FedSOL: Stabilized Orthogonal Learning with Proximal Restrictions in Federated Learning

Supplementary Material

A. Table of Notations

Table 6. 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	Concentration 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
τ	Temperature to be divided in the softmax probability distribution
ρ	Perturbation Strength for SAM-related algorithms
$\mathbf{\Lambda}$	Vector consists with scaling parameters for perturbation vector in SAM-related algorithms
Weights:	
$\boldsymbol{w}_g$	Weight of the global server model
$\boldsymbol{w}_k$	Weight of the k-th client model
$\|\boldsymbol{w}_g - \boldsymbol{w}_k\|$	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 objective for the k-th client

B. Experimental Setups

The code is implemented by PyTorch [54]. The overall code structure is based on FedML [21] library with some modifications for simplicity. We use 2 A6000 GPU cards, but without Multi-GPU training.

B.1. Model Architecture

In our primary experiments, we use the model architecture used in FedAvg [46], which consists of two convolutional layers with subsequent max-pooling layers, and two fully-connected layers. The same model architecture is also used in [33, 36, 43]. We also conduct further experiments on ResNet-18 [22], Vgg-11 [58], and SL-ViT [34]. For SL-ViT, we resize 28×28-sized images into 32×32 to accommodate the required minimum size for the patch embedding.

B.2. Datasets

To validate our approach, 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** [15] (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** [32] (50,000 / 10,000): contains a labeled subset of 80 Million Tiny Images [62] for 10 different classes. The data is augmented using Random Cropping, Horizontal Flipping, Normalization, and Cutout [16].
- **SVHN** [51] (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** [12] (90,000 / 90,000): is a combination of CIFAR and downsized ImageNet [14], 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** [71] (110,000 / 7,180): contains non-overlapping patches from Hematoxylin & Eosin stained colorectal histology slide images. The data is augmented using Random Horizontal Flipping, and Normalization.
- **TissueMNIST** [71] (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 RGB images.

Note that we evaluate our algorithm is on medical imaging datasets - a crucial practical application of federated learning [3, 72]. Illustrative examples of the images are illustrated in Figure 9.

B.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**

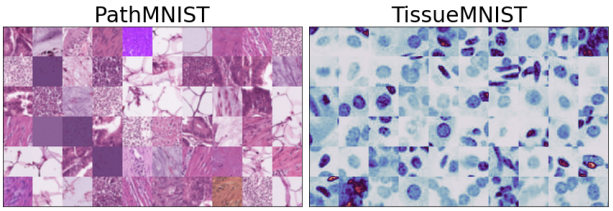


Figure 9. Example images from PathMNIST datasets and TissueMNIST datasets.

and (ii) **Latent Dirichlet Allocation** (LDA).

- **(i) Sharding** [33, 46, 52]: 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)** [36, 43, 63]: 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.

B.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 7.

B.5. Algorithm Implementation Details

We search for hyperparameters and select the best among the candidates. The hyperparameters for each method is provided in Table 8. In the primary experiments, we use KL-divergence loss [23] with softened logits with temperature τ=3 for the proximal loss for the adversarial weight perturbation in FedSOL.

Table 7. 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 8. 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. Learning Curves

To provide further insights into the learning process, we illustrate the learning curves of different FL methods in Figure 10. 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 [33] underperforms compared to FedProx [38] on the TissueMNIST datasets), FedSOL consistently exhibits significant improvements when compared to the other baselines.

D. Personalized Performance

In Table 9, we compare FedSOL with several methods specifically designed for personalized federated learning (pFL): PerFedAvg [18], FedBabu [52], and kNN-Per [44]. Each method is assessed by fine-tuning them for e local

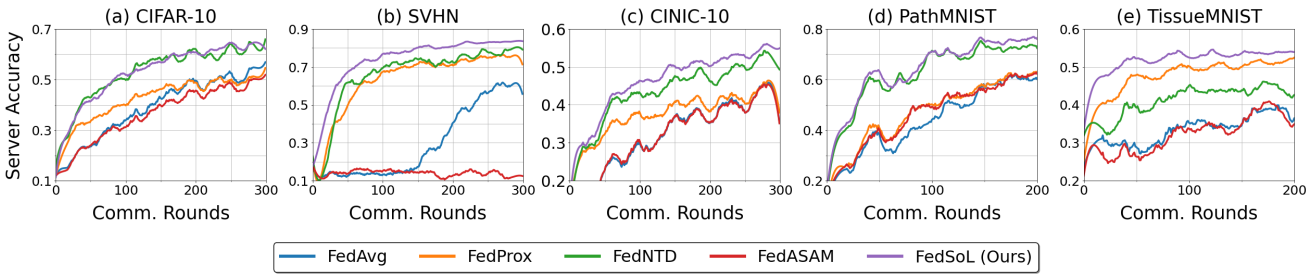


Figure 10. Learning curves of FL methods on LDA (α=0.1). The curves are smoothed for clear visualization.

epochs from the global model after the final communication round. As global alignment is unnecessary for the personalized model, we fine-tune FedSOL using 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 9. 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 $\pm$12.8	87.4 $\pm$13.0	82.4 $\pm$15.5
FedAvg	1	84.1 $\pm$13.4	86.6 $\pm$15.5	82.2 $\pm$17.5
	5	88.9 $\pm$8.9	92.1 $\pm$5.7	89.2 $\pm$10.1
PerFedAvg	1	80.5 $\pm$16.2	64.1 $\pm$30.3	82.3 $\pm$18.9
	5	86.3 $\pm$10.4	72.4 $\pm$21.2	88.8 $\pm$10.2
FedBabu	1	84.6 $\pm$12.7	88.7 $\pm$9.6	85.7 $\pm$14.3
	5	89.2 $\pm$8.4	92.7 $\pm$6.2	90.5 $\pm$8.8
kNN-Per	1	85.7 $\pm$12.3	86.4 $\pm$15.0	86.5 $\pm$14.2
	5	89.7 $\pm$8.1	92.8 $\pm$6.2	91.4 $\pm$7.5
FedSoL-FT (Ours)	1	**87.5** $\pm$9.7	**92.5** $\pm$7.4	**88.1** $\pm$12.2
	5	**90.5** $\pm$7.8	**95.0** $\pm$3.9	**91.6** $\pm$6.9

E. Performance on Larger Datasets

In Table 2, we show that FedSOL consistently achieves performance gains, while most existing FL (Federated Learning) methods are sensitive to learning setups. Although FedNTD marginally outperforms FedSOL in some CIFAR-10 cases, it significantly falls behind in most others. In Table 10, we conducted experiments on the CIFAR-100 and ImageNet-100 datasets. We used the ResNet-18 model, distributing each dataset across 100 clients with a sampling ratio of 0.1 and optimized for 5 local epochs. Our observations indicate that FedSOL maintains its effectiveness in both datasets, whereas FedNTD's performance decreases in CIFAR-100.

Table 10. Test Accurac on CIFAR-100 and ImageNet-100.

Method	CIFAR-100			ImageNet-100
	s = 5	s = 10	α = 0.05	α = 0.1
FedAvg	42.43	53.23	48.69	43.41
FedProx	39.03 $\downarrow$	48.38 $\downarrow$	46.75 $\downarrow$	34.49 $\downarrow$
FedNTD	39.32 $\downarrow$	52.23 $\downarrow$	48.35 $\downarrow$	44.08 $\uparrow$
FedSOL	**44.21** $\uparrow$	**53.78** $\uparrow$	**49.25** $\uparrow$	**44.97** $\uparrow$

F. Head Perturbation in Larger Models

To validate the effectiveness of partial perturbation strategy, we extended the comparison experiment in Table 5 to Table 11. The results on VggNet-11 and ResNet-18 indicate that perturbing only the last classifier layer (*head*) is almost as effective as perturbing the entire model (*full*), saving significant computational cost.

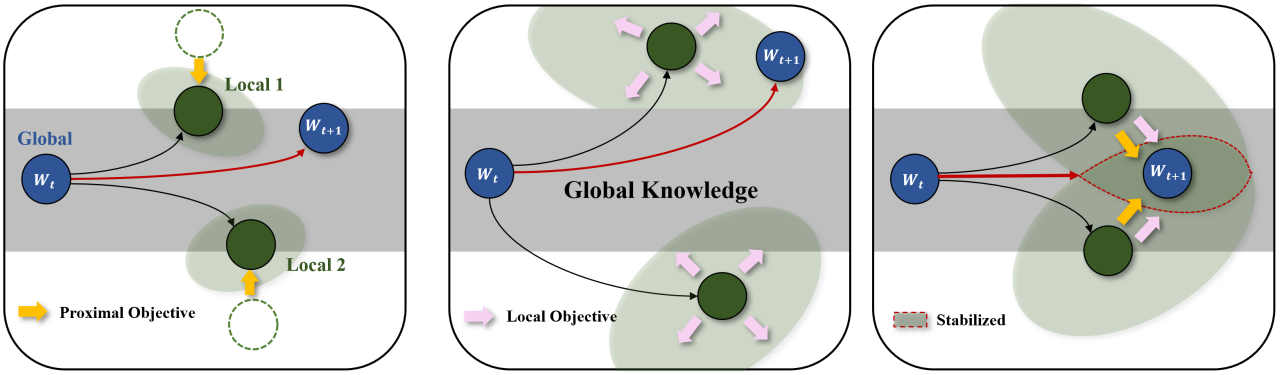
Table 11. Effect of partial perturbation on CIFAR-10 (α=0.1).

Model	FedAvg	FedSOL (*full*)	FedSOL (*head*)
VggNet-11	41.30	56.44 **(+15.14)**	56.39 **(+15.09)**
ResNet-18	49.92	66.69 **(+16.77)**	66.32 **(+16.04)**

G. Comparison to the Sharpness-Aware Optimization

G.1. SAM Optimization in FL

Recent studies have begun to suggest that enhancing local learning generality can significantly boost FL performance [6, 47, 55], aiding the global model in generalizing more effectively. Inspired by the latest findings that connect loss geometry to the generalization gap [8, 24, 26, 29], those



(a) Global Knowledge Preservation (b) Local Knowledge Acquisition (c) FedSOL (Ours)

Figure 11. An overview of FL scenarios. **Gray-shaded** areas represent global knowledge, while **Green-shaded** areas represent local knowledge. (a) Learning with the proximal objective achieves global knowledge preservation but limits local knowledge acquisition. (b) Learning with the local objective effectively acquires local knowledge but results in forgetting global knowledge. In (c), the orthogonal learning strategy of FedSOL stabilizes the learning process by resolving conflicts between these two objectives.

works seek for *flat minima*, utilizing the recently proposed Sharpness-Aware Minimization (SAM) [20] as the local optimizer. For instance, FedSAM [55] and FedASAM [6] demonstrate the benefits of using SAM and its variants as local optimizer. Meanwhile, FedSMOO [60] incorporates a global-level SAM optimizer, and FedSpeed,[61] employs multiple gradient calculations to encourage global consistency. By improving local generality, these approaches mitigate the conflicts between individual local objectives, contributing to the overall smoothness of the aggregated global model [6, 55, 60].

G.2. Limitations of using SAM in FL

Although these approaches have demonstrated competitive performance without proximal restrictions, their ability to generalize effectively within their respective local distributions does not necessarily guarantee the preservation of previous global knowledge during local learning. This is due to the inherent conflict between global and local objectives. In our FedSOL, we introduce the use of proximal perturbation as a means to achieve an orthogonal local gradient, which does not contribute to an increase in proximal loss. This approach can be understood as implicitly incorporating the effect of proximal restriction into SAM, achieved by adjusting the perturbation's direction and magnitude during local learning. It is also worth mentioning that the effect of proximal perturbation depends on the relationship between the local and proximal objectives. In the extreme case where the local objective is identical to the proximal objective, our FedSOL collapses into the original SAM.

In Figure 11, we illustrate a conceptual overview of the global and local knowledge trade-off in FL. In Figure 11(a) and Figure 11(b), learning on one objective undermines the effect of the other. However, the orthogonal learning of FedSOL stabilizes the local learning by tackling the conflicts between the two objectives (Figure 11(c)).

H. Other Perturbation Strategies

In our work, we propose the use of proximal perturbation as our primary strategy. This section compares various perturbation strategies and discusses the effectiveness of using the proximal gradient for weight perturbation. A straightforward approach would involve using a linear combination of the local objective, which includes both the local loss $\mathcal{L}^k_{\text{local}}$ and the proximal loss $\mathcal{L}^k_{\text{p}}$ in Equation 2. This combination is applied in SAM-like optimization (Equation 4) as follows:

$$\min_{\boldsymbol{w}_k} \max_{\|\boldsymbol{\epsilon}\|_2<\rho} \left[\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) + \beta \cdot \mathcal{L}^k_{p}(\boldsymbol{w}_k; \boldsymbol{w}_g)\right]. \tag{13}$$

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 preserves global knowledge from proximal objective nor effectively acquires the local knowledge 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 local loss $\mathcal{L}^k_{\text{local}}$ for weight updates. To further analyze the relationship between loss functions and weight perturbation in FedSOL optimization, we conduct an ablation study on the following strategies.

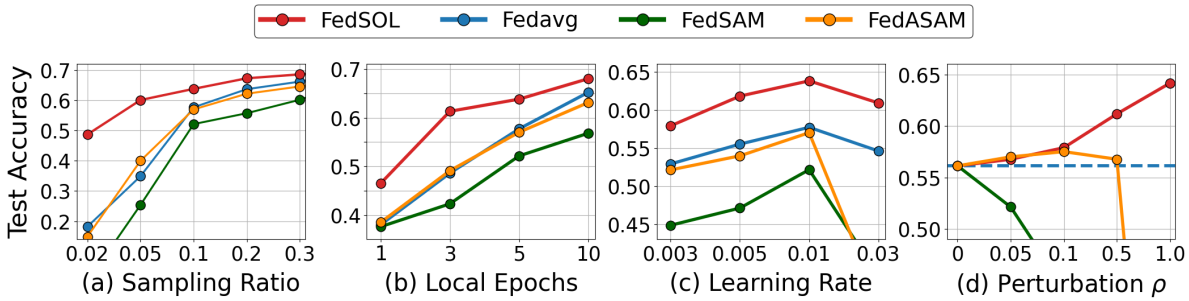


Figure 12. Performance on CIFAR-10 (α=0.1) datasets.

- A_0: Use local loss without any weight perturbation **(FedAvg)**.
- A_1: Use local loss, but get the 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 local loss gradient at proximally perturbed weight loss.

We exclude the strategies that obtaining proximal loss at the perturbed weights using the 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 12 with its corresponding performance. The results in Table 12 demonstrates that utilizing the local loss gradient at weights perturbed by the proximal loss gradient (A_1 in Table 12) yields outperforms the other approaches. We suggest that our FedSOL is an effective way to integrate proximal restriction effect into SAM optimization in FL.

I. Client-side Computational Cost

In Table 5, we analyze the FLOPS required for FedSOL and note that perturbing only the head part is almost as effective as full perturbation, yet it requires only 33% more computation than FedAvg. Although FedSOL necessitates backward computation twice, this does not lead to increased GPU memory usage, as FedSOL does not store these gradients simultaneously. The slight increase in memory usage ($< 0.01\%$) arises from calculating the adaptive perturbation strength for each layer. In Table 13, we present the latency for a single local step with the ImageNet-100 dataset, using a ResNet-18 model on an NVIDIA A6000 GPU. We measured FedSOL's latency using an L2 proximal loss for a fair comparison. While FedSOL involves additional local computation, we emphasize that in FL, the energy consumed for communication typically surpasses that for computation. The faster convergence of FedSOL substantially reduces the total energy consumption. In the ImageNet-100 experiment detailed in Table 10, for example, FedSOL achieves FedAvg's 300-round performance by round 238.

Table 12. 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

Table 13. Measured latency for a single local step.

Usage	FedAvg	FedProx	FedNTD	FedSAM	FedSOL
Latency	2.846 s	2.865 s	2.996 s	3.380 s	2.900 s

J. Effect of Learning Factors on FedSAM

Figure 12 presents the impact of learning factors on FedSAM [55] and FedASAM [6], showing that both methods are more sensitive to these factors compared to FedSOL. In most of our experiments, SAM-related FL methods shows inferior performance compared to FedAvg. This may be because SAM, aiming to enhance local model generality, becomes less effective for small models, under conditions of high data heterogeneity, or with a low sampling ratio. Future research is expected to identify the conditions where employing SAM on the local side becomes beneficial.

K. Proof of Proposition

The notion of $\approx$ in the main papers for (Especially for Equation 10 and Equation 9) are supported by Taylor's theorem. We will use the following formulation for C^2 functions. All matrix norms are the largest singular value.

Theorem 1 (Taylor's theorem) *If f is C^2 function at the open ball contains $\boldsymbol{w}$ and $\boldsymbol{w}+\boldsymbol{v}$, we have:*

$$f(\boldsymbol{w}+\boldsymbol{v}) = f(\boldsymbol{w}) + \langle \nabla f(\boldsymbol{w}),\ \boldsymbol{v}\rangle + R_f(\boldsymbol{v};\boldsymbol{w})\,,\ \textit{where}\ |R_f(\boldsymbol{v};\boldsymbol{w})| \leq \frac{1}{2}\|\boldsymbol{v}\|^2 \max_{t\in[0,1]} \|\nabla^2 f(\boldsymbol{w}+t\boldsymbol{v})\|_2\,.$$

For $R(\boldsymbol{v};\boldsymbol{w})$, there are two well-known representations:

- *There exists $t \in (0,1)$ such that $R_f(\boldsymbol{v};\boldsymbol{w}) = \frac{1}{2}\boldsymbol{v}^\top \nabla^2 f(\boldsymbol{w}+t\boldsymbol{v})\boldsymbol{v}$,*
- $R_f(\boldsymbol{v};\boldsymbol{w}) = \int_0^1 (1-t)\boldsymbol{v}^\top \nabla^2 f(\boldsymbol{w}+t\boldsymbol{v})\,\boldsymbol{v}\,dt$.

Now, We first precise the notion of $\approx$ in the Equation 10:

$$\begin{aligned}\Delta^{\text{FedSOL}}\mathcal{L}^k(\boldsymbol{w}_k) &= \mathcal{L}^k\big(\boldsymbol{w}_k - \gamma\, \boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k)\big) - \mathcal{L}^k(\boldsymbol{w}_k) \\ &= -\gamma\langle \nabla_{\boldsymbol{w}_k}\mathcal{L}^k(\boldsymbol{w}_k),\ \boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k)\rangle + R_{\mathcal{L}^k}(-\gamma\, \boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k);\boldsymbol{w}_k)\,, \end{aligned} \tag{14}$$

and Equation 9:

$$\begin{aligned}\boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k) &= \nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k + \boldsymbol{\epsilon}_p^*) \\ &= \boldsymbol{g}_l(\boldsymbol{w}_k) + \rho\,\nabla^2_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k) + \boldsymbol{R}_{\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}}(\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k);\boldsymbol{w}_k)\,, \end{aligned} \tag{15}$$

where $\boldsymbol{R}_{\nabla\mathcal{L}^k_{\text{local}}}(\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k);\boldsymbol{w}_k)$ is a vector, where the i-th value is $R_{\partial_i\mathcal{L}^k_{\text{local}}}(\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k);\boldsymbol{w}_k)$, which is the residual term with i-th directional derivative $\partial_i\mathcal{L}^k_{\text{local}}$.

By substituting the above expression for $\boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k)$ into $\Delta^{\text{FedSOL}}\mathcal{L}^k(\boldsymbol{w}_k)$, we have:

$$\begin{aligned}&\Delta^{\text{FedSOL}}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k) = -\gamma\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{g}_u(\boldsymbol{w}_k)\rangle + R_{\mathcal{L}^k_{\{\text{local},p\}}}(-\gamma\,\boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k);\boldsymbol{w}_k)\,, \\ &= -\gamma\left(\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{g}_l\rangle + \rho\,\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \nabla^2\mathcal{L}^k_{\text{local}}\,\hat{\boldsymbol{g}}_p\rangle\right) + \\ &\underbrace{-\gamma\langle\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{R}_{\nabla\mathcal{L}^k_{\text{local}}}(\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k);\boldsymbol{w}_k)\rangle + R_{\mathcal{L}^k_{\{\text{local},p\}}}(-\gamma\,\boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k);\boldsymbol{w}_k)}_{\mathcal{E}^k_{\{\text{local},p\}}}\,,\end{aligned}$$

where $\mathcal{E}^k_{\{\text{local},p\}}$ is the total residual term:

$$\mathcal{E}^k_{\{\text{local},p\}} = -\gamma\sum_i \partial_i\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k)R_{\partial_i\mathcal{L}^k_{\text{local}}}(\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k);\boldsymbol{w}_k) + R_{\mathcal{L}^k_{\{\text{local},p\}}}(-\gamma\,\boldsymbol{g}_u^{\text{FedSOL}}(\boldsymbol{w}_k);\boldsymbol{w}_k)\,.$$

For representing the magnitude of residual term effectively, we will assume three constants. It is important to note that the constants can be made smaller by concentrating on the optimization-relevant region rather than the entire weight space.

Assumption 1 *Matrix norm of Hessian and the norm of gradient for $\mathcal{L}^k_{\text{local}}$ are bounded:*

$$D^k = \sup_{\boldsymbol{w}}\|\nabla^2\mathcal{L}^k_{\text{local}}(\boldsymbol{w})\| < \infty\,,\ B^k = \sup_{\boldsymbol{w}_k}\|\nabla_{\boldsymbol{w}_k}\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\| < \infty\,.$$

Assumption 2 *For any linear path connecting $\boldsymbol{w}$ and $\boldsymbol{v}$ with length ρ, we define the following coefficient:*

$$C^k_{\rho,\{\text{local},p\}} = \sup_{\boldsymbol{w},\boldsymbol{v},\|\boldsymbol{w}-\boldsymbol{v}\|=\rho}\ \max_{t\in[0,1]}\left\|\sum_i \partial_i\mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w})\nabla^2\partial_i\mathcal{L}^k_{\text{local}}(\boldsymbol{w}+t(\boldsymbol{v}-\boldsymbol{w}))\right\| < \infty\,.$$

The first term becomes:

$$-\gamma\rho^2 \int_0^1 (1-t)\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k)^\top \left(\sum_i \partial_i \mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k) \nabla^2 \partial_i \mathcal{L}^k_{\text{local}}(\boldsymbol{w} + t\rho\hat{\boldsymbol{g}}_p(\boldsymbol{w}_k)) \right) \hat{\boldsymbol{g}}_p(\boldsymbol{w}_k)\, dt\,,$$

and we can easily see that magnitude of this term can be bounded with $\frac{1}{2}\gamma\rho^2 C^k_{\rho,\{\text{local},p\}}$, by Theorem 1. By same procedure, it is easy to see the second term is bounded by $\frac{1}{2}\gamma^2 D^k (B^k)^2$. Consequently, we can conclude:

$$\Delta^{\text{FedSOL}} \mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k) = \gamma \left(\left\langle \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \boldsymbol{g}_l \right\rangle + \rho \left\langle \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\{\text{local},p\}}(\boldsymbol{w}_k), \nabla^2 \mathcal{L}^k_{\text{local}}\, \hat{\boldsymbol{g}}_p \right\rangle \right) + \mathcal{E}_{\{\text{local},p\}}\,, \tag{16}$$

where $\mathcal{E}_{\{\text{local},p\}}$ has magnitude:

$$\mathcal{E}^k_{\{\text{local},p\}} = \mathcal{O}\left(\gamma\rho^2 C^k_{\rho,\{\text{local},p\}} + \gamma^2 D^k (B^k)^2 \right).$$

Remark. In this analysis section, to align with the adaptive perturbation scheme utilized in real experiments, we cautiously suggest considering the constant strength as $\rho = 2.0/\sqrt{\#\text{ params}}$ in terms of order. In the FedSOL algorithm, adaptive perturbation strength is employed, as detailed in Section 3.3. Here, from Equation 7, the squared sum of perturbation strength is ρ^2, substantially lower than in scenarios assuming a constant strength ρ (where the squared sum would be $\rho^2 \times (\#\text{ params})$). For simplicity, our analysis primarily focuses on scenarios with constant perturbation strength. Consequently, within this analysis, the effective perturbation strength should be considerably lower than the experimental setting of $\rho = 2.0$ in FedSOL. That is, it should be treated as the order of $2.0/\sqrt{\#\text{ params}}$ to match the parameter-wise squared sum of perturbation strengths.

K.1. Proof of Proposition 1

Regarding Proposition 1, from Equation 16, we obtain:

$$\begin{aligned}
\Delta^{\text{FedSOL}} \mathcal{L}^k_p &= -\gamma \left(\langle \boldsymbol{g}_p, \boldsymbol{g}_l \rangle + \rho \langle \boldsymbol{g}_p, \nabla^2 \mathcal{L}^k_{\text{local}}\, \hat{\boldsymbol{g}}_p \rangle \right) + \mathcal{E}^k_p \\
&= -\gamma \Big(\langle \boldsymbol{g}_l\,, \boldsymbol{g}_p \rangle + \rho \cdot \hat{\boldsymbol{g}}_p^\top \nabla^2 \mathcal{L}^k_{\text{local}}\, \boldsymbol{g}_p \Big) + \mathcal{O}\Big(\gamma\rho^2 C^k_{\rho,p} + \gamma^2 D^k (B^k)^2 \Big)\,.
\end{aligned}$$

Furthermore, if the $\nabla^2 \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$ is positive semi-definite, $\boldsymbol{g}_p^\top\, \nabla^2 \mathcal{L}^k_{\text{local}}\, \boldsymbol{g}_p \geq 0$, and we can guarantee that the second term is nonnegative as well.

K.2. Proof of Proposition 2

Similarly, from Equation 16, we derive:

$$\begin{aligned}
\Delta^{\text{FedSOL}} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) &= -\gamma \left(\langle \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k), \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) \rangle + \rho \langle \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k), \nabla^2_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}\, \hat{\boldsymbol{g}}_p \rangle \right) + \mathcal{E}^k_{\text{local}} \\
&= -\gamma \left(\|\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \frac{\rho}{2} \langle \hat{\boldsymbol{g}}_p, \underline{2\nabla^2_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}} \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)} \rangle \right) + \mathcal{O}\Big(\gamma\rho^2 C^k_{\rho,\text{local}} + \gamma^2 D^k (B^k)^2 \Big) \\
&= -\gamma \left(\|\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \underline{\nabla_{\boldsymbol{w}_k} \|\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2} \cdot \frac{\rho}{2}\, \hat{\boldsymbol{g}}_p \right) + \mathcal{O}\Big(\gamma\rho^2 C^k_{\rho,\text{local}} + \gamma^2 D^k (B^k)^2 \Big)
\end{aligned}$$

For the FedAvg update of $\mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$, as derived in Equation 14, we have:

$$\begin{aligned}
\Delta^{\text{FedAvg}} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) &= -\gamma \langle \nabla_{\boldsymbol{w}_k} \mathcal{L}^k(\boldsymbol{w}_k), \boldsymbol{g}_l \rangle + R_{\mathcal{L}^k_{\text{local}}}(-\gamma\, \boldsymbol{g}_l(\boldsymbol{w}_k)\,; \boldsymbol{w}_k) \\
&= -\gamma \|\nabla \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \mathcal{O}\big(\gamma^2 D^k (B^k)^2\big)\,.
\end{aligned} \tag{17}$$

On the other hand, from Equation 17, applying the first-order Taylor approximation to $\boldsymbol{w}_k \mapsto \|\nabla \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2$, we have:

$$\begin{aligned}
&\Delta^{\text{FedAvg}} \mathcal{L}^k_{\text{local}} \left(\boldsymbol{w}_k + \frac{\rho}{2} \hat{\boldsymbol{g}}_p \right) = -\gamma \left\| \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}} \left(\boldsymbol{w}_k + \frac{\rho}{2} \hat{\boldsymbol{g}}_p \right) \right\|^2 + \mathcal{O}(\gamma^2 D^k (B^k)^2) \\
&= \gamma \left(\|\nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)\|^2 + \nabla_{\boldsymbol{w}_k} \left\| \nabla_{\boldsymbol{w}_k} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k) \right\|^2 \cdot \frac{\rho}{2}\, \hat{\boldsymbol{g}}_p \right) + \mathcal{O}\big(\gamma\rho^2 C^k_{0,\text{local}} + \gamma\rho^2 (D^k)^2 + \gamma^2 D^k (B^k)^2 \big)\,.
\end{aligned}$$

Therefore, $\Delta^{\text{FedAvg}} \mathcal{L}^k_{\text{local}}\, (\boldsymbol{w}_k + \frac{\rho}{2} \hat{\boldsymbol{g}}_p)$ is equivalent with $\Delta^{\text{FedSOL}} \mathcal{L}^k_{\text{local}}(\boldsymbol{w}_k)$, up to order:

$$\mathcal{O}\Big(\gamma\rho^2 C^k_{\rho,\text{local}} + \gamma\rho^2 C^k_{0,\text{local}} + \gamma\rho^2 (D^k)^2 + \gamma^2 D^k (B^k)^2 \Big)\,.$$

L. Toy Example for Explaining FedSOL

Consider a two-dimensional weight space, $\mathbb{R}^2$, and denote the weights as $(u, v) \in \mathbb{R}^2$. We define the local loss function $\mathcal{L}_{\text{local}}$ and the proximal loss function $\mathcal{L}_p$. The local minimum is represented as (u_l, v_l), and the aggregated server weight is denoted as $(0, 0)$. The most intuitive form of losses that can be considered is:

$$\mathcal{L}_{\text{local}}(u, v) = \frac{1}{2}(u - u_l)^2 + \frac{\delta}{2}(v - v_l)^2 \text{ and } \mathcal{L}_p(u, v) = \frac{\mu}{2}(u^2 + v^2)\,,$$

where δ is a positive coefficient. A value of δ indicates that the local objective is more influenced by the variable u than by the direction v if $\delta < 1$ and vice versa. δ can be also interpreted as the proportion of one class over another in binary classification.
Since $\epsilon_p^* = \frac{(u,v)}{\sqrt{u^2+v^2}}$, the FedSOL gradient on local side becomes:

$$g^{\text{FedSOL}} = \left((u - u_l) + \rho\frac{u}{\sqrt{u^2 + v^2}}\,, \delta \cdot \left((v - v_l) + \rho\frac{v}{\sqrt{u^2 + v^2}}\right)\right)\,,$$

while with the adding proximal loss term (like FedProx), we have an overall local gradient:

$$g^{\text{FedProx}} = ((u - u_l) + \mu \cdot u, \delta \cdot (v - v_l) + \mu \cdot v)\,.$$

When considering proximal loss, proximal regularization is applied in the u and v directions without considering local loss, which can be sub-optimal. The sub-optimality can be summarized in the sense of global alignment and local learnablity and this can be resolved with FedSOL, which uses local gradient update in the proximal-loss-sensitive point, while not harming local learnability as Equation 2. This argument strengthen the discussion for effectiveness of reducing negative inner product in Equation 1.
The equilibrium point in both algorithms is defined where the gradient equals zero, and in the current setting, it is unique. We investigate each algorithm on this setting as follows:

Proximal Regularization (FedProx)

Analysis: In FedProx, the learned weight (u^*, v^*) is calculated as $\left(\frac{1}{1+\mu}u_l, \frac{\delta}{\delta+\mu}v_l\right)$. This reflects the influence of both the strength of normalization and the local curvature. Furthermore, the resulting vector from $(0, 0)$ to (u^*, v^*) is biased towards the direction with greater curvature compared to (u_l, v_l).
Implication: While this directional bias may not pose significant issues in local learning, it becomes problematic in FL where weights from different clients are averaged. Since each client may have a different δ, the result of normalization influenced by local client loss might not be optimal in an FL context.

FedSOL

Analysis: FedSOL introduces a normalization that maintains the same direction but reduces the vector magnitude as ρ.

Proof: Let us consider the polar coordinate of (u^*, v^*) and (u_l, v_l) as (r^*, θ^*) and (r_l, θ_l) respectively. Furthermore, (u^*, v^*) satisfies:

$$(u^* - u_l, v^* - v_l) = \left(-\rho\frac{u^*}{\sqrt{(u^*)^2 + (v^*)^2}}, -\rho\frac{v^*}{\sqrt{(u^*)^2 + (v^*)^2}}\right)\,.$$

Then $\theta^* = \theta_l = \theta$ is ensured by $\frac{v_l}{u_l} = \frac{v^*}{u^*}$. Now, the above equation gives $r^* = r_l - \rho$. □
Benefit: This implies that aggregation can occur appropriately in FedSOL. Even with diverse clients, the gradient without normalization is simply scaled, thus preserving the effectiveness of aggregation.

Conclusion

The approach of FedSOL in handling proximal loss demonstrates significant advantages over FedProx within the Federated Learning context. By aligning weight adjustments parallel to the local optimum, FedSOL ensures more efficient and effective aggregation among diverse clients. In contrast, FedProx exhibits a bias towards greater curvature and, as a result, potentially leads to sub-optimal global alignment, as seen with the variable impacts of δ on different clients. The simulation in Figure 13

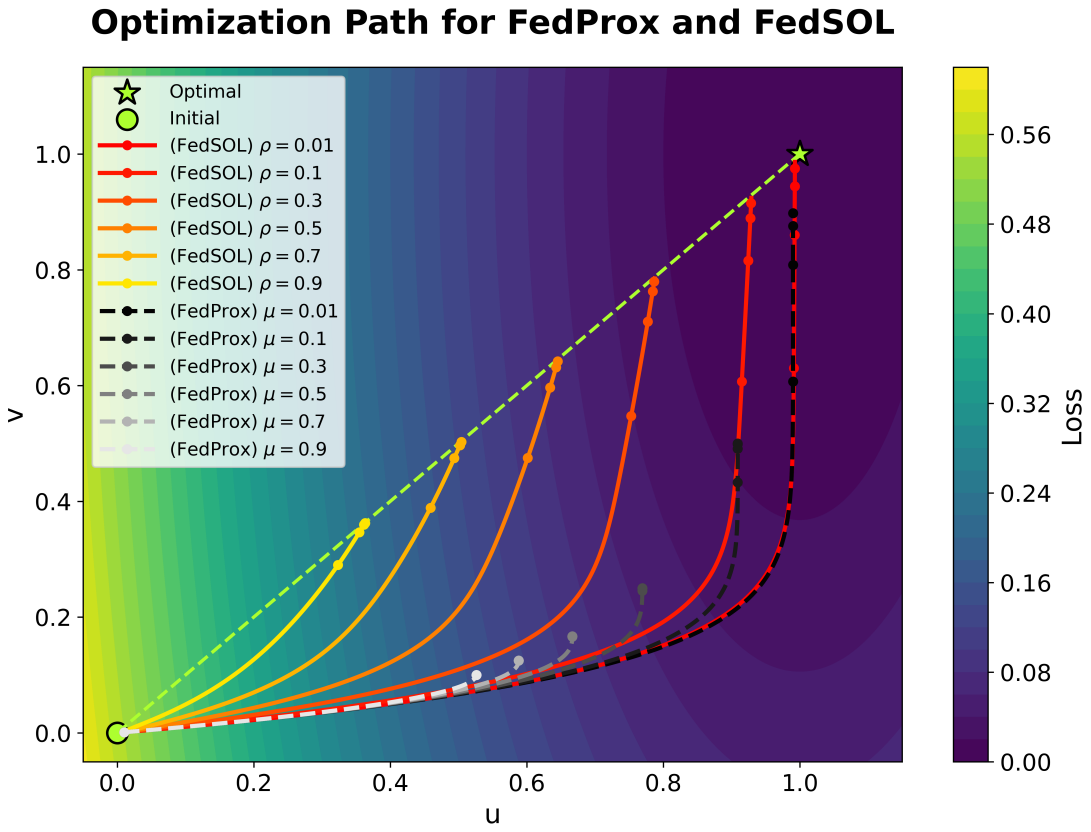


Figure 13. Gradient update paths for FedSOL and FedProx in a toy example on $\delta = 0.1$, over 4000 epochs with learning rate 0.01, and initial weights $(u_l, v_l) = (1, 1)$. The illustration highlights the distinct trajectories and convergence points of each algorithm, underlining their differing approaches.

corroborates our analysis, showing that the optimized points for FedSOL align with the lime line connecting the initial and optimal points. Meanwhile, FedProx's optimized points are skewed towards the u-axis, highlighting the theoretical distinctions between the two algorithms.