FedSoL: Bridging Global Alignment and Local Generality in Federated Learning

Gihun Lee[1], **Minchan Jeong**[1], **Sangmook Kim**[1], **Jaehoon Oh**[2], **Se-Young Yun**[1],
[1]Graduate School of AI, KAIST
[2]Samsung Advanced Institute of Technology
{opcrisis, mcjeong, sangmook.kim, yunseyoung}@kaist.ac.kr, jh0104.oh@gmail.com

Abstract

While FL enables learning a model with data privacy, it often suffers from significant performance degradation when client data distributions are heterogeneous. Many previous FL algorithms have addressed this issue by introducing various proximal restrictions. These restrictions aim to encourage global alignment by constraining the deviation of local learning from the global objective. However, they inherently limit local learning by interfering with the original local objectives. Recently, an alternative approach has emerged to improve local learning generality. By obtaining local models within a smooth loss landscape, this approach mitigates conflicts among different local objectives of the clients. Yet, it does not ensure stable global alignment, as local learning does not take the global objective into account. In this study, we propose *Federated Stability on Learning* (FedSoL), which combines both the concepts of global alignment and local generality. In FedSoL, the local learning seeks a parameter region robust against proximal perturbations. This strategy introduces an implicit proximal restriction effect in local learning while maintaining the original local objective for parameter update.

1 Introduction

Federated Learning (FL) is an emerging distributed learning framework that preserves data privacy while leveraging client data for training [23, 24]. In this approach, individual clients train their local models using their private data, while the server aggregates these models into a global model. By precluding the need for direct access to private data, FL enables the utilization of extensive data collected from edge devices such as mobile phones, vehicles, and facilities [3, 55].

However, FL encounters a notorious challenge known as data heterogeneity [19]. Due to the diverse underlying distributions of the clients, the local datasets are non-independent and identically distributed (Non-IID). Its inevitable occurrence in many real-world scenarios leads to an inconsistency between global and local objectives, often significantly degrading performance [31, 33].

To tackle the data heterogeneity problem, most prior studies have introduced various *proximal restriction* into the local objective [20, 27, 30, 32]. These restrictions are designed to maintain alignment between global and local objectives by preventing the deviation of local learning from the global objective. Nevertheless, this approach inherently limits local learning by interfering with the original local objectives [37, 56]. Furthermore, it often falls into a sharp global landscape under high heterogeneity, which results in unreliable minima and poor stability [44, 47] (Figure 1 (a)).

In contrast, an alternative approach has emerged that focuses on the *local generality* in FL [37, 44]. Building on recent findings that highlight the benefits of smooth loss landscape for better generalization [5, 17, 18], this approach aims to seek flatness during local learning by employing the recently proposed Sharpness-Aware Minimization (SAM) [13] as the local optimizer [44]. By

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

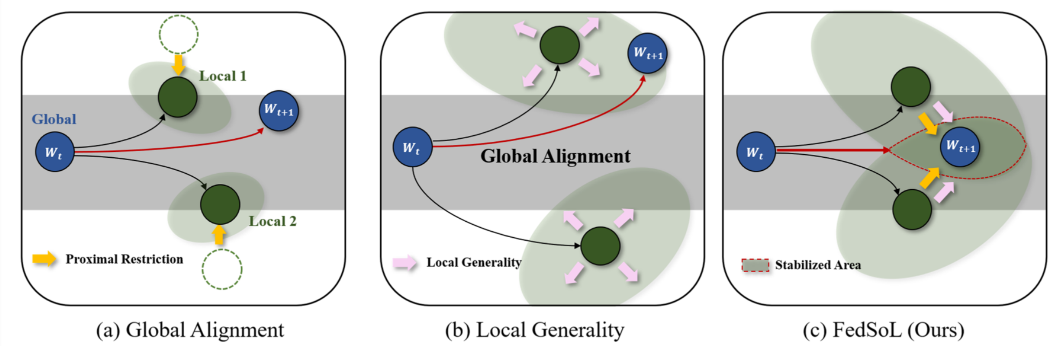

Figure 1: An overview of FL scenarios. Shaded regions stand for the degree of local generality of the trained local models. FedSoL ensures global alignment while promoting local generality.

improving local generality, this approach mitigates the conflicts between individual local objectives, contributing to the overall smoothness of the aggregated global model [4, 47]. Although these approaches have demonstrated competitive performance without the proximal restrictions, their ability to generalize well within their respective local distributions does not necessarily ensure alignment with the global objective (Figure 1 (b)).

Our key motivation is to tackle data heterogeneity by harnessing both the strengths of global alignment and local generality. To this end, we propose a novel algorithm Federated Stability on Learning (FedSoL) (Figure 1 (c)). FedSoL seamlessly incorporates the proximal restriction effect into the SAM optimization, without interfering with the original local objective. More specifically, FedSoL updates the local model using the gradient of the original local objective, which is determined at the weights perturbed by the gradient of the proximal restriction objective. By identifying a parameter region that is minimally influenced by the proximal perturbation, FedSoL diminishes the negative impact of local updates on global alignment, while maintaining the original local objective to preserve local generality. We comprehensively demonstrate the efficacy of FedSoL on global alignment and local generality. Experimental results show FedSoL achieves state-of-the-art performance in various setups. To summarize, our main contributions are as follows:

- We propose FedSoL, a novel and effective FL algorithm that leverages both the strengths of global alignment and local generality. FedSoL conducts proximal perturbations with the SAM strategy, yielding an implicit regularization effect during local learning. **(Section 3**
- We validate the efficacy of FedSoL on various setups and show that it consistently achieves state-of-the-art performances. We highlight FedSoL performs exceptionally well under high levels of heterogeneity. **(Section 4)**
- We provide a comprehensive analysis of the benefits that FedSoL brings to FL. Not only does it enhance the smoothness of the global model, but it also preserves global knowledge during local learning. **(Section 5)**

2 Background

2.1 Proximal Restriction

Consider an FL system that consists of K clients and a central server. Each client k has a local dataset $\mathcal{D}^k$, where the entire dataset is a union of the local datasets as $\mathcal{D} = \bigcup_{k \in [K]} \mathcal{D}^k$. FL aims to train a global server model with weights $\boldsymbol{w}$ that minimize the loss across all clients:

$$\mathcal{L}_{\text{global}}(\boldsymbol{w}) = \sum_{k \in [K]} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}) \,, \tag{1}$$

where $|\mathcal{D}^k|$ and $|\mathcal{D}|$ are the number of instances in each datasets. When using a proximal restriction objective, the loss function for each client k is a linear combination of its original local loss, $\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$, and a proximal loss, $\mathcal{L}^k_p(\boldsymbol{w}_k; \boldsymbol{w}_g)$, controlled by a hyperparameter β:

$$\mathcal{L}^k(\boldsymbol{w}_k) = \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) + \beta \cdot \mathcal{L}^k_p(\boldsymbol{w}_k; \boldsymbol{w}_g). \tag{2}$$

Here, $\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$ is the loss on the client's local distribution (e.g., cross-entropy loss), and $\mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)$ quantifies the discrepancy between the global model $\boldsymbol{w}_g$ and the local model $\boldsymbol{w}_k$. This discrepancy can be measured in various ways, such as the Euclidean distance between the two parameters [32] or the KL-divergence between probability vectors computed using the client's data [27]. Introducing proximal restriction within the local objective constrains the deviation of local learning from the global objective [20].

2.2 Overview of SAM

The SAM [13] optimizer pursues flatter minima for the given loss $\mathcal{L}$ by solving the minimax problem:

$$\min_{\boldsymbol{w}} \max_{\|\boldsymbol{\epsilon}\|_2<\rho} \mathcal{L}(\boldsymbol{w}+\boldsymbol{\epsilon}). \tag{3}$$

In the above equation, the inner maximization finds a parameter perturbation $\boldsymbol{\epsilon}$ that induces maximal loss change within the ρ-ball neighborhood. In practice, it is approximated by a single re-scaled gradient step $\boldsymbol{\epsilon}^*{=}\rho\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w})/\|\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w})\|_2$. Then, the outer minimization is conducted by a base optimizer such as SGD [39], by taking the gradient $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}+\boldsymbol{\epsilon}^*)$ at the perturbed weights. SAM demonstrates an exceptional ability to perform well across different model structures [6, 58] and tasks [1, 52] with high generalization performance. In FL, using SAM enhances the generalization of each client's local model [4, 44]. We further provide the related literature in Appendix M.

3 FedSoL: Federated Stability on Learning

3.1 Proximal Perturbation

In local learning, the local model $\boldsymbol{w}_k$ begins with the same parameters as the distributed global model $\boldsymbol{w}_g$, thereby initially having a minimal proximal loss. The main challenge is guiding the local learning to reduce the original local loss $\mathcal{L}^k_{\text{local}}$ without inducing an increase in the proximal loss $\mathcal{L}^k_p$. We address this problem in the context of SAM optimization by seeking a gradient that not only minimizes the original local loss $\mathcal{L}^k_{\text{local}}$, but also is robust against the increases in proximal loss $\mathcal{L}^k_p$. To this end, FedSoL aims to minimize the original local loss, which is minimally affected by the weight perturbation that maximizes proximal loss. By decoupling the roles of these two types of losses, FedSoL conducts the following steps for each local update of client k:

Step1: Weight Perturbation FedSoL finds a weight perturbation $\boldsymbol{\epsilon}^*_p$ that causes the most significant change for any given proximal loss $\mathcal{L}^k_p$:

$$\boldsymbol{\epsilon}^*_p = \rho\frac{\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)}{\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)\|_2} \approx \underset{\|\boldsymbol{\epsilon}\|_2\leqslant\rho}{\operatorname{argmax}}\ \mathcal{L}^k_p(\boldsymbol{w}_k{+}\boldsymbol{\epsilon};\boldsymbol{w}_g)\,. \tag{4}$$

Step2: Parameter Update After perturbation, FedSoL updates parameters by computing the gradient of the original local loss $\mathcal{L}^k_{\text{local}}$ at this perturbed weights:

$$\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k - \gamma\cdot\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k+\boldsymbol{\epsilon}^*_p)\,, \tag{5}$$

where γ is a learning rate. In the above procedures, the update gradient is computed on the original local loss $\mathcal{L}^k_{\text{local}}$, whereas the proximal loss $\mathcal{L}^k_p$ only plays in an implicit role. Note that $\boldsymbol{\epsilon}^*_p$ is used solely for weight perturbation, thereby we do not need to compute its gradient. To clarify how FedSoL influences local learning, we analyze the update gradient $\boldsymbol{g}_u$ of FedSoL by employing the first-order Taylor approximation of $\mathcal{L}^k_{\text{local}}$ at $\boldsymbol{w}_k$, with perturbation $\boldsymbol{\epsilon}^*_p$:

$$\begin{aligned}\boldsymbol{g}_u(\boldsymbol{w}_k) &= \nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k+\boldsymbol{\epsilon}^*_p)\\ &\approx \boldsymbol{g}_l(\boldsymbol{w}_k) + \rho\,\nabla^2_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k)\,.\end{aligned} \tag{6}$$

We denote $\boldsymbol{g}_l{=}\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$, $\boldsymbol{g}_p{=}\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)$, and $\hat{\boldsymbol{g}}_p{=}\boldsymbol{g}_p/\|\boldsymbol{g}_p\|_2$, omitting $\boldsymbol{w}_k$ and $\boldsymbol{w}_g$ if there is no conflict. Based on Equation (6), we examine the change of each loss induced by a single local update with a learning rate γ, as defined in Equation (7):

$$\begin{aligned}\Delta^{\text{algo}}\mathcal{L}^k(\boldsymbol{w}_k) &= \mathcal{L}^k\big(\boldsymbol{w}_k-\gamma\boldsymbol{g}_u(\boldsymbol{w}_k)\big) - \mathcal{L}^k(\boldsymbol{w}_k)\\ &\approx -\gamma\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k(\boldsymbol{w}_k),\boldsymbol{g}_u(\boldsymbol{w}_k)\rangle\,.\end{aligned} \tag{7}$$

In the above equation, we apply a first-order Taylor approximation to $\mathcal{L}^k$ at $\boldsymbol{w}_k$, taking into account a local update by $-\gamma \boldsymbol{g}_u(\boldsymbol{w}_k)$. Here, $\mathcal{L}^k$ can be either local loss or proximal loss. The detailed procedure of FedSoL is outlined in Appendix A. By combining the approximation for $\boldsymbol{g_u}$ from Equation (6) into the loss difference in Equation (7), we derive the following two key propositions.

Proposition 1. *Given a convex local loss $\mathcal{L}^k_{\text{local}}$, the change of proximal loss $\mathcal{L}^k_p$ by FedSoL update with a learning rate γ reduces the conflicts between $\boldsymbol{g}_l$ and $\boldsymbol{g}_p$ as ρ grows:*

$$\Delta^{\text{FedSoL}} \mathcal{L}^k_p \approx -\gamma \Big(\langle \boldsymbol{g}_l \, , \boldsymbol{g}_p \rangle + \rho \cdot \underbrace{\hat{\boldsymbol{g}}_p^\top \nabla^2 \mathcal{L}^k_{\text{local}} \, \boldsymbol{g}_p}_{\geqslant 0} \Big) \, , \tag{8}$$

where $\nabla^2 \mathcal{L}^k_{\text{local}}$ is the Hessian at $\boldsymbol{w}_k$. In **Proposition 1**, we study the change of proximal loss after the FedSoL update, denoted as $\Delta^{\text{FedSoL}} \mathcal{L}^k_p$. The proposition suggests that FedSoL implicitly regularizes local learning for global alignment, reducing the negative impact of local updates on proximal loss. This regularization effect grows as the curvature of local loss $\nabla^2 \mathcal{L}^k_{\text{local}}$ becomes steeper. Note that when ρ is set to 0, Equation (8) becomes the change of proximal loss after FedAvg update, $\Delta^{\text{FedAvg}} \mathcal{L}^k_p$.

Proposition 2. *The change of original local loss $\mathcal{L}^k_{\text{local}}$ by FedSoL update with a learning rate γ is equivalent to the FedAvg update at $\nabla \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k + \frac{\rho}{2} \boldsymbol{\epsilon}^*_p)$ as:*

$$\Delta^{\text{FedSoL}} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) \approx \Delta^{\text{FedAvg}} \mathcal{L}^k_{\text{local}} \Big(\boldsymbol{w}_k + \frac{\rho}{2} \boldsymbol{\epsilon}^*_p \Big) \, . \tag{9}$$

Meanwhile, **Proposition 2** analyzes how the original local loss changes after the FedSoL update, $\Delta^{\text{FedSoL}} \mathcal{L}^k_{\text{local}}$, and compare it to its counterpart in FedAvg, $\Delta^{\text{FedAvg}} \mathcal{L}^k_{\text{local}}$. The proposition suggests that even though FedSoL calculates the gradient of the original local loss at the weight perturbed by proximal gradient, its behavior on the original local loss is almost identical to that of FedAvg with standard gradient descent using $\mathcal{L}^k_{\text{local}}$. This implies that FedSoL does not noticeably interfere with or slow down the learning process on the local data distribution. The detailed proofs for the propositions are provided in Appendix L. We further analyze the effect of FedSoL in Appendix K, with analysis on possible various perturbation strategies in Appendix F.

4 Experiment

4.1 Experimental Setups

Data Setups We employ 6 datasets: MNIST [9], CIFAR-10 [25], SVHN [40], CINIC-10 [7], PathMNIST [54], and TissueMNIST [54]. We distribute data to clients via two strategies: Sharding [36] and Latent Dirichlet Allocation (LDA) [50]. Sharding sorts data by label and assigns equal-size shards to clients. The heterogeneity increases as the shard per user, s, becomes smaller. On the other hand, LDA assigns class c data samples to each client k with probability $p_c (\approx \text{Dir}(\alpha))$.

Learning Setups We distribute CIFAR-10, and SVHN datasets across 100 clients with a sampling ratio of 0.1, while CINIC-10, PathMNIST, and TissueMNIST across 200 clients with a ratio of 0.05. We use a model architecture as described in [36], which consists of two convolutional layers, max-pooling layers, and two fully connected layers. Each client optimizes its local datasets for 5 local epochs using momentum SGD with a learning rate of 0.01, momentum 0.9, and weight decay 1e-5. The learning rate is decayed by a factor of 0.99 at every communication round. We conduct 300 communication rounds in general, and 200 for PathMNIST, and TissueMNIST. We use KL-divergence loss as the proximal loss. Please see more detailed experimental setups in Appendix C.

4.2 Performance on Data Heterogeneity

Heterogeneity Level Table 1 presents a comparison between our approach, FedSoL, and other baselines such as FedProx [32], FedNova [51], Scaffold [20], FedNTD [27], FedSAM [44], and FedASAM [4] as baseline methods. Notably, many recently proposed FL methods tend to underperform when compared to the standard FedAvg baseline, where a similar observation is reported in [27, 57]. In contrast, FedSoL consistently exceeds the performance of FedAvg across all evaluated scenarios. FedSoL achieves state-of-the-art results in most cases, particularly showing consistent improvement on high heterogeneity levels (s=2 and α=0.05).

Table 1: Test accuracy@1(%) comparison among baselines and FedSoL. The values in the parenthesis are the standard deviation. The arrow ($\downarrow$, $\uparrow$) shows the comparison to the FedAvg. We set $s \in \{2, 5, 10\}$ and $\alpha \in \{0.05, 0.1, 0.5\}$ for CIFAR-10 datasets, whereas $s = 2$ and $\alpha = 0.1$ for the others.

Non-IID Partition Strategy : Sharding

Method	**CIFAR-10** $s=2$	**CIFAR-10** $s=5$	**CIFAR-10** $s=10$	**SVHN**	**CINIC-10**	**PathMNIST**	**TissueMNIST**
FedAvg	51.48 (3.41)	70.96 (0.91)	74.60 (0.88)	73.63 (3.16)	42.40 (2.70)	57.40 (1.48)	49.36 (1.64)
FedProx	52.80 (2.66) $\uparrow$	64.71 (0.74) $\downarrow$	69.37 (1.21) $\downarrow$	71.09 (3.13) $\downarrow$	40.00 (3.01) $\downarrow$	60.77 (3.64) $\uparrow$	48.20 (1.95) $\downarrow$
FedNova	46.89 (2.57) $\downarrow$	67.11 (0.25) $\downarrow$	70.59 (0.52) $\downarrow$	67.35 (2.84) $\downarrow$	40.94 (2.29) $\downarrow$	58.85 (4.10) $\uparrow$	36.44 (0.95) $\downarrow$
Scaffold	62.60 (0.70) $\uparrow$	74.28 (0.39) $\uparrow$	76.71 (0.16) $\uparrow$	77.84 (2.28) $\uparrow$	47.76 (0.45) $\uparrow$	71.12 (1.04) $\uparrow$	30.99 (6.09) $\downarrow$
FedNTD	**67.25** (1.08) $\uparrow$	**70.47** (0.33) $\uparrow$	76.46 (0.07) $\uparrow$	**85.30** (0.78) $\uparrow$	52.72 (1.12) $\uparrow$	65.00 (1.26) $\uparrow$	52.63 (0.59) $\uparrow$
FedSAM	51.85 (3.14) $\uparrow$	69.29 (0.39) $\downarrow$	72.98 (0.34) $\downarrow$	65.85 (3.77) $\downarrow$	45.91 (2.02) $\uparrow$	67.32 (3.15) $\uparrow$	49.62 (1.61) $\uparrow$
FedASAM	52.08 (2.19) $\uparrow$	63.24 (1.16) $\uparrow$	74.74 (0.88) $\uparrow$	79.48 (2.17) $\downarrow$	43.15 (2.73) $\uparrow$	59.47 (2.91) $\uparrow$	49.46 (1.91) $\uparrow$
FedSoL (Ours)	**66.72** (0.61) $\uparrow$	**69.88** (0.15) $\uparrow$	**77.79** (0.19) $\uparrow$	**85.18** (0.37) $\uparrow$	**55.17** (0.32) $\uparrow$	**73.85** (1.55) $\uparrow$	**53.42** (0.46) $\uparrow$

Non-IID Partition Strategy : LDA

Method	**CIFAR-10** $\alpha=0.05$	**CIFAR-10** $\alpha=0.1$	**CIFAR-10** $\alpha=0.5$	**SVHN**	**CINIC-10**	**PathMNIST**	**TissueMNIST**
FedAvg	42.27 (1.34)	56.13 (0.78)	73.90 (0.66)	55.36 (4.85)	36.49 (4.37)	65.98 (4.76)	42.78 (2.03)
FedProx	50.58 (0.57) $\uparrow$	59.80 (1.12) $\uparrow$	72.87 (0.55) $\uparrow$	72.40 (3.15) $\uparrow$	40.09 (3.97) $\uparrow$	70.44 (1.92) $\uparrow$	52.25 (1.40) $\uparrow$
FedNova	10.00 (*Failed*) $\downarrow$	10.00 (*Failed*) $\downarrow$	70.04 (0.45) $\downarrow$	53.07 (3.30) $\downarrow$	21.89 (1.71) $\downarrow$	38.94 (2.34) $\downarrow$	15.03 (3.74) $\downarrow$
Scaffold	10.00 (*Failed*) $\downarrow$	10.00 (*Failed*) $\downarrow$	**75.49** (0.21) $\uparrow$	21.46 (1.75) $\downarrow$	16.89 (2.25) $\downarrow$	18.07 (0.04) $\downarrow$	32.04 (0.07) $\downarrow$
FedNTD	58.08 (0.48) $\uparrow$	**63.16** (1.02) $\uparrow$	74.91 (0.33) $\uparrow$	79.25 (0.61) $\uparrow$	50.22 (3.71) $\uparrow$	74.26 (1.25) $\uparrow$	44.55 (1.95) $\uparrow$
FedSAM	36.14 (1.21) $\downarrow$	52.14 (0.94) $\downarrow$	70.74 (0.40) $\downarrow$	13.27 (2.78) $\downarrow$	36.70 (4.28) $\uparrow$	66.64 (3.76) $\uparrow$	44.07 (3.02) $\uparrow$
FedASAM	43.12 (1.25) $\uparrow$	57.00 (0.30) $\uparrow$	73.91 (0.51) $\uparrow$	60.25 (4.56) $\uparrow$	36.93 (4.60) $\uparrow$	69.45 (3.19) $\uparrow$	42.73 (2.35) $\uparrow$
FedSoL (Ours)	**60.01** (0.30) $\uparrow$	**64.13** (0.46) $\uparrow$	**75.60** (0.32) $\uparrow$	**83.92** (0.29) $\uparrow$	**55.07** (1.48) $\uparrow$	**78.88** (0.46) $\uparrow$	**53.40** (0.85) $\uparrow$

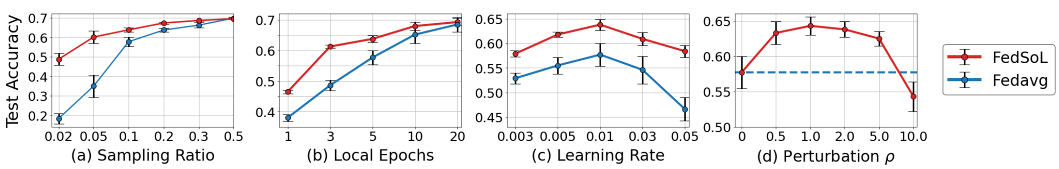

Figure 2: Performance of *FedAvg* and *FedSoL* on CIFAR-10 (α=0.1) with various setups: (a) sampling ratio, (b) the number of local epochs, (c) initial learning rate, and (d) perturbation strength. The error bars stand for the standard deviations.

Learning Factors In Figure 2, we examine the learning factors that influence FedSoL's performance: Partial Participation (Figure 2 (a)), Number of Local Epochs (Figure 2 (b)), and Learning Rate (Figure 2 (c). Throughout experiments, FedSoL consistently surpasses the FedAvg across varying factors. Most of all, FedSoL enlarges its gain as the smaller portion of clients participate in each round. For instance, FedAvg significantly declines in performance at a sampling ratio of 0.02, reaching to a near-random accuracy. However, FedSoL maintains robust performance under such condition.

Perturbation Strength In FedSoL, a hyperparameter ρ controls the overall perturbation strength. Figure 2 (d) plots FedSoL's performance against varying ρ values. Although the model often diverges when using the SAM strategy with high perturbation strength (as shown in Table 1), our FedSoL remains relatively robust and achieve its best performance within the ρ range between 0.5 and 2.0.

Additional Experiments We further provide the overall learning curves in Appendix D, performance on various model architectures in Appendix J, adaptive version of FedSoL in Appendix G, combination of various proximal losses in Appendix H, partial perturbation variant in Appendix I, and personalized performance in Appendix E.

5 Analysis

5.1 Knowledge Preservation

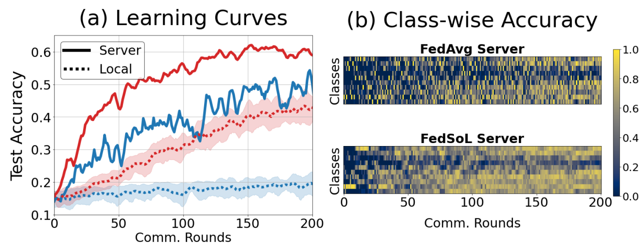

Figure 3: Comparison of FedAvg (*blue* lines) and FedSoL (ρ=2.0) (*red* lines) on CIFAR-10 (s=2). (a) shows learning curves for global and local models, with shaded areas reflecting standard deviation across clients. (b) exhibits the class-wise accuracy of the global model.

To understand how FedSoL stabilizes the local learning at the prediction level, we examine how well a local model maintains its performance on the global distribution after local learning. As illustrated in Figure 3 (a), FedAvg's local models undergo a significant drop in performance on the global distribution after local learning. Conversely, FedSoL maintains high performance, indicating better alignment of local learning with the global objective, and thus stabilizing the learning process. We further analyze the class-wise accuracy of FedAvg and FedSoL server models. As Figure 3 (b) demonstrates, while FedAvg exhibits significant fluctuations and inconsistent class-wise performance, FedSoL preserves its class-wise accuracy as the communication proceeds.

5.2 Smoothness of Loss Landscape

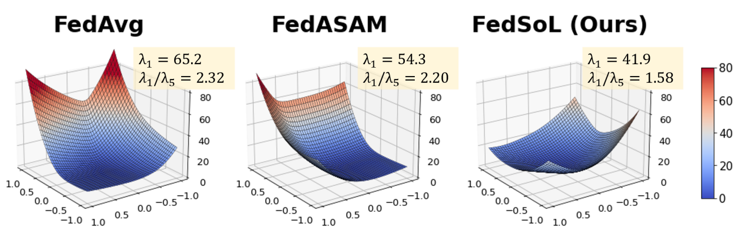

Figure 4: Loss landscape visualization of global model on CIFAR-10 LDA (α=0.1). The λ_1 and λ_5 in each figure stand for the top-1 and top-5 eigenvalues of the Hessian matrix.

We visualize the loss landscapes [29] of global models obtained from FedAvg, FedASAM, and FedSoL in Figure 4. In these plots, each axis corresponds to one of the two dominant eigenvectors (top-1 and top-2) of the Hessian matrix, representing the directions of the most significant shifts in the loss landscape. Along with each landscape, we provide the value of the dominant eigenvalue (λ_1) and its ratio to the fifth largest eigenvalue (λ_1/λ_5), following the criteria used in [13, 38]. Here, FedSoL's smaller ratio indicates that the variations in loss are more evenly distributed across various directions. Both the landscape visualization and Hessian eigenvalues underscore the efficacy of FedSoL in smoothing the loss landscape.

6 Conclusion

In this study, we emphasize the importance of both global alignment and local generality in tackling data heterogeneity within FL. To unify these essential components, we propose Federated Stability on Learning (FedSoL), a novel and versatile method that seeks a robust parameter region against proximal weight perturbations. This allows for an implicit proximal restriction effect on local learning, without interfering with the original local objective. We present a comprehensive analysis of FedSoL and demonstrate its benefits in FL.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning, 90%] and [No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 10%].

References

[1] Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. In *International conference on machine learning*, pages 10–32. PMLR, 2022.

[2] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[3] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[4] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. *arXiv preprint arXiv:2203.11834*, 2022.

[5] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[6] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

[7] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[13] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[14] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[18] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

[19] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[20] 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.

[21] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[22] Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR, 2022.

[23] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[24] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[25] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6, 2009.

[26] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.

[27] Gihun Lee, Yongjin Shin, Minchan Jeong, and Se-Young Yun. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021.

[28] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[29] 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.

[30] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021.

[31] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[32] 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.

[33] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[34] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *arXiv preprint arXiv:2106.05001*, 2021.

[35] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pages 15070–15092. PMLR, 2022.

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

[37] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

[38] Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *arXiv preprint arXiv:2210.05177*, 2022.

[39] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.*, 2011.

[41] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[44] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. *arXiv preprint arXiv:2206.02618*, 2022.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

[47] Yan Sun, Li Shen, Shixiang Chen, Liang Ding, and Dacheng Tao. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. *arXiv preprint arXiv:2305.11584*, 2023.

[48] Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. *arXiv preprint arXiv:2302.10429*, 2023.

[49] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

[50] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[51] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020.

[52] Pengfei Wang, Zhaoxiang Zhang, Zhen Lei, and Lei Zhang. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3769–3778, 2023.

[53] Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pages 10334–10343. PMLR, 2020.

[54] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[55] 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.

[56] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10174–10183, 2022.

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

[58] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha Dvornek, Sekhar Tatikonda, James Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. *arXiv preprint arXiv:2203.08065*, 2022.

A Algorithm

Algorithm 1 Federated Stability on Learning (FedSoL)

Input: local loss $\mathcal{L}^k_{\text{local}}$ and proximal loss $\mathcal{L}^k_p$ for each client $k \in [K]$, learning rate γ, and base perturbation radius ρ
Initialize global server weight $\boldsymbol{w}_g$
for each communication round t **do**
    Server samples clients $K^{(t)} \subset [K]$
    Server broadcasts $\boldsymbol{w}_g$ for all $k \in K^{(t)}$
    Client replaces $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_g$
    for each client $k \in K^{(t)}$ **in parallel do**
        for each local step **do**
            # Set Adaptive Perturbation Radius (Sec 3.2)
            $\boldsymbol{\rho}_{\text{adaptive}} = \rho \cdot \boldsymbol{\Lambda}$ (*element-wise rescale*)
            # Perturb using Proximal Gradient (Sec 3.1)
            $\boldsymbol{\epsilon}^*_p = \boldsymbol{\rho}_{\text{adaptive}} \odot \frac{\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)}{\|\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_p(\boldsymbol{w}_k;\boldsymbol{w}_g)\|}$
            # Update Local Model Parameters (Sec 3.1)
            $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k - \gamma \cdot \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}\big(\boldsymbol{w}_k + \boldsymbol{\epsilon}^*_p\big)$
        end for
    end for
    Upload $\boldsymbol{w}_k$ to server
    Server Aggregation :$\boldsymbol{w}_g \leftarrow \frac{1}{|K^{(t)}|} \sum_{k \in K^{(t)}} \boldsymbol{w}_k$
end for
Server output : $\boldsymbol{w}_g$

B Table of Notations

Table 2: Table of Notations throughout the paper.

Indices:	
k	Index for clients ($k \in [K]$)
g	Index for global server
Environment:	
$\mathcal{D}$	Whole dataset
$\mathcal{D}^k$	Local dataset of the k-th client
α	Parameter for the Dirichlet Distribution
s	The number of shards per user
FL algorithms:	
β, μ	Multiplicative coefficient for the proximal loss
γ	Learning rate
ρ	Perturbation Radius for SAM-related algorithms
Weights:	
$\boldsymbol{w}_g$	Weight of the global server model on the round t
$\boldsymbol{w}_k$	Weight of the k-th client model on the round t
$\|\boldsymbol{w}_g - \boldsymbol{w}_k\|$	Collection of L^2-norm between server and client models, among all rounds.
Objective Functions:	
$\mathcal{L}^k_{\text{local}}$	Local objective for the k-th client
$\mathcal{L}^k_p$	Proximal Loss for the k-th client

C Experimental Setups

The code is implemented by PyTorch [43]. The overall code structure is based on FedML [14] library with some modifications for simplicity. We use 2 A6000 GPU cards, but without Multi-GPU training.

C.1 Model Architecture

For the primary experiments, we use the model architecture used in FedAvg [36], which consists of two convolutional layers with subsequent max-pooling layers, and two fully-connected layers. The same model is also used in [27, 30, 34]. We also conduct experiments on ResNet-18 [15], Vgg-11 [45], and SL-ViT [28]. For SL-ViT, we resize 28×28-sized images into 32×32 to fit in the required minimum patch size.

C.2 Datasets

To validate our algorithm, we employ 6 distinct datasets, as listed below. The values in the parentheses denote the number of samples used to *train* and *test*, respectively.

- **MNIST** [9] (60,000 / 10,000): contains hand-written digits images, ranging from 0 to 9. The data is augmented using Random Cropping, Random Horizontal Flipping, and Normalization. The data is converted to 3-channel RGB images.
- **CIFAR-10** [25] (50,000 / 10,000): contains a labeled subset of 80 Million Tiny Images [49] for 10 different classes. The data is augmented using Random Cropping, Horizontal Flipping, Normalization, and Cutout [10].
- **SVHN** [40] (73,257 / 26,032): contains digits of house numbers obtained from *Google Street View*. The data is augmented using Random Cropping, Random Horizontal Flipping, and Normalization.
- **CINIC-10** [7] (90,000 / 90,000): contains a combination of CIFAR and downsized ImageNet [8], which is compiled to serve as a bridge between the two datasets. The data is augmented using Random Cropping, Random Horizontal Flipping, and Normalization.
- **PathMNIST** [54] (110,000 / 7,180): contains non-overlapping patches from Hematoxylin & Eosin stained colorectal cancer histology slide images. The data is augmented using Random Horizontal Flipping, and Normalization.
- **TissueMNIST** [54] (189,106 / 47,280): contains microscope images of human kidney cortex cells, which are segmented from 3 reference tissue specimens. The data is augmented using Random Horizontal Flipping, and Normalization. The data is converted to 3-channel RGB images.

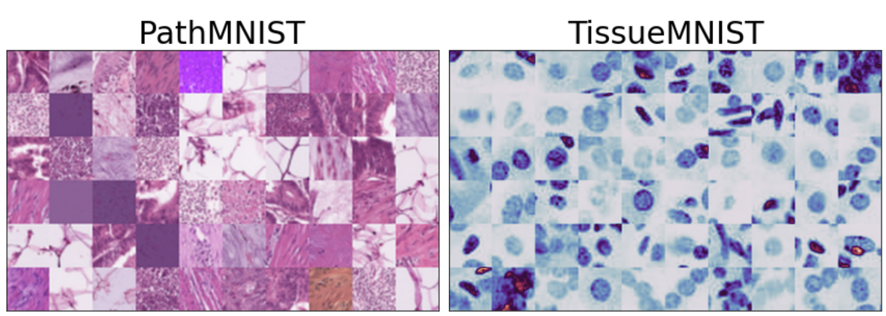

Figure 5: Example images from PathMNIST datasets and TissueMNIST datasets.

Note that we evaluate our algorithm is on medical imaging datasets - a crucial practical application of federated learning [3, 55]. Illustrative examples of the images are in Figure 5.

C.3 Non-IID Partition Strategy

To comprehensively address the data heterogeneity issue in federated learning, we distribute the local datasets using the following two distinct data partition strategies: (i) **Sharding** and (ii) **Latent Dirichlet Allocation** (LDA).

- **(i) Sharding** [27, 36, 41]: sorts the data by label and divide the data into shards of the same size, and distribute them to the clients. In this strategy, the heterogeneity level increases as the shard per user, s, becomes smaller, and vice versa. As the number of shards is the same across all the clients, *the dataset size is identical for each client*.
- **(ii) Latent Dirichlet Allocation (LDA)** [30, 34, 50]: allocates the data samples of class c to each client k with the probability p_c, where $p_c \approx \text{Dir}(\alpha)$. In this strategy, *both the distribution and dataset size differ for each client*. The heterogeneity level increases as the concentration parameter, α, becomes smaller, and vice versa.

Note that although only the statistical distributions varies across the clients in Sharding strategy, both the distribution and dataset size differ in LDA strategy. Figure 6 illustrates the difference between these partition strategies.

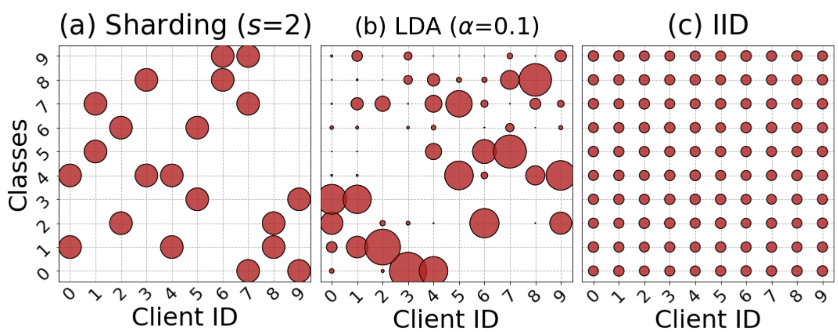

Figure 6: CIFAR-10 partition examples across 10 clients.

C.4 Learning Setups

We use a momentum SGD optimizer with an initial learning rate of 0.01, a momentum value of 0.9, and weight decay 1e-5. The momentum is employed only for local learning and is not uploaded to the server. Note that SAM optimization also requires its base optimizer, which performs the parameter update using the obtained gradient at the perturbed weights. The learning rate is decays with a factor of 0.99. As we are assuming a synchronized FL scenario, we simulate the parallel distributed learning by sequentially conducting local learning for the sampled clients and then aggregate them into a global model. The standard deviation is measured over 3 runs. The detailed learning setups for each datasets is provided in Table 3.

Table 3: Learning scenarios for each datasets.

Datasets	Clients	Comm. Rounds	Sampling Ratio
MNIST	100	200	0.1
CIFAR-10	100	300	0.1
SVHN	100	200	0.1
CINIC-10	200	300	0.05
PathMNIST	200	200	0.05
TissueMNIST	200	200	0.05

Table 4: Algorithm-specific hyperparameters.

Methods	Selected	Searched Candidates
FedAvg	None	None
FedProx	μ=1.0	$\mu \in \{0.1, 0.5, 1.0, 2.0\}$
Scaffold	None	None
FedNova	None	None
FedNTD	β=1.0, τ=1.0	$\beta \in \{0.5, 1.0\}$, $\tau \in \{1.0, 3.0\}$
FedSAM	ρ=0.1	$\rho \in \{0.1, 0.5, 1.0, 2.0\}$
FedASAM	ρ=1.0	$\rho \in \{0.1, 0.5, 1.0, 2.0\}$
FedDyn	None	None
MOON	μ=0.1, τ=0.5	$\mu \in \{0.1, 0.5\}$, $\tau \in \{0.5, 1.0\}$
FedSoL	$\rho = 2.0$	$\rho \in \{0.1, 0.5, 1.0, 2.0\}$

C.5 Algorithm Implementation Details

We search for hyperparameters and select the best among the candidates. The hyperparameters for each method is provided in Table 4. In the primary experiments, we use KL-divergence loss [16] with softened logits with temperature τ=3 for the proximal loss for the weight perturbation in FedSoL.

D Learning Curves

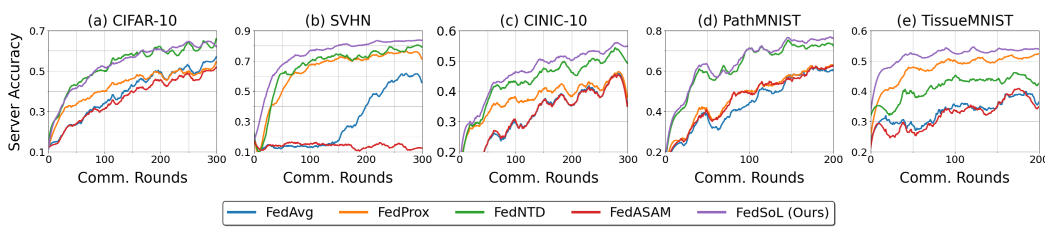

Figure 7: Learning curves of FL methods on LDA (α=0.1). The curves are smoothed for clear visualization.

To provide further insights into the learning process, we illustrate the learning curves of different FL methods in Figure 7. Although we utilize different communication rounds for each dataset, the performance of the model becomes sufficiently saturated at the end of communication rounds. For all datasets, FedSoL not only achieves a superior final model at the end of the communication round but also demonstrates much faster convergence. Moreover, although some algorithms that perform well on a dataset fail on another (ex. FedNTD [27] underperforms compared to FedProx [32] on the TissueMNIST datasets), FedSoL consistently exhibits significant improvements when compared to the other baselines.

E Personalized Performance

In Table 5, we compare FedSoL with several methods specifically designed for personalized federated learning (pFL): PerFedAvg [12], FedBabu [41], and kNN-Per [35]. Each method is assessed by fine-tuning them for e local epochs from the global model after the final communication round. As global alignment is unnecessary for the personalized model, we fine-tune FedSoL using original the local objective without perturbation and denote it as FedSoL-FT. The standard deviation is measured across the clients. The results reveal that our FedSoL-FT consistently outperforms other pFL methods under various scenarios. Furthermore, the gap is enlarged when local (e=1), implying that the global model obtained by FedSoL adapts more quickly to local distributions. We suggest that by integrating FedSoL with other methods specialized for pFL, we can attain superior performance for both the global server model and client local models.

Table 5: Personalized FL performance after τ epochs of fine-tuning. The heterogeneity level is set as LDA ($\alpha = 0.1$).

Method	e	**CIFAR-10**	**SVHN**	**TissueMNIST**
Local-only	-	84.7 ± 12.8	87.4 ± 13.0	82.4 ± 15.5
FedAvg	1	84.1 ± 13.4	86.6 ± 15.5	82.2 ± 17.5
	5	88.9 ± 8.9	92.1 ± 5.7	89.2 ± 10.1
PerFedAvg	1	80.5 ± 16.2	64.1 ± 30.3	82.3 ± 18.9
	5	86.3 ± 10.4	72.4 ± 21.2	88.8 ± 10.2
FedBabu	1	84.6 ± 12.7	88.7 ± 9.6	85.7 ± 14.3
	5	89.2 ± 8.4	92.7 ± 6.2	90.5 ± 8.8
kNN-Per	1	85.7 ± 12.3	86.4 ± 15.0	86.5 ± 14.2
	5	89.7 ± 8.1	92.8 ± 6.2	91.4 ± 7.5
FedSoL-FT (Ours)	1	**87.5** ± 9.7	**92.5** ± 7.4	**88.1** ± 12.2
	5	**90.5** ± 7.8	**95.0** ± 3.9	**91.6** ± 6.9

F Proximal Perturbation with SAM

In our work, we consider how to combine the proximal restriction into SAM optimization to improve overall FL performance. A straightforward approach might involve using the linearly combined local objective between the original local loss $\mathcal{L}^k_{\text{local}}$ and the proximal loss $\mathcal{L}^k_{\text{p}}$ in Equation (2) for the SAM optimization in Equation (3) as follows:

$$\min_{\boldsymbol{w}_k} \max_{\|\boldsymbol{\epsilon}\|_2 < \rho} \left[\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) + \beta \cdot \mathcal{L}^k_{p}(\boldsymbol{w}_k; \boldsymbol{w}_g) \right]. \tag{10}$$

In the above equation, the gradients for weight perturbation and parameter update are obtained from the same objective.

However, this approach encounters the same drawbacks as when using each method on its own. The combined loss also varies considerably across clients due to heterogeneous local distributions, causing the smoothness to largely rely on individual local distributions. Furthermore, the negative correlation between the gradients of the two objectives within the combined loss still limits local learning. Consequently, this approach neither encourages global alignment nor preserves the local generality as desired.

Instead in FedSoL, we overcome this issue by decoupling this directly combined loss into the proximal loss $\mathcal{L}^k_{\text{p}}$ for weight perturbation and the original local loss $\mathcal{L}^k_{\text{local}}$ for weight updates. To further analyze the relationship between loss functions and weight perturbation in SAM optimization, we conduct an ablation study on the following strategies.

- A_0: Use original local loss without any weight perturbation **(FedAvg)**.
- A_1: Use original local loss, but get the original local loss gradient at weights perturbed by the proximal gradient **(FedSoL)**.
- A_2: Use combined loss, but get the proximal loss gradient at weights perturbed by the proximal gradient.
- A_3: Use combined loss, but get the proximal loss gradient at weights perturbed by the proximal gradient.
- A_4: Use combined loss, but get the combined loss gradient at weights perturbed by the combined gradient.

- A_5: Use combined loss without any weight perturbation (**Proximal Restriction**).
- A_6: Use combined loss, but get the original local loss gradient at proximally perturbed weight loss.

We exclude the strategies that obtaining proximal loss at the perturbed weights using the original local loss gradient i.e., $\mathcal{L}_p(\boldsymbol{w}_k + \boldsymbol{\epsilon}_c^*)$, where $\boldsymbol{\epsilon}_c^* = \rho \frac{\boldsymbol{g}_p + \boldsymbol{g}_l}{\|\boldsymbol{g}_p + \boldsymbol{g}_l\|}$, as it leads the learning to diverge. The detailed formulation for each method is provided in Table 6 with its corresponding performance. The results in Table 6 demonstrates that utilizing the original local loss gradient at weights perturbed by the proximal loss gradient (A_1 in Table 6) yields outperforms the other approaches. We suggest that our FedSoL is an effective way to integrate proximal restriction effect into SAM optimization in FL.

Table 6: Detailed formulation for each method and their performance on CIFAR-10 datasets (LDA α=0.1).

Name	Method Formulation	Performance
A_0	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k)$	56.13
A_1	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*)$	**64.13**
A_2	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k) + \beta \cdot \mathcal{L}_p(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*)$	53.85
A_3	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*) + \beta \cdot \mathcal{L}_p(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*)$	60.28
A_4	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k + \boldsymbol{\epsilon}_c^*) + \beta \cdot \mathcal{L}_p(\boldsymbol{w}_k + \boldsymbol{\epsilon}_c^*)$	45.72
A_5	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k) + \mathcal{L}_p(\boldsymbol{w}_k)$	61.76
A_6	$\mathcal{L}_{\text{local}}(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*) + \beta \cdot \mathcal{L}_p(\boldsymbol{w}_k)$	44.12

G Adaptive Perturbation Radius

While SAM defines a fixed radius ρ, it is often insufficient in capturing the loss landscape dynamics [22, 26]. In FedSoL, we introduce an adaptive radius reflecting the global and local parameter discrepancies. For each layer m, we construct a scaling vector $\boldsymbol{\lambda}^{(m)}$, where the i-th entry corresponds to each parameter in that layer:

$$\boldsymbol{\lambda}^{(m)}[i] = \frac{\left|\boldsymbol{w}_k^{(m)}[i] - \boldsymbol{w}_g^{(m)}[i]\right|}{\left\|\boldsymbol{w}_k^{(m)} - \boldsymbol{w}_g^{(m)}\right\|_2}. \tag{11}$$

Here, $\boldsymbol{w}_g^{(m)}$ and $\boldsymbol{w}_k^{(m)}$ denote layer m in the global and local model respectively. The denominator represents the normalization of the discrepancy within the layer, accounting for the layer-wise scale variance. The adaptive radius allows more perturbation for the parameter with large difference, and vice versa. It fits with the typical behavior of proximal loss, which increases as $\|\boldsymbol{w}_k - \boldsymbol{w}_g\|_2$ grows. By concatenating these layer-specific vectors, $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)}, \dots, \boldsymbol{\lambda}^{(\text{last})})$, and incorporating it into the Equation (4)), the proximal perturbation $\boldsymbol{\epsilon}_p^*$ becomes:

$$\begin{aligned} \boldsymbol{\epsilon}_p^* = \;& \rho \cdot \boldsymbol{\Lambda} \odot \frac{\nabla_{\boldsymbol{w}_k} \mathcal{L}_p^k(\boldsymbol{w}_k; \boldsymbol{w}_g)}{\|\nabla_{\boldsymbol{w}_k} \mathcal{L}_p^k(\boldsymbol{w}_k; \boldsymbol{w}_g)\|_2} \\ \approx \;& \underset{\left\|\boldsymbol{\Lambda}^{-1} \odot \boldsymbol{\epsilon}\right\|_2 \leqslant \rho}{\operatorname{argmax}} \; \mathcal{L}_p^k(\boldsymbol{w}_k + \boldsymbol{\epsilon}; \boldsymbol{w}_g)\,, \end{aligned} \tag{12}$$

where $\odot$ denotes the element-wise product. Intuitively, the adaptive radius allows local learning to deviate certain parameters from the global model, only if they are crucial enough to withstand larger weight perturbations.

Adaptive Radius The advantage of the adaptive approach are depicted in Figure 8. As shown in Figure 8(a), using the adaptive radius not only improve performance but also reduces sensitivity to the selection of ρ. Meanwhile, Figure 8(b) displays the averaged λ values for each local model layer, highlighting the increased deviation in the later layers, as a consequence of the data heterogeneity [34]. Note that using a fixed value for ρ corresponds to setting $\boldsymbol{\Lambda}$ in Equation (12) as a vector with all entries to one.

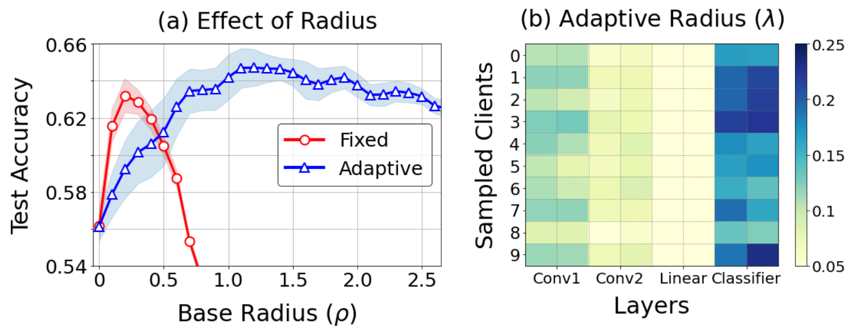

Figure 8: Effect of adaptive radius in CIFAR-10 (α=0.1). (a) Server test accuracy after 300 rounds. (b) Layer-wisely averaged λ values of FedSoL ($\rho = 1.0$) at round 200.

H Proximal Losses

In our primary experiments, we utilize KL-divergence loss as the proximal loss. However, FedSoL can be combine with various other proximal objectives. Table 7 shows the impact of incorporating FedSoL with other proximal restrictions: FedProx [32], FedNova [51], Scaffold [20], FedDyn [2], and Moon [30]. These methods are compared in two distinct scenarios: as an auxiliary objective alongside the original local objective (Base) and as proximal perturbation within FedSoL (Combined). The results show the enhanced performances with FedSoL.

Table 7: Comparison of proximal methods combined with FedSoL (ρ=2.0). The heterogeneity is set as LDA (α = 0.1).

Method	**CIFAR-10**		**SVHN**		**CINIC-10**	
	Base	Combined	Base	Combined	Base	Combined
FedProx	59.80	**63.93** ↑	72.40	**84.32** ↑	40.09	**55.25** ↑
FedNova	10.00	**31.77** ↑	53.07	**79.95** ↑	21.89	**42.37** ↑
Scaffold	10.00	**62.70** ↑	21.46	**77.52** ↑	16.89	**49.96** ↑
FedDyn	60.80	**62.85** ↑	78.15	**79.43** ↑	48.25	**52.17** ↑
MOON	55.72	**60.91** ↑	29.67	**76.82** ↑	38.15	**49.14** ↑

I Partial Perturbation

As the data heterogeneity does not affect all layers equally [34], we investigate the use of *partial* perturbation in FedSoL, by selectively perturbing specific layers instead of the entire model. The results in Table 8 reveal that perturbing only the last classifier layer (*Head* in Table 8) is sufficient for FedSoL. The performance is nearly as high as the full-model perturbation, but the computational requirement is significantly lower by avoiding multiple forward and backward computations across all layers when using the standard SAM strategy. Interestingly, perturbing all layers except the classifier head (*Body* in Table 8) consumes almost the same amount of computation but rather drops in performance, showing the importance of the later layers.

Table 8: Effect of partial weight perturbation in CIFAR10 (α=0.1). The FLOPs shows relative computation w.r.t. FedAvg. δ stands for the computation for the proximal loss.

Target Position	**Perturbation (ρ)**					**FLOPs**
	0.0	0.5	1.0	1.5	2.0	
All (*full*)		61.17	**64.16**	**64.38**	**63.94**	$2\times +\delta$
Body (*partial*)	56.13	60.98	62.95	63.94	63.80	$1.96\times +\delta$
Head (*partial*)		**62.65**	63.62	64.13	63.25	$1.33\times +\delta$

J Model Architecture

We conduct further experiments on different model architectures: VggNet-11 [45], ResNet-18 [15], and SL-ViT [28], which is a specialized structure of ViT [11] for small-sized datasets. The results provided in Table 9 validates the efficacy of FedSoL across varying model architectures.

Table 9: Comparison on different model architectures. The heterogeneity is LDA ($\alpha = 0.1$).

Model	Method	CIFAR-10	SVNH	PathMNIST
Vgg11	FedAvg	41.30$_{\pm 1.07}$	50.02$_{\pm 4.25}$	61.79$_{\pm 9.88}$
	FedProx	40.45$_{\pm 1.41}$	31.07$_{\pm 6.72}$	63.47$_{\pm 2.68}$
	FedNTD	**60.55**$_{\pm 2.14}$	56.62$_{\pm 2.64}$	69.82$_{\pm 2.27}$
	FedSoL	56.39$_{\pm 1.40}$	**74.74**$_{\pm 0.04}$	**78.38**$_{\pm 1.12}$
Res18	FedAvg	49.92$_{\pm 0.62}$	76.98$_{\pm 2.90}$	57.91$_{\pm 1.27}$
	FedProx	59.00$_{\pm 2.58}$	82.09$_{\pm 2.35}$	75.84$_{\pm 1.58}$
	FedNTD	57.79$_{\pm 3.42}$	78.50$_{\pm 0.18}$	76.87$_{\pm 0.57}$
	FedSoL	**66.32**$_{\pm 0.48}$	**85.97**$_{\pm 0.04}$	**80.59**$_{\pm 0.11}$
SL-ViT	FedAvg	35.48$_{\pm 2.09}$	53.94$_{\pm 5.17}$	72.44$_{\pm 1.91}$
	FedProx	38.73$_{\pm 1.23}$	58.25$_{\pm 4.23}$	74.10$_{\pm 1.23}$
	FedNTD	47.59$_{\pm 2.84}$	61.46$_{\pm 1.76}$	71.65$_{\pm 1.71}$
	FedSoL	**47.95**$_{\pm 1.51}$	**67.19**$_{\pm 0.33}$	**77.96**$_{\pm 0.47}$

K Effect on Local Learning

K.1 Proximal Restriction

We examine the proximal restriction effect in FedSoL by analyzing how its update gradient g_u interacts with the proximal loss $\mathcal{L}_p$, as shown in Figure 9. As ρ increases, g_u becomes more orthogonal to the proximal gradient g_p (Figure 9(b)). This orthogonality helps in maintaining the proximal loss low during the local learning process, which implies better global alignment (Figure 9(a)).

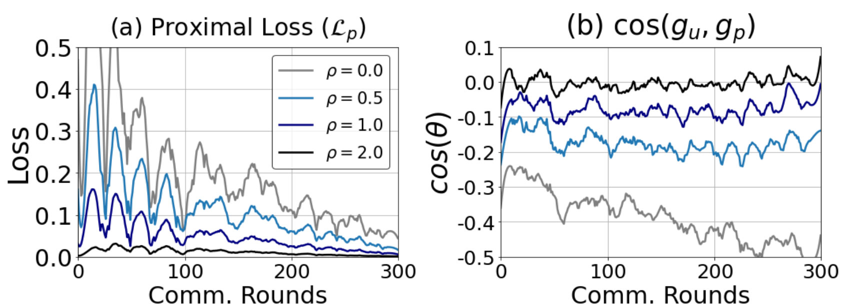

Figure 9: Effect of FedSoL on local learning in CIFAR-10 (α=0.1) by varying ρ values. (a) Average proximal loss of local models. (b) Cosine similarity between FedSoL gradient (g_u) and proximal gradient (g_p) during local learning.

K.2 Weight Divergence

To assess the deviation of local learning from the global model, we measure the L2 distance between models: $\|w_g - w_k\|$ where w_g is the global model and w_k is the client k's trained local model. The results, averaged across sampled clients, are shown in Figure 10. In Figure 10(a), FedSoL effectively reduces the divergence, ensuring that local models remain closely aligned with the global model, verifying the proximal restriction effect in FedSoL. Figure 10(b) illustrates that this alignment also fosters increased consistency among local models, reducing their mutual divergence.

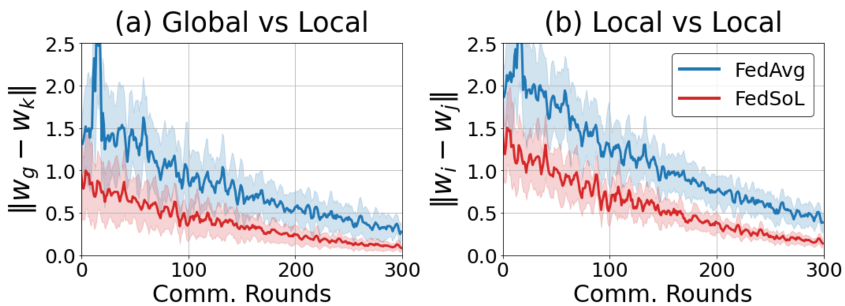

Figure 10: Comparative analysis of weight divergence in FedAvg and FedSoL (ρ=2.0) on CIFAR-10 LDA (α=0.1). (a) shows global-local model divergence, while (b) presents the divergence across local models.

L Proof of Proposition

We begin by organizing Equation (7), substituting Equation (6) into FedSoL:

$$\begin{aligned}\Delta^{\text{FedSoL}}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k) &\approx -\gamma\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{g}_u(\boldsymbol{w}_k)\rangle \\ &\approx -\gamma\left(\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{g}_l\rangle + \rho\left\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \nabla^2\mathcal{L}^k_{\text{local}}\,\hat{\boldsymbol{g}}_p\right\rangle\right),\end{aligned} \tag{13}$$

and for the update of $\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$ in FedAvg:

$$\Delta^{\text{FedAvg}}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) \approx -\gamma\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k(\boldsymbol{w}_k), \boldsymbol{g}_l\rangle = -\gamma\|\nabla\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2\,. \tag{14}$$

L.1 Proof of Proposition 1

Regarding Proposition 1, from Equation (13), we obtain:

$$\Delta^{\text{FedSoL}}\mathcal{L}^k_p \approx -\gamma\left(\langle\boldsymbol{g}_p, \boldsymbol{g}_l\rangle + \rho\langle\boldsymbol{g}_p, \nabla^2\mathcal{L}^k_{\text{local}}\,\hat{\boldsymbol{g}}_p\rangle\right) \approx -\gamma\Big(\langle\boldsymbol{g}_l\,, \boldsymbol{g}_p\rangle + \rho\cdot\hat{\boldsymbol{g}}_p^\top\nabla^2\mathcal{L}^k_{\text{local}}\,\boldsymbol{g}_p\Big)\,.$$

Furthermore, if the local objective is convex, then the Hessian is always positive semi-definite. Consequently, $\boldsymbol{g}_p^\top\,\nabla^2\mathcal{L}^k_{\text{local}}\;\boldsymbol{g}_p \geqslant 0$, and we can guarantee that the second term is nonnegative as well.

L.2 Proof of Proposition 2

Similarly, from Equation (13), we derive:

$$\begin{aligned}&\Delta^{\text{FedSoL}}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) \\ &\approx -\gamma\left(\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k), \nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\rangle + \rho\left\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k), \nabla^2_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}\,\hat{\boldsymbol{g}}_p\right\rangle\right) \\ &= -\gamma\left(\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \frac{\rho}{2}\left\langle\hat{\boldsymbol{g}}_p, 2\nabla^2_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\right\rangle\right)\,. \\ &= -\gamma\left(\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \nabla_{\boldsymbol{w}_k}\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2\cdot\frac{\rho}{2}\,\hat{\boldsymbol{g}}_p\right)\,.\end{aligned}$$

On the other hand, from Equation (14), applying the first-order Taylor approximation to $\boldsymbol{w}_k \mapsto \|\nabla\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2$, we have:

$$\begin{aligned}\Delta^{\text{FedAvg}}\mathcal{L}^k_{\text{local}}\left(\boldsymbol{w}_k + \frac{\rho}{2}\hat{\boldsymbol{g}}_p\right) &\approx -\gamma\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}\left(\boldsymbol{w}_k + \frac{\rho}{2}\hat{\boldsymbol{g}}_p\right)\|^2 \\ &\approx \gamma\left(\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \nabla_{\boldsymbol{w}_k}\left\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\right\|^2\cdot\frac{\rho}{2}\,\hat{\boldsymbol{g}}_p\right)\,.\end{aligned}$$

Therefore, $\Delta^{\text{FedAvg}}\mathcal{L}^k_{\text{local}}\,(\boldsymbol{w}_k + \frac{\rho}{2}\hat{\boldsymbol{g}}_p)$ is equal to $\Delta^{\text{FedSoL}}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$, up to the first-order Taylor approximation.

M Related Work

M.1 Federated Learning (FL)

Federated learning is a distributed learning paradigm to train models without directly accessing private client data [23, 24]. The standard algorithm, FedAvg [36], aggregates locally trained models by averaging their parameters. While a variety of FL algorithms have been introduced, they commonly conduct parameter averaging in a certain manner [20, 27, 32, 56]. Although FedAvg ideally performs well when all client devices are active and IID distributed [46, 53], its performance significantly degrades when clients have heterogeneous data distributions [19, 33, 57]. Our work focuses on mitigating this data heterogeneity issue by modifying the local learning strategy.

M.2 Proximal Restriction in FL

A prevalent strategy to address data heterogeneity in FL is the introduction of a proximal term into local learning objectives [20, 27, 30, 32]. This approach aims to restrict the local learning deviation induced by the biased local distributions. For example, FedProx [32] employs L_2 distance between models, while MOON [30] uses the contrastive loss [42], regarding the previously trained local model's representations as negative pairs. Meanwhile, Scaffold [20] use estimated global direction as a control variate to adjust local gradients. However, such explicit alteration of local objectives may hinder the acquisition of new knowledge during local learning [37, 44]. In our study, we aim to leverage the benefits of proximal restriction effect during local learning, but without changing the original local objectives.

M.3 SAM Optimization in FL

Recent studies have begun to suggest that enhancing local learning generality can significantly boost FL performance [4, 37, 44], aiding the global model in generalizing more effectively. Inspired by the latest findings that connect loss geometry to the generalization gap [5, 17, 18, 21], those works seek for *flat minima*, utilizing the recently proposed Sharpness-Aware Minimization (SAM) [13] as the local optimizer. For instance, FedSAM [44] and FedASAM [4] demonstrate the benefits of using SAM and its variants as local optimizer. Meanwhile, FedSMOO [47] incorporates a global-level SAM optimizer, and FedSpeed,[48] employs multiple gradient calculations to encourage global consistency. In our work, we introduce the proximal restriction effect into SAM in an implicit manner, by adjusting the perturbation direction and magnitude during local learning.