad \eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle] \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\|.
\end{aligned}$$

By organizing the equations, we can prove the conclusion. $\square$

4.3 Theorem of SA

For SA with any local optimization method, we refer to the form of QHM [32]: $d^t = (1 - \beta_t)\mathbb{G}_{\mathcal{S}_t} + \beta_t d^{t-1}$, $x^{t+1} = x^t - \eta_t [(1 - \nu_t)\mathbb{G}_{\mathcal{S}_t} + \nu_t d^t]$ where the parameter $\nu_t \in [0, 1]$ interpolates between SGD [33] ($\nu_t = 0$) and (normalized) SHB [34] ($\nu_t = 1$). When the parameters η_t , β_t and ν_t are held constant (thus the subscript t can be omitted) and $\nu = \beta$, it recovers a normalized variant of NAG [24] with an additional coefficient $1 - \beta_t$ on the stochastic gradient term. For adaptive learning rate methods (e.g., Adam ($\phi(\mathbb{G}_{\mathcal{S}_t}) = \sqrt{\beta_2 \mathbb{G}_{\mathcal{S}_t}^2 + (1 - \beta_2)v_{t-1}}$) [20], RMSProp [35], Adagrad [36], etc.), we simply abstract them as $\eta_t^\phi = \frac{\eta_t}{\phi(\mathbb{G}_{\mathcal{S}_t})}$.

Theorem 4.5. (SA) Assuming the clients satisfy Assumptions 2.1 and 2.2, and the single-round update distance is:

$$\|x_{t+1} - x_{\mathcal{S}_t}^*\|^2 = \left(1 - \wp_{\nu, \beta}^{\phi, t}\right) \|\delta_{\mathcal{S}_t}^t\|^2 - \eta_t^\phi \left(\eta_t^\phi (1 - \nu_t \beta_t) \Delta_{t+1} + \nu_t \beta_t \bar{\mathfrak{D}}_\phi \|d^{t-1}\| \right), \quad (6)$$

where $\wp_{\nu, \beta}^{\phi, t} = 2\hat{\eta}_{\nu, \beta}^{\phi, t}(1 - \hat{\eta}_{\nu, \beta}^{\phi, t})(1 - \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta)$, $\bar{\mathfrak{D}}_\phi = [\eta_{\nu, \beta}^{\phi, t} \|d^{t-1}\| - 2\bar{h}_\phi \|\delta_{\mathcal{S}_t}\|]$, $\bar{h}_\phi = ((\hat{\eta}_{\nu, \beta}^{\phi, t} - 1) \cos\langle \delta_{\mathcal{S}_t}, d^{t-1} \rangle - \hat{\eta}_{\nu, \beta}^{\phi, t} \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, d^{t-1} \rangle)$ and $\hat{\eta}_{\nu, \beta}^{\phi, t} = \eta_t^\phi (1 - \nu_t \beta_t)$, $\eta_{\nu, \beta}^{\phi, t} = -\eta_t^\phi \nu_t \beta_t$.

Role of $\eta_t^\phi, \nu_t, \beta_t$ These parameters cause an overall decay to the right-hand side of the equation. Focusing on $\eta_t^\phi (1 - \nu_t \beta_t) \Delta_{t+1} + \nu_t \beta_t \bar{\mathfrak{D}}_\phi \|d^{t-1}\|$, under appropriate coefficient conditions, the decrease in distance can be viewed as a weighted sum of Δ_{t+1} and d^{t-1} , where the weights sum approx to 1. When A is indefinite, Δ_{t+1} becomes ineffective, but d^{t-1} maybe can still act as a positive term that brings x_{t+1} closer to $x_{\mathcal{S}_t}^*$. Therefore, when d^{t-1} satisfies the correction term condition, its role is to increase the update distance per step (which manifests as a certain acceleration effect over the entire round) and to correct the trajectory oscillations caused by the action of A . Therefore, when d^{t-1} satisfies the correction term condition 4.4, its role is to increase the update distance per step (which manifests as a certain acceleration effect over the entire round) and to correct the trajectory oscillations caused by the indefinite of A .

A Related Work: Theoretical Perspectives on Gradient Heterogeneity

In federated learning (FL), data heterogeneity is often cited as the reason why standard convergence analyses—based on the local Lipschitz-gradient assumption and a gradient-divergence bound—predict deteriorating rates as the number of local steps grows. Yet empirical evidence shows that more local updates can actually accelerate training even when the product of the Lipschitz constant and divergence is large. To reconcile theory and practice, Wang et al. [37] replace the usual Lipschitz condition with a weaker *heterogeneity-driven pseudo-Lipschitz* assumption. Under this and the standard divergence bound, they derive a tighter upper bound for FedAvg (and its variants), where the large local Lipschitz constant is replaced by a much smaller pseudo-Lipschitz constant, without altering the asymptotic order. It is shown in Fig. 5,6 that the performance of more of the following algorithms deteriorates with increasing heterogeneity. The environment of each of these methods is uniform.

A.1 Adaptive Optimization, Server Momentum, and Sharpness-Aware Methods

Table 2: The function properties only satisfy the convergence conclusion of the latest method analysis of L-smooth. In this table, $\mathcal{F} = F(x_0) - F(x^*)$.

Research	Corollary	Local Learning Rate
Heterogeneity-driven Pseudo-Lipschitz [37]	$\mathcal{O}\left(\sqrt{\frac{\mathcal{F}L\vartheta^2}{TK S }} + \frac{\zeta^2 + \vartheta^2/K}{TKN}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{TKN}}\right)$
FedGM [5]	$\mathcal{O}\left(\sqrt{\frac{K}{Tm}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{TK}}\right)$
FedAVG-M [1]	$\mathcal{O}\left(\sqrt{\frac{L\Delta\vartheta^2}{NKT}} + \frac{L\Delta}{T}\right)$	$\mathcal{O}\left(\frac{1}{\beta K\eta_t L^2 T}\right)$
FEDAS[17]	$\mathcal{O}\left(\frac{\sqrt{\mathcal{F}}\vartheta}{\sqrt{TKM}} + \frac{\sqrt{\mathcal{F}}\zeta}{\sqrt{TM}} + \frac{\mathcal{F}}{T} + \frac{\mathcal{F}G}{T\sqrt{M}} + \frac{\mathcal{F}\tau_{\max}\tau_{\text{avg}}}{T}\right)$	$\mathcal{O}\left(\frac{\sqrt{\mathcal{F}}}{\sqrt{TK(\vartheta^2 + K\zeta^2)}}\right)$
FedAMS[38]	$\mathcal{O}\left(\frac{\sqrt{K}}{\sqrt{T S_t }}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{TK}}\right)$
FedAdam[12]	$\mathcal{O}\left(\frac{\mathcal{F}}{\sqrt{ S KT}} + \frac{2\vartheta^2 L}{G^2\sqrt{ S KT}} + \frac{\vartheta^2}{GKT} + \frac{\vartheta^2 L\sqrt{ S }}{G^2\sqrt{KT^{3/2}}}\right)$	$\mathcal{O}\left(\frac{1}{KL\sqrt{T}}\right)$

Adaptive optimizers and momentum have long improved centralized training, and recently, several works have adapted them to FL. Reddi et al. [12] introduce federated versions of Adagrad, Adam, and Yogi, proving convergence under nonconvex heterogeneity and showing significant empirical gains. Sun et al. [5] propose a unified server-momentum framework that supports stage-wise scheduling and asynchronous clients, with rigorous convergence guarantees in heterogeneous settings. Kim et al. [2] develop Delta-SGD, which auto-tunes each client’s step size via its local smoothness, matching or exceeding centralized baselines without extra tuning. Fan et al. (FedLESAM) [39] argue that local SAM perturbations misalign with global sharpness; instead, they estimate the global perturbation by differencing successive global models, achieving tighter bounds and faster convergence in practice.

A.2 Regularization and Drift-Correction Methods

In convex FL, regularization and control variates correct client drift and improve communication–computation trade-offs. SCAFFOLD [10] uses client-side control variates to eliminate drift, reducing communication rounds and leveraging data similarity for quadratic objectives. DANE, DANE+ & FedRed [3] revisit the proximal-point method: DANE enjoys communication reduction under Hessian similarity; DANE+ and FedRed introduce a doubly-regularized drift-correction scheme that relaxes local solver accuracy while preserving communication complexity. ProxSkip [11] skips expensive proximal updates in most iterations, cutting prox calls from $\mathcal{O}(\kappa \log \frac{1}{\epsilon})$ to $\mathcal{O}(\sqrt{\kappa} \log \frac{1}{\epsilon})$ while keeping overall iteration complexity, outperforming FedAvg and SCAFFOLD under heterogeneity. S-DANE [40] augments DANE with auxiliary prox-centers to further relax local accuracy requirements, supporting stochastic solvers and partial participation, and admits adaptive line-search variants. FedAVGM [1] integrates server momentum into FedAvg and SCAFFOLD, proving FedAvg converges with a constant learning rate and no bounded-heterogeneity assumption, and accelerating SCAFFOLD under partial participation, with momentum-based variance-reduction extensions.

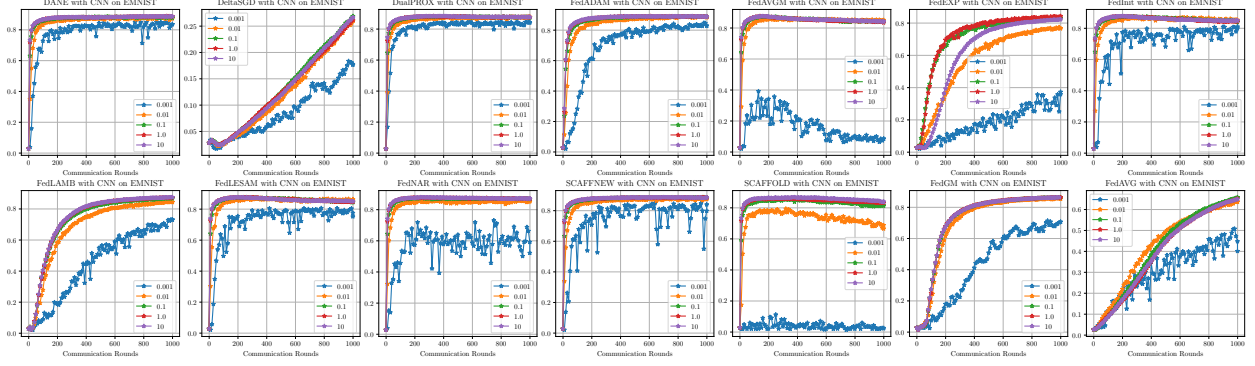


Figure 5: The performance of FedAVG, FedAVGM, DeltaSGD, DualPROX, FedEXP, FedGM, FedInit, FedLESAM, FedNAR, FedPROX, SCAFFOLD, SCAFFNEW, and FedADAM on the EMNIST classification task.

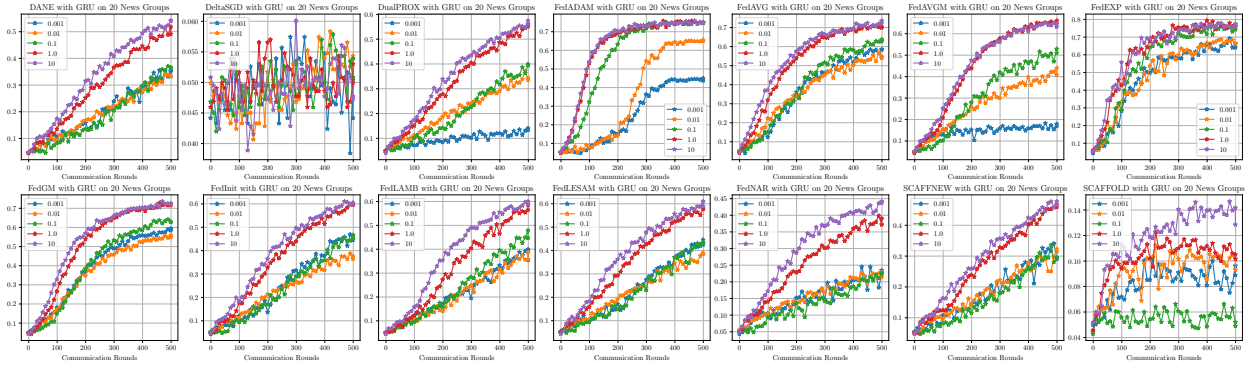


Figure 6: The performance of FedAVG, FedAVGM, DeltaSGD, DualPROX, FedEXP, FedGM, FedInit, FedLESAM, FedNAR, FedPROX, SCAFFOLD, SCAFFNEW, and FedADAM on the 20 News Groups classification task.

B Visualization tools

We refer to the method [27] of visualizing loss landscapes, which has the advantage of being able to simultaneously visualize the relative regions of multiple neural networks. In this paper, our drawing includes the approximate positions of local optima in client-side neural networks and the trajectory of the client-side loss function. The method first initializes two "base models" W_x, W_y using a Gaussian distribution as the basis for projection. This paper's visualization is divided into two cases:

Relative Position Visualization Method (Fig. 2 right) : For locating model $W_{i,t}$ in 2D coordinates x_i, y_i . Extract its classification layer weights $W_{i,t}^c$, and extract the classification layer weights of the base models W_x^c and W_y^c . The coordinates are determined by projecting $W_{i,t}^c$ onto W_x^c and W_y^c , divided by the norm of the basis vectors:

$$x = \frac{\|W_{i,t}^c \cos(\langle W_{i,t}^c, W_x^c \rangle)\|}{\|W_x^c\|}, \quad y = \frac{\|W_{i,t}^c \cos(\langle W_{i,t}^c, W_y^c \rangle)\|}{\|W_y^c\|} \quad (7)$$

In Fig. 7, we plotted the relative positions of optimal points and x_t across different tasks under varying communication rounds. Since the entire process shares a pair of common base models, the trajectory of x_t can be traced. The results in Fig. 7 reveal that in more non-i.i.d. scenarios, the positions of the optimal points become increasingly dispersed.

Loss Landscape Visualization Method (Fig. 2 left): All client-side models share the same set of base models W_x, W_y . For a model $W_{i,t}$ to be visualized on dataset D_i , we plot the function

$$f(\sigma, \beta; x_i, y_i, D_i) = f(W_{i,t} + (\sigma - x_i)W_x + (\beta - y_i)W_y; D_i)$$

where x_i, y_i are the relative positions of $W_{i,t}$ calculated using Eq. 7 and $\sigma, \beta \in \mathbb{R}$. Here, the addition and multiplication operations are applied to all parameters of the neural network.

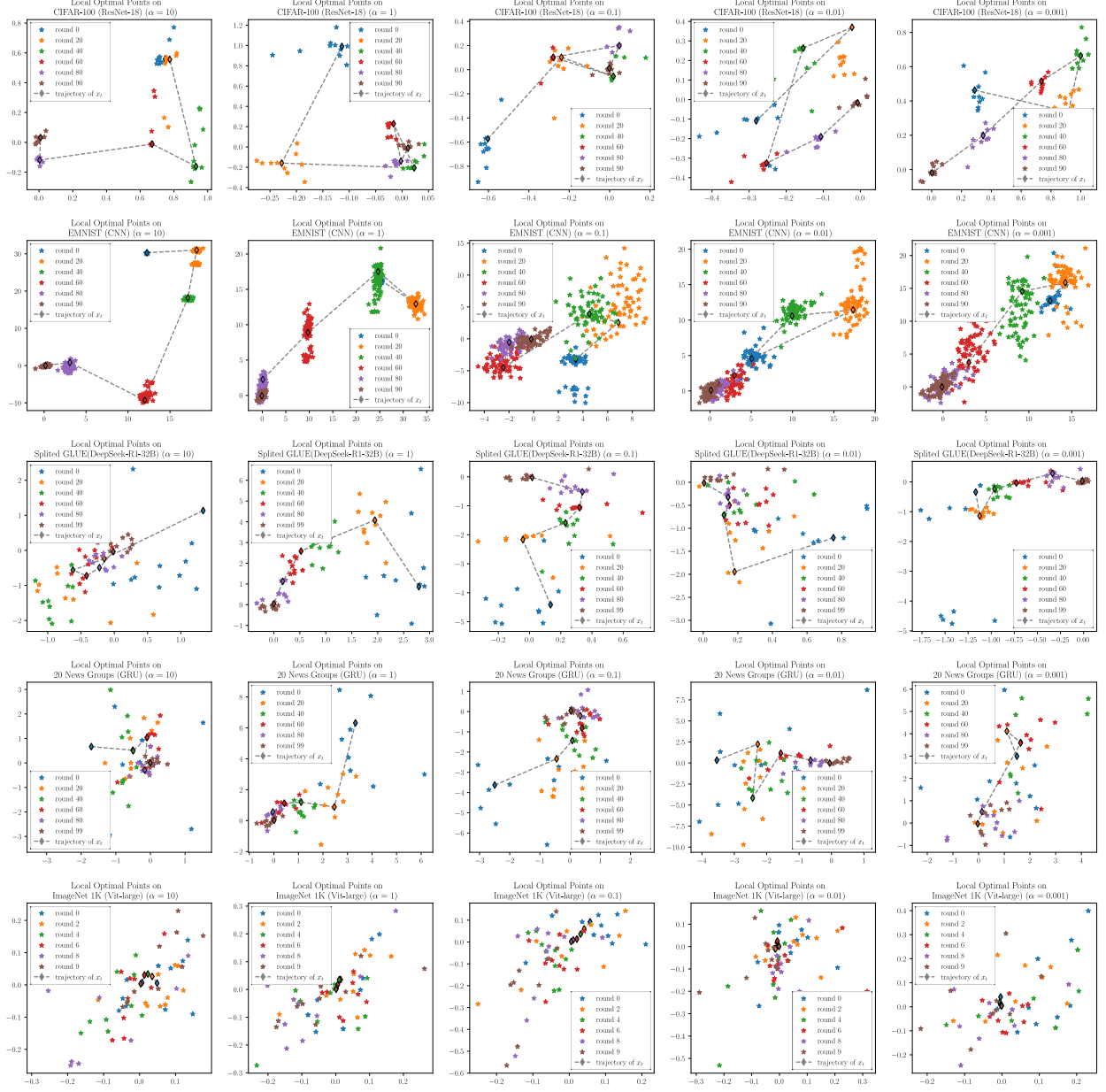


Figure 7: Visualization of the positional relationships between the optimization trajectory of x_t under multiple tasks and the local optimal points of different clients at time t .

Gather Local Optima Visualization To gather all the functions near the client’s optimal point on a single graph, we use the following form: $g(\sigma, \beta) = \min(f(\sigma, \beta; \mathcal{D}_1), f(\sigma, \beta; \mathcal{D}_2), f(\sigma, \beta; \mathcal{D}...))$ In Fig 2,8 we use this expression.

C Experiment Detail

C.1 Simulating the Experimental Setup for Heterogeneous

C.1.1 Fixed Optimal Points Simulation: Paraboloid

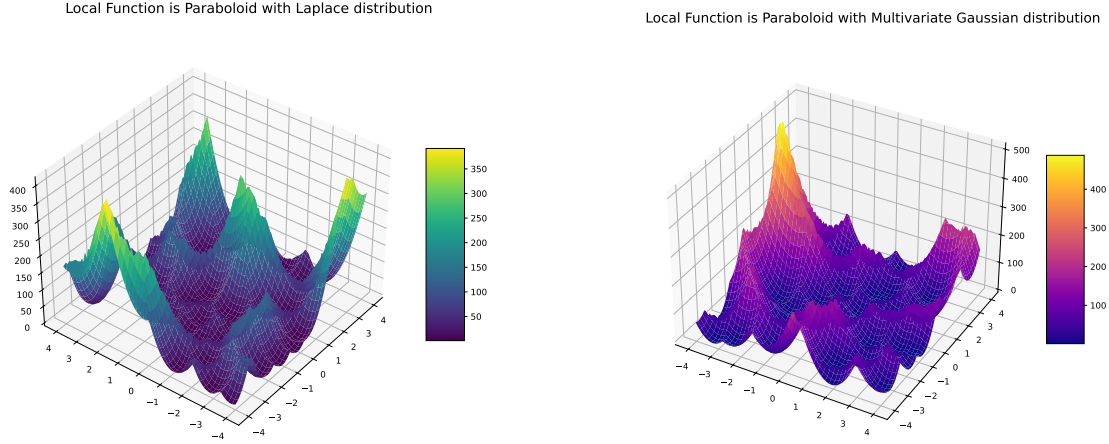


Figure 8: When f_i is a parabolic surface, the merging situation near the optimal point on the client side. We use $g(x) = \min(f_1(x), f_2(x), f_{...}(x))$ to display the optimal points for each client.

Since the optimal point of the neural network tends to change with the change of x_t , in the simple verification of the existence of the oscillating region (Fig. 4), we use the fixed optimal point of the parabola to verify the oscillating region:

$$f(x) = \frac{1}{M} \sum_{i=0}^M (x - \delta_i)^2$$

where δ_i samples from the multivariate Gaussian and Laplace distributions. In Fig. 8 we plot the case of a localized image of $(x - \delta_i)^2$.

C.2 Various neural network tasks

C.2.1 Methods for Data Splitting and Allocation

We utilize the Dirichlet distribution to randomly allocate data across various classes detailed in Alg. 2, thereby achieving a non-i.i.d. partitioning of data among multiple clients. First, for each class, the algorithm gathers the indices of all samples belonging to that class. Then, for each class, a probability vector is drawn from the Dirichlet distribution, reflecting the distribution proportion of data across the clients for that class. Next, by multiplying the sampled probability vector with the total number of remaining unassigned samples in the current class and taking the ceiling of the result, the number of samples to be assigned to each client for that class is obtained. To ensure randomness and fairness in the assignment process, the order in which the clients are processed is randomly permuted in each iteration.

After the data for all classes has been allocated, the algorithm performs a post-processing step to ensure that each client has a sufficient number of samples: if a client’s sample count falls below a preset minimum threshold, some samples are transferred from the client with the most samples to meet the requirement. Finally, the algorithm outputs the set of data indices corresponding to each client, achieving a Dirichlet distribution-based non-i.i.d. data partitioning that effectively simulates the uneven distribution of data among clients in real-world scenarios.

Algorithm 2 Non-i.i.d Data Splitting via Dirichlet Allocation

Input: Source data with labels y , number of clients C , Dirichlet parameter α . Data index allocation $\{D_1, D_2, \dots, D_C\}$ for each client.

Partition: For each class c , let $I_c \leftarrow \{i \mid y_i = c\}$.

```

1: for classes  $c$  do
2:   Sample a probability vector  $p^c \sim \text{Dirichlet}(\sigma, \dots, \sigma)$  over  $C$  clients.
3: end for
4: while there exists a class  $c$  with  $|I_c| > 0$  do
5:   Shuffle the client order:  $\pi \leftarrow$  random permutation of  $\{1, \dots, C\}$ .
6:   for classes  $c$  do
7:     Let  $n_c \leftarrow |I_c|$ .
8:     Compute allocation counts:  $d^c \leftarrow \lceil p^c \cdot n_c \rceil$ .
9:     for clients  $i \in \pi$  do
10:      Assign the first  $d_i^c$  indices from  $I_c$  to client  $i$ ,
11:      i.e., update  $D_i \leftarrow D_i \cup$  first  $d_i^c$  elements of  $I_c$ .
12:      Remove these indices from  $I_c$ .
13:    end for
14:   end for
15: end while
16: for clients  $i$  do
17:   if  $|D_i|$  is below a minimal threshold then
18:     Transfer a few samples from the client with the most data.
19:   end if
20: end for
21: return  $\{D_1, D_2, \dots, D_C\}$ .

```

C.2.2 GRU on 20 News Groups

The 20 Newsgroups[41] dataset is a widely used and well-known text classification dataset that contains articles from 20 different newsgroups, covering a wide variety of topics and categories. Our goal is to analyze the content features of each article and accurately predict its specific newsgroup category.

In the data processing phase for the 20 Newsgroups dataset, text sequences are standardized to a fixed length of 128 tokens. For samples exceeding this length, a random truncation strategy is implemented: when the remaining sequence length after truncation exceeds 128 tokens, a random starting index between 0 and the remaining length is selected to preserve semantically critical segments. Shorter sequences are padded with zeros at the beginning to align all inputs to a fixed 128×1 tensor dimension. This approach enhances data randomness while maintaining batch processing efficiency, with zero-padded positions masked via the embedding layer’s padding identifier (padding_idx=0).

The network architecture employs a two-layer unidirectional GRU structure. The embedding layer maps a 128,000-dimensional discrete vocabulary into a 128-dimensional continuous vector space, followed by layer normalization to standardize the 128×128 sequence features. Each recurrent layer contains 128 hidden units, capturing temporal dependencies through time-step unfolding. The final hidden state at the 128th time step serves as the global feature representation. A fully connected layer projects the 128-dimensional hidden state into a 20-dimensional classification space, corresponding to the 20 newsgroup categories. The design utilizes parameter-sharing mechanisms to control complexity while maintaining temporal modeling capabilities. Layer normalization mitigates gradient vanishing issues, enhancing robustness for long-text processing.

C.2.3 Resnet-18 on CIFAR 100

The CIFAR-100[42] is a dataset for image classification tasks and is part of the CIFAR series of datasets. FedOPT employs a two-step LDA process for both coarse and fine labels. We conduct training on CIFAR-100 using a modified ResNet-18 like FedOPT [12], wherein the batch normalization layers have been substituted with group normalization layers. Specifically, each group normalization layer is configured with two groups. The preprocessing for CIFAR-100 involves images with 3 channels of 32×32 pixels each, where each pixel is represented by an unsigned int8. We conduct preprocessing on both training and test images. For training images, random cropping is applied to reshape them to (24, 24, 3), followed by a random horizontal flip. For testing images, a central crop is performed to adjust the shape to (24, 24, 3).

C.2.4 ViT-base for ImageNet 1k

This experiment leverages a pre-trained Vision Transformer (ViT) [29] model to validate the effectiveness and linear separability of ViT representations in downstream classification tasks on the ImageNet-1K [43] dataset. Key components include:

This experimental workflow consists of three core components. The dataset preparation phase utilizes the ImageNet-1K benchmark, with 1,281,167 training images and 50,000 validation images across 1,000 classes. Images undergo preprocessing through the ViTImageProcessor to meet the pretrained model specifications, including resizing to the required resolution, pixel value normalization to the $[-1,1]$ range, and conversion into PyTorch tensors.

For feature extraction, the pretrained CLIP-ViT-Large-Patch14 model – a Vision Transformer with a 14x14 patch division strategy – processes images in batches of 256 without gradient computation. The 768-dimensional [CLS] token embeddings from the final transformer layer (captured via `last_hidden_state[:,0,:]`) are aggregated into (N,768) feature matrices for both training and validation sets, accompanied by corresponding label vectors. The classification architecture employs a two-layer MLP implemented as `torch.nn.Sequential`: a 768→2048 linear projection followed by Tanh activation, then a 2048→1000 linear layer mapping to class logits. This classifier operates with frozen visual backbone parameters, focusing optimization exclusively on the final classification layer during fine-tuning.

C.2.5 Deepseek-32b on Split GLUE

In this experiment, we adopted multiple tasks from the GLUE[44] dataset and unified them into a 23-class multi-class classification task. Specifically, for each task (including CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, RTE, and WNLI), we first defined the corresponding text fields and label fields based on the task configuration. For tasks with two text fields (e.g., MRPC, STS-B, MNLI, etc.), we concatenated the texts in the fields into a single input using predefined separators. Notably, for the STS-B task, since the original labels were continuous values, we converted them to integers via rounding to facilitate subsequent classification processing.

To ensure that labels from different tasks do not overlap with each other, we implemented a label offset (offset) strategy. Specifically, when processing each task, we added an accumulated offset to its label values, ensuring that the label distributions across tasks were non-overlapping and enabling the division of 23 distinct categories. Finally, we stored the processed training and validation sets from all tasks in a single file for subsequent model training and evaluation.

DeepSeek-R1 [30] is an advanced language model combining reinforcement learning (RL) and distillation to improve reasoning quality over predecessors. It uses two-stage RL (pattern exploration + human preference alignment) with supervised fine-tuning (SFT) and distills complex reasoning into a 32B-parameter Qwen-based architecture. The model excels in math, coding, and logic tasks, outperforms peers like GPT-4-mini in key areas, and supports advanced reasoning features (chain-of-thought, self-verification). Open-sourced for community-driven optimization and lightweight model distillation.

Using DeepSeek-R1-Distill-Qwen-32B as a feature extractor, we encode text through its pre-trained model to generate 5120-dimensional text representation vectors as input. The classification model employs a simplified multilayer perceptron (MLP) structure: the input layer projects 5120-dimensional features to a 1024-dimensional hidden layer via linear transformation, enhances non-linear expressive capacity through ReLU activation, and then directly outputs 23-dimensional logits (corresponding to 23 classes) via a second linear transformation.

C.3 Total experimental environment setup

Table 3: Cross-device Configurations in Federated Learning Experimental Setups

Model	Clients	Epochs	η	Batch Size	Comm. Rounds	Participants	Server LR
CNN on EMNIST	620	1	0.10	2048	1000	62	0.001
GRU on 20 News Groups	100	1	0.001	512	500	10	0.010
ResNet-18 on CIFAR100	1000	5	0.001	128	4000	10	0.001
ViT-Large on ImageNet-1K	10000	5	0.001	768	1000	10	0.001
DeepSeek-R1-Distill-Qwen-32B on GLUE	100	1	0.001	256	500	10	0.001

This paper focuses on the application of Federated Learning in a cross-device scenario [45], where clients such as smartphones [46] are reluctant to upload local private data, while FL enables clients to participate in the training process by transmitting locally trained models. The experiments specify the experimental conditions for Fig. 1, 5, and 6 in Tab. 3, which align with the cross-device scenario. Here, the Server Learning Rate (Server LR η) corresponds to the η parameter in the FedAdam algorithm. In the last few dozen rounds of the communication in Fig. 1, we will gradually

reduce η_l to converge to a local optimum. Our experiments were conducted on a system running Ubuntu 24.04.2 LTS, equipped with 8 NVIDIA RTX 3090 GPUs and 512GB of memory.

D Proofs

D.1 Theorem and Corollary of LA-FedAVG

Theorem D.1. (LA-FedAVG) Assuming that local optimal points satisfy Assumption 2.1 and all clients $i \in \mathcal{S}_t$ satisfy the effective descent condition 2.2. The single update distance from x_{t+1} of LA-FedAVG to the sampled local optimum $x_{\mathcal{S}_t}^* = \sum_{i \in \mathcal{S}_t} \rho_i x_i^*$ is:

$$\|x_{t+1} - x_{\mathcal{S}_t}^*\|^2 = \|x_t - x_{\mathcal{S}_t}^*\|^2 - \frac{1}{|\mathcal{S}_t|^2} (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t})^\top A (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t}), \quad (8)$$

where $\mathcal{P}_{\mathcal{S}_t} = [\rho_i^t]_{i \in \mathcal{S}_t}$ is weight vector, $X_{\mathcal{S}_t} = [\delta_{t,i}^K]_{i \in \mathcal{S}_t}$, the element of matrix A is $A_{i,j} = \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle$.

Proof. With any $\sum_{i \in \mathcal{S}_t} \rho_i^t = 1, \forall \rho_i^t > 0, x_{t,F}^* = \sum_{i \in \mathcal{S}_t} \rho_i^t x_i^*$

$$\begin{aligned} \|x_{t+1} - x_{t,F}^*\|^2 &= \left\| \sum_{i \in \mathcal{S}_t} \rho_i^t (x_{K,i}^t - x_i^*) \right\|^2 = \left\| \sum_{i \in \mathcal{S}_t} \rho_i^t \delta_{t,i}^K \right\|^2 = \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \langle \rho_i^t \delta_{t,i}^K, \rho_j^t \delta_{t,j}^K \rangle \right] \\ &= \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \|\rho_i^t \delta_{t,i}^K\| \|\rho_j^t \delta_{t,j}^K\| \cos \langle \rho_i^t \delta_{t,i}^K, \rho_j^t \delta_{t,j}^K \rangle \right] \end{aligned}$$

For the second inequality, we let $\delta_{t,i}^K = x_{K,i}^t - x_i^*$. The second last inequality uses the vector cosine similarity formula: the cosine of the angle between vectors $\mathbf{a}$ and $\mathbf{b}$ is given by their dot product divided by the product of their magnitudes.

$$\langle \mathbf{a} \cdot \mathbf{b} \rangle = \|\mathbf{a}\| \|\mathbf{b}\| \cos \langle \mathbf{a} \cdot \mathbf{b} \rangle$$

Similarly, we can express the distance between x_t and $x_{t,F}^*$ in the following form:

$$\|x_t - x_{t,F}^*\|^2 = \left\| \sum_{i \in \mathcal{S}_t} \rho_i^t (x_t - x_i^*) \right\|^2 = \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \|\rho_i^t \delta_{t,i}\| \|\rho_j^t \delta_{t,j}\| \cos \langle \rho_i^t \delta_{t,i}, \rho_j^t \delta_{t,j} \rangle \right]$$

Let $\delta_{t,i} = x_t^t - x_i^*$. We need to determine under what conditions x_{t+1} will converge toward $x_{t,F}^*$, so by subtracting the above two equations, we obtain:

$$\begin{aligned} &\|x_{t+1} - x_{t,F}^*\|^2 - \|x_t - x_{t,F}^*\|^2 \\ &= \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} (\|\rho_i^t \delta_{t,i}^K\| \|\rho_j^t \delta_{t,j}^K\| \cos \langle \rho_i^t \delta_{t,i}^K, \rho_j^t \delta_{t,j}^K \rangle - \|\rho_i^t \delta_{t,i}\| \|\rho_j^t \delta_{t,j}\| \cos \langle \rho_i^t \delta_{t,i}, \rho_j^t \delta_{t,j} \rangle) \right] \\ &= \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| (\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle) \right] \end{aligned}$$

□

Lemma D.2. Decoupling [47, 31, 48] is a technique of replacing quadratic forms of random variables by bilinear forms. Let A be an $n \times n$ matrix with zero diagonal. Let $X = (X_1, \dots, X_n)$ be a random matrix.

$$\mathbb{E} \sum_{i \in [n]} \sum_{j \in [n]} a_{ij} \langle X_i, X_j \rangle = 4 \mathbb{E} \sum_{i \in I} \sum_{\substack{j \in I^c \\ j \neq i}} a_{ij} \langle X_i, X_j' \rangle \quad (9)$$

where X' is an independent copy of X , $I := \{i : \delta_i = 1\}$ and $I^c := [n]/I$.

Proof. We adopted some proof ideas from the research to address the chaos in quadratic forms. Study point [47, 31, 48] that they replace the chaos $\sum_{i,j} a_{ij} \langle X_i, X_j \rangle$ by the “partial chaos”

$$\sum_{(i,j) \in I \times I^c} a_{ij} \langle X_i, X_j \rangle$$

where the subset of indices $I \subset \{1, \dots, n\}$ will be chosen by random sampling. The advantage of partial chaos is that the summation is done over disjoint sets for i and j . Thus one can automatically replace X_j by X'_j without changing the distribution. Finally, the study completes the partial chaos to the full sum $\sum_{i,j} a_{ij} \langle X_i, X'_j \rangle$.

To randomly select a subset of indices I , let us consider selectors of column $\delta_1, \dots, \delta_n \in \{0, 1\}$, which are independent Bernoulli random variables with $\mathbb{P}\{\delta_i = 0\} = \mathbb{P}\{\delta_i = 1\} = 1/2$. Define

$$I := \{i : \delta_i = 1\}.$$

Condition on X . Since by assumption $a_{ii} = 0$ and

$$\mathbb{E}\delta_i(1 - \delta_j) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} \quad \text{for all } i \neq j,$$

we may express the chaos as

$$\sum_{i \in [n]} \sum_{\substack{j \in [n] \\ i \neq j}} a_{ij} \langle X_i, X_j \rangle = 4\mathbb{E}_\delta \sum_{i \neq j} \delta_i(1 - \delta_j) a_{ij} \langle X_i, X_j \rangle = 4\mathbb{E}_I \sum_{i \in I} \sum_{i \in I^c} a_{ij} \langle X_i, X_j \rangle.$$

Since X_i , $i \in I$ are independent from X_j , $j \in I^c$, the distribution of this sum will not change if we replace X_j by X'_j , the coordinates of X' , and thus we can obtain Lemma 9. □

Corollary D.3. *Under the conditions outlined in Theorem 4.1, assuming that $x_{i,t}^K$ is uniformly distributed within a norm ball centered at x_i^* with radius $\|\delta_i^t\|$, and given that $\min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle > 0$ (which implies that the matrix A is positive definite), through decoupling Lemma 9, we can determine the range of the descent distance $\Delta_{t+1} = \frac{1}{|\mathcal{S}_t|^2} (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t})^\top A (\mathcal{P}_{\mathcal{S}_t} \cdot X_{\mathcal{S}_t})$ in the LA-FedAVG algorithm as follows:*

$$\rho_{\min,t}^2 \mathcal{U}_{\min} \left\| \delta_{\min}^{t,i} \right\|^2 \leq \mathbb{E} [\Delta_{t+1}] \leq \rho_{\max,t}^2 \mathcal{U}_{\max} \left\| \delta_{\max}^{t,i} \right\|^2, \quad (10)$$

where $\mathcal{U}_{\max} = \frac{1}{|\mathcal{S}_t|} (1 - \sigma_{\min,t}^2) + \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle$ and $\mathcal{U}_{\min} = \frac{1}{|\mathcal{S}_t|} (1 - \sigma_{\max,t}^2) + 4\epsilon(1 - \epsilon) \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle$ and $\epsilon \in (0, 1)$ is a constant due to the sampling bias.

Proof.

$$\begin{aligned} \Delta_{t+1} &= \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) \right] \\ &= \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{\substack{j \in \mathcal{S}_t \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right] \\ &\leq \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{\substack{j \in \mathcal{S}_t \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right] \end{aligned}$$

To analyze the expectation of the first term within the square brackets, we need to decouple $\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle$. By decoupling 9 the cosine values of pairs of vectors, they become independent, which facilitates the calculation of the expectation for each term in the summation.

$$\begin{aligned}
& \mathbb{E} [\Delta_{t+1}] \\
& \leq \frac{1}{|\mathcal{S}_t|^2} \left[4 \mathbb{E} \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta'_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta'_{t,j,K} \rangle - \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right] \\
& = \frac{1}{|\mathcal{S}_t|^2} \left[4 \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta'_{t,j}\| \left(\sigma_i^t \sigma_j^t \mathbb{E} [\cos \langle \delta_{t,i}^K, \delta'_{t,j,K} \rangle] - \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right] \\
& = \frac{1}{|\mathcal{S}_t|^2} \left[4 \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta'_{t,j}\| \left(- \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right]
\end{aligned}$$

where $I := \{i : \delta_i = 1\}$, $I^c := \mathcal{S}_t / I$, and $\delta'_{t,j}, \delta'_{t,j,K}$ denote independent copies of $\delta_{t,j}$ and $\delta_{t,j,K}$. Given that the inner product terms $\delta_{t,i}^K, \delta'_{t,j,K}$ are independent and assuming that the angles between vectors are uniformly distributed, since the cosine of these angles is symmetric over the interval $[0, \pi]$, their integral—and thus their expectation—is zero, $\mathbb{E} [\cos \langle \delta_{t,i}^K, \delta'_{t,j,K} \rangle] = \int_0^\pi \cos \theta d\theta = 0$.

Under the condition that client update directions are independent and their angles are uniformly distributed, the single-update absolute descent in federated learning $\mathbb{E}[\Delta_{t+1}] < 0$ is guaranteed if and only if $\min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle > 0$.

If $\min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle > 0$, we got :

$$\begin{aligned}
\mathbb{E} [\Delta_{t+1}] & \leq \frac{\min_{i \in \mathcal{S}_t} \|\delta_{t,i}\|^2}{|\mathcal{S}_t|^2} \left[4 \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \left(- \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \right] \\
& \leq \frac{\min_{i \in \mathcal{S}_t} \|\delta_{t,i}\|^2}{|\mathcal{S}_t|^2} \left[-4II^c \rho_{t,min}^2 \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle + |\mathcal{S}_t| \rho_{min,t}^2 (\sigma_{max,t}^2 - 1) \right] \\
& \leq -\frac{\rho_{min,t}^2}{|\mathcal{S}_t|} \left[1 + 4 \frac{II^c}{|\mathcal{S}_t|} \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle - \sigma_{max,t}^2 \right] \times \|\delta_{min}^{t,i}\|^2 \\
& \leq -\rho_{min,t}^2 \left[\frac{1}{|\mathcal{S}_t|} (1 - \sigma_{max,t}^2) + 4\epsilon(1 - \epsilon) \min_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right] \times \|\delta_{min}^{t,i}\|^2
\end{aligned}$$

where $\epsilon = I / |\mathcal{S}_t|$.

Next we proof inf of Δ_{t+1}

$$\begin{aligned}
\Delta_{t+1} & = \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{j \in \mathcal{S}_t} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) \right] \\
& = \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{\substack{j \in \mathcal{S}_t \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right] \\
& \geq \frac{1}{|\mathcal{S}_t|^2} \left[\sum_{i \in \mathcal{S}_t} \sum_{\substack{j \in \mathcal{S}_t \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(\sigma_i^t \sigma_j^t \cos \langle \delta_{t,i}^K, \delta_{t,j}^K \rangle - \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right]
\end{aligned}$$

$$\mathbb{E}[\Delta_{t+1}] \geq \frac{1}{|\mathcal{S}_t|^2} \left[4 \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \|\delta_{t,i}\| \|\delta_{t,j}\| \left(-\max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \|\delta_{t,i}\|^2 \right]$$

If $\max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle > 0$, we get :

$$\begin{aligned} \mathbb{E}[\Delta_{t+1}] &\geq \frac{\max_{i \in \mathcal{S}_t} \|\delta_{t,i}\|}{|\mathcal{S}_t|^2} \left[4 \sum_{i \in I} \sum_{\substack{j \in I^c \\ i \neq j}} \rho_i^t \rho_j^t \left(-\max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right) + \sum_{i \in \mathcal{S}_t} \rho_{i,t}^2 (\sigma_{i,t}^2 - 1) \right] \\ &\geq \frac{\rho_{\max,t}^2}{|\mathcal{S}_t|^2} \left[-4II^c \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle + \sum_{i \in \mathcal{S}_t} (\sigma_{\min,t}^2 - 1) \right] \times \|\delta_{\max}^{t,i}\|^2 \\ &\geq \frac{\rho_{\max,t}^2}{|\mathcal{S}_t|^2} \left[-4II^c \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle + \sum_{i \in \mathcal{S}_t} (\sigma_{\min,t}^2 - 1) \right] \times \|\delta_{\max}^{t,i}\|^2 \\ &\geq -\rho_{\max,t}^2 \left[\frac{1}{|\mathcal{S}_t|} (1 - \sigma_{\min,t}^2) + 4\epsilon(1 - \epsilon) \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right] \times \|\delta_{\max}^{t,i}\|^2 \\ &\geq -\rho_{\max,t}^2 \left[\frac{1}{|\mathcal{S}_t|} (1 - \sigma_{\min,t}^2) + \max_{i,j \in \mathcal{S}_t} \cos \langle \delta_{t,i}, \delta_{t,j} \rangle \right] \times \|\delta_{\max}^{t,i}\|^2 \end{aligned}$$

where last ineq because $4\epsilon(1 - \epsilon) < 1$. $\square$

D.2 Theorem of LA-FedSGD-Corr

Theorem D.4. (LA-FedSGD-Corr) Assuming the clients satisfy Assumptions 2.1 and 2.2, and the updating model takes the form of $x_{i,K}^{h,t} = x_t - \eta_l^i \sum_{i=0}^{K_i-1} \nabla f_i + \eta_l^i K_i h_t = x_{i,K}^t + h_t^i$. Let $\delta_{\mathcal{S}_t}^K = x_{t+1} - x_{\mathcal{S}_t}^*$ and $\delta_{\mathcal{S}_t} = x_t - x_{\mathcal{S}_t}^*$, the single-round update distance is:

$$\|x_{t+1} - x_{\mathcal{S}_t}^*\|^2 = (1 - \eta_t \wp) \|x_t - x_{\mathcal{S}_t}^*\|^2 - \eta_t (\eta_t \Delta_{t+1} + \bar{\mathfrak{O}} \|\mathbb{H}_{\mathcal{S}_t}\|), \quad (11)$$

where $\wp = 2(1 - \eta_t)(1 - \cos \langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_{\Delta})$, $\bar{\mathfrak{O}} = [2\hbar \|\delta_{\mathcal{S}_t}\| - \eta_t \|\mathbb{H}_{\mathcal{S}_t}\|]$, $\hbar = -\eta_t \sigma_{\Delta} \cos \langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + (\eta_t - 1) \cos \langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle$ and $\sigma_{\Delta_{t+1}} = \sqrt{\|\delta_{\mathcal{S}_t}\| - \Delta_{t+1} / \|\delta_{\mathcal{S}_t}\|}$, $\mathbb{H}_{\mathcal{S}_t} = \sum_{i \in \mathcal{S}_t} \rho_i h_t^i$.

Proof. Through Lemma 4.2, we can know that the pseudo-gradient and LA-FedAVG algorithm vector are consistent. For the starting operation of LA-FedSGD-Corr (referred to as LAFC for short), it involves substituting the pseudo-gradient using $x_{t+1}^{\text{LA-FedAVG}} - x_t = -\mathbb{G}_{\mathcal{S}_t}$ to establish a connection with Theorem 4.1.

Assume client update as follow :

$$x_{i,K}^{h,t} = x_t - \eta_l^i \sum_{i=0}^{K_i-1} \nabla f_i + \eta_l^i K_i h_t = x_{i,K}^t + h_t^i$$

The update x_{t+1}^{LAFC} of 1 is

$$x_{t+1}^{\text{LAFC}} - x_t = -\eta_t \left[\sum_{i \in \mathcal{S}_t} \rho_i (x_t - x_{i,K}^t - h_t^i) \right] = -\eta_t (\mathbb{G}_{\mathcal{S}_t} - \mathbb{H}_{\mathcal{S}_t})$$

where $\mathbb{G}_{\mathcal{S}_t} = \sum_{i \in \mathcal{S}_t} \rho_i (x_t - x_{i,K}^t)$ and $\mathbb{H}_{\mathcal{S}_t} = \sum_{i \in \mathcal{S}_t} \rho_i h_t^i$.

$$\begin{aligned} x_{t+1}^{\text{LAFC}} - x_t &= -\eta_t (\mathbb{G}_{\mathcal{S}_t} - \mathbb{H}_{\mathcal{S}_t}) \\ x_{t+1}^{\text{LAFC}} - x_{\mathcal{S}_t}^* - (x_t - x_{\mathcal{S}_t}^*) &= \eta_t (x_{t+1}^{\text{LA-FedAVG}} - x_t) + \eta_t \mathbb{H}_{\mathcal{S}_t} \\ &= \eta_t (x_{t+1}^{\text{LA-FedAVG}} - x_{\mathcal{S}_t}^* - (x_t - x_{\mathcal{S}_t}^*)) + \eta_t \mathbb{H}_{\mathcal{S}_t} \\ x_{t+1}^{\text{LAFC}} - x_{\mathcal{S}_t}^* &= \eta_t (x_{t+1}^{\text{LA-FedAVG}} - x_{\mathcal{S}_t}^*) + (1 - \eta_t)(x_t - x_{\mathcal{S}_t}^*) + \eta_t \mathbb{H}_{\mathcal{S}_t} \end{aligned}$$

$$\begin{aligned}
\|x_{t+1}^{\text{LAFC}} - x_{\mathcal{S}_t}^*\|^2 &= \|\eta_t (x_{t+1}^{\text{LA-FedAVG}} - x_{\mathcal{S}_t}^*) + (1 - \eta_t)(x_t - x_{\mathcal{S}_t}^*) + \eta_t \mathbb{H}_{\mathcal{S}_t}\|^2 \\
&= \eta_t^2 \|\delta_{\mathcal{S}_t}^K\|^2 + (1 - \eta_t)^2 \|\delta_{\mathcal{S}_t}\|^2 + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + 2\eta_t(1 - \eta_t) \|\delta_{\mathcal{S}_t}^K\| \|\delta_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \\
&\quad + 2\eta_t^2 \|\delta_{\mathcal{S}_t}^K\| \|\mathbb{H}_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle \\
&\quad + 2\eta_t(1 - \eta_t) \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle \\
&= \eta_t^2 \left(\|\delta_{\mathcal{S}_t}\|^2 - \Delta \right) + (1 - \eta_t)^2 \|\delta_{\mathcal{S}_t}\|^2 + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + 2\eta_t(1 - \eta_t) \sqrt{\|\delta_{\mathcal{S}_t}\|^2 - \Delta} \|\delta_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \\
&\quad + 2\eta_t^2 \sqrt{\|\delta_{\mathcal{S}_t}\|^2 - \Delta} \|\mathbb{H}_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle \\
&\quad + 2\eta_t(1 - \eta_t) \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\| \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle \\
&= (2\eta_t^2 - 2\eta_t + 1) \|\delta_{\mathcal{S}_t}\|^2 - \eta_t^2 \Delta + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta \|\delta_{\mathcal{S}_t}\|^2 \\
&\quad + [2\eta_t^2 \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle] \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\| \\
&= (2\eta_t^2 - 2\eta_t + 1 + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta) \|\delta_{\mathcal{S}_t}\|^2 - \eta_t^2 \Delta + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + [2\eta_t^2 \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle] \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\|
\end{aligned}$$

where $\delta_{\mathcal{S}_t}^K = x_{t+1}^{\text{LA-FedAVG}} - x_{\mathcal{S}_t}^*$, $\delta_{\mathcal{S}_t} = x_t - x_{\mathcal{S}_t}^*$

$$\begin{aligned}
&\|x_{t+1}^{\text{LAFC}} - x_{\mathcal{S}_t}^*\|^2 - \|\delta_{\mathcal{S}_t}\|^2 \\
&= 2\eta_t(\eta_t - 1) (1 - \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta) \|\delta_{\mathcal{S}_t}\|^2 - \eta_t^2 \Delta + \eta_t^2 \|\mathbb{H}_{\mathcal{S}_t}\|^2 \\
&\quad + [2\eta_t^2 \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + 2\eta_t(1 - \eta_t) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle] \|\delta_{\mathcal{S}_t}\| \|\mathbb{H}_{\mathcal{S}_t}\| \\
&= -\eta_t \wp \|\delta_{\mathcal{S}_t}\|^2 - \eta_t^2 \Delta_{t+1} - \eta_t \tilde{\mathfrak{O}} \|\mathbb{H}_{\mathcal{S}_t}\|
\end{aligned}$$

where $\wp = 2(1 - \eta_t) (1 - \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta)$, $\tilde{h} = (-\eta_t \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + (\eta_t - 1) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle)$, $\tilde{\mathfrak{O}} = [2\tilde{h} \|\delta_{\mathcal{S}_t}\| - \eta_t \|\mathbb{H}_{\mathcal{S}_t}\|]$.

The correct term $\|\mathbb{H}_{\mathcal{S}_t}\|$ is absolutely effective if $\tilde{\mathfrak{O}} > 0$:

$$\begin{aligned}
\tilde{\mathfrak{O}} = [2\tilde{h} \|\delta_{\mathcal{S}_t}\| - \eta_t \|\mathbb{H}_{\mathcal{S}_t}\|] &> 0 \Rightarrow 2\frac{\tilde{h}}{\eta_t} \|\delta_{\mathcal{S}_t}\| > \|\mathbb{H}_{\mathcal{S}_t}\| \Rightarrow \|\mathbb{H}_{\mathcal{S}_t}\| < 2\frac{\tilde{h}}{\eta_t} \|\delta_{\mathcal{S}_t}\| \\
\tilde{h} = -\eta_t \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle + (\eta_t - 1) \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle &> 0 \Rightarrow \frac{1 - \eta_t}{\eta_t \sigma_\Delta} \cos\langle \delta_{\mathcal{S}_t}, \mathbb{H}_{\mathcal{S}_t} \rangle < \cos\langle \delta_{\mathcal{S}_t}^K, \mathbb{H}_{\mathcal{S}_t} \rangle
\end{aligned}$$

□

D.3 Theorem of LA-SA

For LA-SA, we refer to the form of QHM [32]: $d^t = (1 - \beta_t)\mathbb{G}_{\mathcal{S}_t} + \beta_t d^{t-1}$, $x^{t+1} = x^t - \eta_t [(1 - \nu_t)\mathbb{G}_{\mathcal{S}_t} + \nu_t d^t]$ where the parameter $\nu_t \in [0, 1]$ interpolates between SGD [33] ($\nu_t = 0$) and (normalized) SHB[34] ($\nu_t = 1$). When the parameters η_t , β_t and ν_t are held constant (thus the subscript t can be omitted) and $\nu = \beta$, it recovers a normalized variant of NAG [24] with an additional coefficient $1 - \beta_t$ on the stochastic gradient term. For adaptive learning rate methods (e.g., Adam ($\phi(\mathbb{G}_{\mathcal{S}_t}) = \sqrt{\beta_2 \mathbb{G}_{\mathcal{S}_t}^2 + (1 - \beta_2)v_{t-1}}$) [20], RMSProp [35], Adagrad [36], etc.), we simply abstract them as $\eta_t^\phi = \frac{\eta_t}{\phi(\mathbb{G}_{\mathcal{S}_t})}$.

Theorem D.5. (LA-SA) Assuming the clients satisfy Assumptions 2.1 and 2.2, and the single-round update distance is:

$$\|x_{t+1} - x_{\mathcal{S}_t}^*\|^2 = \left(1 - \wp_{\nu, \beta}^{\phi, t}\right) \|\delta_{\mathcal{S}_t}^t\|^2 - \eta_t^\phi \left(\eta_t^\phi (1 - \nu_t \beta_t) \Delta_{t+1} + \nu_t \beta_t \tilde{\mathfrak{O}}_\phi \|d^{t-1}\| \right), \quad (12)$$

where $\wp_{\nu,\beta}^{\phi,t} = 2\hat{\eta}_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})(1-\cos\langle\delta_{S_t}^K, \delta_{S_t}\rangle\sigma_\Delta)$, $\tilde{\partial}_\phi = [\eta_{\nu,\beta}^{\phi,t}\|d^{t-1}\| - 2\hbar_\phi\|\delta_{S_t}\|]$, $\hbar_\phi = ((\hat{\eta}_{\nu,\beta}^{\phi,t}-1)\cos\langle\delta_{S_t}, d^{t-1}\rangle - \hat{\eta}_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle)$ and $\hat{\eta}_{\nu,\beta}^{\phi,t} = \eta_t^\phi(1-\nu_t\beta_t)$, $\eta_{\nu,\beta}^{\phi,t} = -\eta_t^\phi\nu_t\beta_t$.

Proof. Adaptive Method ($h_t = 0$):

$$x_{t+1}^{\text{LA-SA}} = x_t - \frac{\eta_t}{\phi(\mathbb{G}_{S_t})}((1-\nu_t)\mathbb{G}_{S_t} + \nu_t d^t) \Rightarrow x_{t+1}^{\text{LA-SA}} - x_t = -\hat{\eta}_{\nu,\beta}^{\phi,t}\mathbb{G}_{S_t} + \eta_{\nu,\beta}^{\phi,t}d^{t-1}$$

where $d^t = (1-\beta_t)\mathbb{G}_{S_t} + \beta_t d^{t-1}$, $\hat{\eta}_{\nu,\beta}^{\phi,t} = \eta_t^\phi(1-\nu_t\beta_t)$ and $\eta_{\nu,\beta}^{\phi,t} = -\eta_t^\phi\nu_t\beta_t$.

$$\begin{aligned} x_{t+1}^{\text{LA-SA}} - x_{S_t}^* - (x_t - x_{S_t}^*) &= \hat{\eta}_{\nu,\beta}^{\phi,t}(x_{t+1}^{\text{LA-FedAVG}} - x_{S_t}^* - (x_t - x_{S_t}^*)) + \eta_{\nu,\beta}^{\phi,t}d^{t-1} \\ x_{t+1}^{\text{LA-SA}} - x_{S_t}^* &= \hat{\eta}_{\nu,\beta}^{\phi,t}(x_{t+1}^{\text{LA-FedAVG}} - x_{S_t}^*) + (1-\hat{\eta}_{\nu,\beta}^{\phi,t})(x_t - x_{S_t}^*) + \eta_{\nu,\beta}^{\phi,t}d^{t-1} \end{aligned}$$

$$\begin{aligned} \|x_{t+1}^{\text{LA-SA}} - x_{S_t}^*\|^2 &= \left\| \hat{\eta}_{\nu,\beta}^{\phi,t}(x_{t+1}^{\text{LA-FedAVG}} - x_{S_t}^*) + (1-\hat{\eta}_{\nu,\beta}^{\phi,t})(x_t - x_{S_t}^*) + \eta_{\nu,\beta}^{\phi,t}d^{t-1} \right\|^2 \\ &= (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \|\delta_{S_t}^K\|^2 + (1-\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \|\delta_{S_t}\|^2 + (\eta_{\nu,\beta}^{\phi,t})^2 \|d^{t-1}\|^2 \\ &\quad + 2\hat{\eta}_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t}) \|\delta_{S_t}^K\| \|\delta_{S_t}\| \cos\langle\delta_{S_t}^K, \delta_{S_t}\rangle \\ &\quad + 2\hat{\eta}_{\nu,\beta}^{\phi,t}\eta_{\nu,\beta}^{\phi,t} \|\delta_{S_t}^K\| \|d^{t-1}\| \cos\langle\delta_{S_t}^K, d^{t-1}\rangle \\ &\quad + 2\eta_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t}) \|\delta_{S_t}\| \|d^{t-1}\| \cos\langle\delta_{S_t}, d^{t-1}\rangle \\ &= (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 (\|\delta_{S_t}\|^2 - \Delta) + (1-\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \|\delta_{S_t}\|^2 + (\eta_{\nu,\beta}^{\phi,t})^2 \|d^{t-1}\|^2 \\ &\quad + 2\hat{\eta}_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\sqrt{\|\delta_{S_t}\|^2 - \Delta} \|\delta_{S_t}\| \cos\langle\delta_{S_t}^K, \delta_{S_t}\rangle \\ &\quad + 2\hat{\eta}_{\nu,\beta}^{\phi,t}\eta_{\nu,\beta}^{\phi,t}\sqrt{\|\delta_{S_t}\|^2 - \Delta} \|d^{t-1}\| \cos\langle\delta_{S_t}^K, d^{t-1}\rangle \\ &\quad + 2\eta_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t}) \|\delta_{S_t}\| \|d^{t-1}\| \cos\langle\delta_{S_t}, d^{t-1}\rangle \\ &= \left(2(\hat{\eta}_{\nu,\beta}^{\phi,t})^2 - 2\hat{\eta}_{\nu,\beta}^{\phi,t} + 1 + 2\hat{\eta}_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}^K, \delta_{S_t}\rangle\sigma_\Delta \right) \|\delta_{S_t}\|^2 - (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \Delta \\ &\quad + (\eta_{\nu,\beta}^{\phi,t})^2 \|d^{t-1}\|^2 \\ &\quad + \left[2\hat{\eta}_{\nu,\beta}^{\phi,t}\eta_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle + 2\eta_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}, d^{t-1}\rangle \right] \|\delta_{S_t}\| \|d^{t-1}\| \end{aligned}$$

where $\delta_{S_t}^K = x_{t+1}^{\text{LA-FedAVG}} - x_{S_t}^*$, $\delta_{S_t} = x_t - x_{S_t}^*$

$$\begin{aligned} \|x_{t+1}^{\text{LA-SA}} - x_{S_t}^*\|^2 - \|\delta_{S_t}\|^2 &= \left(2(\hat{\eta}_{\nu,\beta}^{\phi,t})^2 - 2\hat{\eta}_{\nu,\beta}^{\phi,t} + 2\hat{\eta}_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}^K, \delta_{S_t}\rangle\sigma_\Delta \right) \|\delta_{S_t}\|^2 - (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \Delta_{t+1} + (\eta_{\nu,\beta}^{\phi,t})^2 \|d^{t-1}\|^2 \\ &\quad + \left[2\hat{\eta}_{\nu,\beta}^{\phi,t}\eta_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle + 2\eta_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}, d^{t-1}\rangle \right] \|\delta_{S_t}\| \|d^{t-1}\| \\ &= -\hat{\eta}_{\nu,\beta}^{\phi,t}\wp\|\delta_{S_t}\|^2 - (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \Delta_{t+1} \\ &\quad + \left[(\eta_{\nu,\beta}^{\phi,t})^2 \|d^{t-1}\| + \left(2\hat{\eta}_{\nu,\beta}^{\phi,t}\eta_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle + 2\eta_{\nu,\beta}^{\phi,t}(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}, d^{t-1}\rangle \right) \|\delta_{S_t}\| \right] \|d^{t-1}\| \\ &= -\hat{\eta}_{\nu,\beta}^{\phi,t}\wp\|\delta_{S_t}\|^2 - (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \Delta_{t+1} \\ &\quad + \eta_{\nu,\beta}^{\phi,t} \left[\hat{\eta}_{\nu,\beta}^{\phi,t} \|d^{t-1}\| + \left(2\hat{\eta}_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle + 2(1-\hat{\eta}_{\nu,\beta}^{\phi,t})\cos\langle\delta_{S_t}, d^{t-1}\rangle \right) \|\delta_{S_t}\| \right] \|d^{t-1}\| \\ &= -\hat{\eta}_{\nu,\beta}^{\phi,t}\wp\|\delta_{S_t}\|^2 - (\hat{\eta}_{\nu,\beta}^{\phi,t})^2 \Delta_{t+1} \\ &\quad - \eta_{\nu,\beta}^{\phi,t} \left[-\eta_{\nu,\beta}^{\phi,t} \|d^{t-1}\| + \left(-2\hat{\eta}_{\nu,\beta}^{\phi,t}\sigma_\Delta\cos\langle\delta_{S_t}^K, d^{t-1}\rangle + 2(\hat{\eta}_{\nu,\beta}^{\phi,t}-1)\cos\langle\delta_{S_t}, d^{t-1}\rangle \right) \|\delta_{S_t}\| \right] \|d^{t-1}\| \\ &= -\eta_t^\phi(1-\nu_t\beta_t)\wp\|\delta_{S_t}\|^2 - (\eta_t^\phi(1-\nu_t\beta_t))^2 \Delta_{t+1} - \eta_t^\phi\nu_t\beta_t\tilde{\partial}_\phi\|d^{t-1}\| \end{aligned}$$

where $\wp = 2(1 - \hat{\eta}_{\nu,\beta}^{\phi,t}) (1 - \cos\langle \delta_{\mathcal{S}_t}^K, \delta_{\mathcal{S}_t} \rangle \sigma_\Delta)$, $\hbar_\phi = (-\hat{\eta}_{\nu,\beta}^{\phi,t} \cos\langle \delta_{\mathcal{S}_t}^K, d^{t-1} \rangle + (\hat{\eta}_{\nu,\beta}^{\phi,t} - 1) \cos\langle \delta_{\mathcal{S}_t}, d^{t-1} \rangle)$, $\tilde{\wp}_\phi = [\eta_{\nu,\beta}^{\phi,t} \|d^{t-1}\| - 2\hbar_\phi \|\delta_{\mathcal{S}_t}\|]$.

And $\hat{\eta}_{\nu,\beta}^{\phi,t} = \eta_t^\phi (1 - \nu_t \beta_t)$ and $\eta_{\nu,\beta}^{\phi,t} = -\eta_t^\phi \nu_t \beta_t$.

The correct term $\|d^{t-1}\|$ is absolutely effective if $\tilde{\wp}_\phi < 0$:

$$\begin{aligned} \tilde{\wp} &= [2\hbar \|\delta_{\mathcal{S}_t}\| - \eta_t \|d^{t-1}\|] > 0 \Rightarrow 2\frac{\hbar}{\eta_t} \|\delta_{\mathcal{S}_t}\| > \|d^{t-1}\| \Rightarrow \|d^{t-1}\| < 2\frac{\hbar}{\eta_t} \|\delta_{\mathcal{S}_t}\| \\ \hbar &= -\eta_t \sigma_\Delta \cos\langle \delta_{\mathcal{S}_t}^K, d^{t-1} \rangle + (\eta_t - 1) \cos\langle \delta_{\mathcal{S}_t}, d^{t-1} \rangle > 0 \\ &\Rightarrow \frac{1 - \eta_t}{\eta_t \sigma_\Delta} \cos\langle \delta_{\mathcal{S}_t}, d^{t-1} \rangle < \cos\langle \delta_{\mathcal{S}_t}^K, d^{t-1} \rangle \end{aligned}$$

□

References

- [1] Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. Momentum benefits non-iid federated learning simply and provably. In *ICLR*, 2024.
- [2] Junhyung Lyle Kim, Taha Toghani, Cesar A Uribe, and Anastasios Kyrillidis. Adaptive federated learning with auto-tuned clients via local smoothness. In *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.
- [3] Xiaowen Jiang, Anton Rodomanov, and Sebastian U Stich. Federated optimization with doubly regularized drift correction. *arXiv preprint arXiv:2404.08447*, 2024.
- [4] Divyansh Jhunjunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [5] 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*, pages 15164–15172, 2024.
- [6] Yan Sun, Li Shen, and Dacheng Tao. Understanding how consistency works in federated learning via stage-wise relaxed initialization. *Advances in Neural Information Processing Systems*, 36:80543–80574, 2023.
- [7] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*, pages 18250–18280. PMLR, 2022.
- [8] Junbo Li, Ang Li, Chong Tian, Qirong Ho, Eric Xing, and Hongyi Wang. Fednar: Federated optimization with normalized annealing regularization. *Advances in Neural Information Processing Systems*, 36:74753–74763, 2023.
- [9] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.
- [10] 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*, pages 5132–5143. PMLR, 2020.
- [11] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pages 15750–15769. PMLR, 2022.
- [12] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.
- [13] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.
- [14] Bo Li, Xiaowen Jiang, Mikkel N Schmidt, Tommy Sonne Alstrøm, and Sebastian U Stich. An improved analysis of per-sample and per-update clipping in federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

- [15] 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, December 2022.
- [16] Sohom Mukherjee, Nicolas Loizou, and Sebastian U Stich. Locally adaptive federated learning. *arXiv preprint arXiv:2307.06306*, 2023.
- [17] Yujia Wang, Shiqiang Wang, Songtao Lu, and Jinghui Chen. Fadas: Towards federated adaptive asynchronous optimization. *arXiv preprint arXiv:2407.18365*, 2024.
- [18] Xiaoyun Li and Ping Li. Analysis of error feedback in federated non-convex optimization with biased compression: Fast convergence and partial participation. In *International Conference on Machine Learning*, pages 19638–19688. PMLR, 2023.
- [19] Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Haochuan Li, Alexander Rakhlin, and Ali Jadbabaie. Convergence of adam under relaxed assumptions. *Advances in Neural Information Processing Systems*, 36:52166–52196, 2023.
- [22] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl akad nauk Sssr*, volume 269, page 543, 1983.
- [23] Mahmoud Assran and Michael Rabbat. On the convergence of nesterov’s accelerated gradient method in stochastic settings. *arXiv preprint arXiv:2002.12414*, 2020.
- [24] Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, Cham, 2018.
- [25] T. Chen and H. Chen. Approximations of continuous functionals by neural networks with application to dynamic systems. *IEEE Transactions on Neural Networks*, 4(6):910–918, 1993.
- [26] Stinchcombe and White. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. In *International 1989 Joint Conference on Neural Networks*, pages 613–617 vol.1, 1989.
- [27] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.
- [28] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.
- [29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [30] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [31] Roman Vershynin. High-dimensional probability, 2009.
- [32] Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. Understanding the role of momentum in stochastic gradient methods. *Advances in neural information processing systems*, 32, 2019.
- [33] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- [34] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.
- [35] 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.
- [36] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [37] Jiayi Wang, Shiqiang Wang, Rong-Rong Chen, and Mingyue Ji. A new theoretical perspective on data heterogeneity in federated optimization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 51650–51700, 2024.

- [38] Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. In *International Conference on Machine Learning*, pages 22802–22838. PMLR, 2022.
- [39] Ziqing Fan, Shengchao Hu, Jiangchao Yao, Gang Niu, Ya Zhang, Masashi Sugiyama, and Yanfeng Wang. Locally estimated global perturbations are better than local perturbations for federated sharpness-aware minimization. In *International Conference on Machine Learning*, pages 12858–12881. PMLR, 2024.
- [40] Xiaowen Jiang, Anton Rodomanov, and Sebastian U Stich. Stabilized proximal-point methods for federated optimization. *arXiv preprint arXiv:2407.07084*, 2024.
- [41] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.
- [42] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 2009.
- [43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [44] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [45] 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.
- [46] 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.
- [47] Roman Vershynin. A simple decoupling inequality in probability theory. *preprint*, 2011.
- [48] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.