LATENT INFORMATION SHARING FOR ACCELERATING FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is a communication-efficient distributed learning paradigm. However, client drift remains one of the most critical challenges, hindering the efficient training of a global model. In this study, we propose a novel latent information sharing scheme that directly mitigates data heterogeneity across clients. Our theoretical and empirical results show that sharing a small amount of hidden-layer activations significantly improves training efficiency while preserving convergence guarantees and data privacy. Furthermore, we compare our method with existing FL approaches designed to address client drift, including FedProx, SCAFFOLD, FedPVR, FedProto, and SplitFed, and demonstrate superior model accuracy under a fixed round budget without incurring excessive communication overhead. Overall, this work presents a promising new knowledge aggregation scheme and provides a comprehensive analysis of the impact of activation sharing on federated optimization.

1 INTRODUCTION

While federated learning (FL) enables privacy-preserving and communication-efficient aggregation across multiple data holders, referred to as clients, its training efficiency largely depends on the degree of data heterogeneity among local datasets. Severe data heterogeneity leads to client drift, one of the most well-known challenges in FL, which significantly hinders convergence. To enable practical large-scale FL applications, it is crucial to address client drift without sacrificing the benefits of communication-efficient distributed learning.

To mitigate data heterogeneity, prior works such as FedProx Li et al. (2020) and FedDyn Durmus et al. (2021) introduce proximal regularization to stabilize local updates. While these regularization approaches alleviate client drift to some extent, they do not fundamentally resolve discrepancies across clients, as they operate independently without explicit coordination. Other studies address data heterogeneity by reducing the stochastic variance of gradients, such as SCAFFOLD Karimireddy et al. (2020), FedVARP Jhunjhunwala et al. (2022), and FedMDVR Zhang et al. (2025). Although these methods effectively alleviate client drift, they often incur substantial memory and communication overhead, limiting their practicality in large-scale deployments. FedLAMA Lee et al. (2023) proposes a structural approach based on layer-wise adaptive aggregation to implicitly mitigate client discrepancy. However, it still relies on parameter-space coordination and does not explicitly align intermediate representations across clients. These limitations motivate exploring alternative forms of collaboration beyond parameter-space coordination.

In this study, we propose *Activation Sharing Federated Learning* (ASFL), a latent information sharing approach that tackles client drift from a novel representation-based perspective. Building upon the conventional model aggregation scheme, our approach additionally exchanges intermediate activations from a critical hidden layer. By directly mitigating data heterogeneity through this exchange, our method accelerates convergence, thereby achieving higher model accuracy within a fixed round budget. Specifically, our study addresses two key questions: (i) which layer generates the most informative latent representations for exchange and (ii) how can we effectively leverage remote latent information without incurring prohibitive memory / communication overhead? Furthermore, we provide a theoretical analysis of the impact of activation sharing on the convergence properties of FedAvg for non-convex and smooth objective functions. To bridge the gap between theory and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

practice, we also discuss how to exploit activation sharing with minimal additional communication cost by reusing remote activations multiple times.

Our extensive empirical study substantiates the practicality of activation sharing, demonstrating its superiority not only in convergence speed but also in system efficiency. On representative benchmarks, including CIFAR-10/100 Krizhevsky et al. (2009), FEMNIST Caldas et al. (2018), and SVHN Netzer et al. (2011), under non-IID settings, ASFL consistently achieves higher accuracy than state-of-the-art FL methods, including FedProx Li et al. (2020), SCAFFOLD Karimireddy et al. (2020), FedPVR Li et al. (2023), FedProto Tan et al. (2022), FedImpro Tang et al. (2024), and SplitFed Thapa et al. (2022). These results indicate that ASFL effectively mitigates client drift by allowing each client to learn from the latent representations of other clients, thereby achieving higher accuracy within a fixed round budget. In addition, our wall-clock time analysis shows that ASFL reduces the per-round training time by up to 33% compared to the other FL methods under the same number of local steps per round. Finally, we empirically show that, under model inversion attacks, ASFL exhibits privacy leakage comparable to that of existing representation-sharing methods such as SplitFed.

2 BACKGROUND

Federated optimization – We consider federated minimization problems defined as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x}) \right], \tag{1}$$

where m is the number of local models and $F_i(\mathbf{x}) = \mathbb{E}_{\xi_i\sim D_i}[F_i(\mathbf{x},\xi_i)]$ is the local objective function associated with local data distribution D_i. FedAvg McMahan et al. (2017) is a pioneering distributed learning method designed for federated optimization, utilizing local SGD with periodic model aggregations. It has been widely adopted to train global models without explicitly sharing decentralized data. However, it suffers from slow convergence when data heterogeneity is high.

Client drift – Client drift is a well-known phenomenon in FL. In federated optimization, each client performs local updates using only its own data, which are often drawn from heterogeneous (non-IID) distributions. As a result, local updates become biased with respect to the global objective, leading to a mismatch between the expected local gradient and the true global gradient:

$$\mathbb{E}[g_t^i] = \nabla F_i(\mathbf{x}_t^i) \neq \nabla F(\mathbf{x}_t^i), \tag{2}$$

where g_t^i, $\nabla F_i(\mathbf{x}_t^i)$, and $\nabla F(\mathbf{x}_t^i)$ denote the stochastic gradient, the local full-batch gradient, and the global full-batch gradient, respectively. This deviation becomes more pronounced as the degree of data heterogeneity increases or the number of local update steps grows, ultimately slowing down the convergence of the global model. To mitigate this issue, a variety of approaches have been proposed in the literature such as FedProx, SCAFFOLD, and SILVER Oko et al. (2024).

3 RELATED WORK

Variance reduction methods – Recent federated optimization methods mitigate client drift using variance reduction techniques, including FedProx, SCAFFOLD, FedPVR, FedVARP, and SILVER. Although effective, these methods typically introduce additional memory and communication overhead to maintain auxiliary states such as control or tracking variables, which becomes increasingly burdensome for large-scale models. FedProto Tan et al. (2022) shares class-wise prototypes computed by averaging embeddings within each class and regularizes local training toward the aggregated global prototypes to reduce cross-client discrepancies in class representations. MOON Li et al. (2021) reduces client drift using a model-contrastive objective, at the cost of additional memory and computation overhead for maintaining auxiliary models and computing contrastive losses. In this work, we compare ASFL with these representative variance-reduced FL methods in terms of both model accuracy under a fixed round budget and communication overhead.

Split learning methods – Split Federated Learning (SFL) integrates Split Learning (SL) Vepakomma et al. (2018) with Federated Learning (FL) to alleviate the computational burden on the client side. The most prominent difference between SFL and conventional FL is that

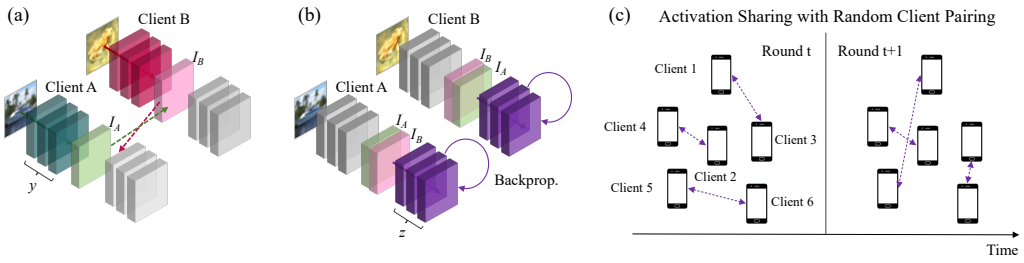


Figure 1: A schematic illustration of latent information sharing. Figure (a) shows the *Latent Information Sharing* step. The activation outputs from $\mathbf{y}$, I_A and I_B, are exchanged between the two clients. Figure (b) shows the *Tail Round* in which the two clients run backpropagation only within $\mathbf{z}$ using both I_A and I_B. Figure (c) shows the random client pairing-based activation sharing.

SFL transmits intermediate activations from a specific 'cut layer' to a central server. The server then executes the forward and backward passes for the remaining layers. This split training mechanism not only reduces client-side computational overhead but also mitigates client drift, as the server-side sub-model is updated using aggregated knowledge from all participating clients. While SplitFed aggregates latent representations on the server and trains a subset of network layers there, our proposed method requires neither data centralization across clients nor server-side computation. We will also compare our proposed activation sharing method with SplitFed in terms of model accuracy and system efficiency.

4 METHOD

4.1 LATENT INFORMATION SHARING

In this section, we describe how latent information is shared across clients via two steps, *Activation Sharing* and *Tail Round*. Figure 1 provides a schematic illustration of this process.

Activation sharing – We define *Latent Information* as the activation output of a specific hidden layer in a neural network. This information inherently captures local data patterns. Therefore, we expect that sharing it across clients will alleviate data heterogeneity. First, a given neural network x is partitioned into two consecutive groups of layers, denoted as $\mathbf{x} := \{\mathbf{y}, \mathbf{z}\}$, where $\mathbf{y}$ consists of the input-side layers and $\mathbf{z}$ consists of the remaining output-side layers.

Then, we define the latent information I as $\tau'/2$ activation output batches from y:

$$I_i := \{\mathbf{y}(\mathcal{B}_{i,t})\}_{t=1}^{\tau'/2}, \tag{3}$$

where τ' is the number of local steps with remote activations, $\{\mathcal{B}_{i,t}\}_{t=1}^{\tau'/2}$ denotes $\tau'/2$ mini-batches randomly sampled from the local dataset of client i. Once I_i is ready, randomly paired clients exchange their I_i and corresponding labels through inter-process communications. We call this communication *Activation Sharing*.

Tail round – Once client i receives I_j $(i \neq j)$, it performs local training using the combined set $I_i \cup I_j$, updating only the subnetwork $\mathbf{z}$. Specifically, client i trains $\mathbf{z}$ for τ' iterations using mini-batches randomly sampled from $I_i \cup I_j$. Since I consists of the activation outputs from the last layer of $\mathbf{y}$, both I_i and I_j can be directly fed into $\mathbf{z}$. Backpropagation is restricted to the layers within $\mathbf{z}$. We refer to these steps as a *tail round*.

Algorithm 1 presents the pseudocode for the *Latent Information Sharing* method. Each client i first computes I_i using its local data and shares it with a randomly selected client j (Line 3, *activation sharing*). Subsequently, each client i trains $\mathbf{z}$ for τ' iterations using activations sampled from $I_i \cup I_j$ (Lines 5–6, *tail round*).

Algorithm 2 outlines FedAvg integrated with the proposed *Latent Information Sharing* method. At the beginning of each communication round, the model is first updated via Algorithm 1 (Line 4). Consequently, each round consists of τ' tail update steps followed by τ full update steps.

Algorithm 1 Activation Sharing

Require: τ': the total number of local update steps within a tail round.
1: Partition x into y and z.
2: Each client $i \in [m]$ calculates $I_i = \mathbf{y}(\mathcal{B}_{i,t})_{t=1}^{\tau'/2}$.
3: $j \leftarrow$ a random client, $j \neq i$.
4: Each client i sends I_i to and receives I_j from client j.
5: **for** $t \rightarrow \tau'$ **do**
6: $\quad \hat{\mathcal{B}} \leftarrow \text{Random}_b(I_i \cup I_j)$.
7: $\quad \mathbf{z}_{i,t+1} = \mathbf{z}_{i,t} - \eta \nabla f(\mathbf{z}, \hat{\mathcal{B}})$
8: **end for**
9: Return $\mathbf{x} = \{\mathbf{y}, \mathbf{z}_{i,\tau'}\}$.

Algorithm 2 FedAvg with Activation Sharing

Require: τ: the total number of local updates within a communication round, τ': the total number of local updates within a tail round, R: the total number of communication rounds, $\mathcal{N}$: the total list of clients, a: the number of active clients per round.
1: **for** $r \rightarrow R$ **do**
2: $\quad \mathcal{A} \leftarrow \text{Random}_a(\mathcal{N})$
3: $\quad$ Server sends global model $\mathbf{x}$ to clients in $\mathcal{A}$.
4: $\quad \mathbf{x}_{i,0} \leftarrow$ Latent Information Sharing$(\mathbf{x})$ (Algorithm 1)
5: $\quad$ **for** $k \rightarrow \tau$ **do**
6: $\quad\quad$ Each client $i \in \mathcal{A}$ gets a mini-batch $\mathcal{B}_{i,k}$ from its local dataset.
7: $\quad\quad \mathbf{x}_{i,k+1} = \mathbf{x}_{i,k} - \eta \nabla f(\mathbf{x}_{i,k}, \mathcal{B}_{i,k})$
8: $\quad$ **end for**
9: $\quad$ Server aggregates the local models $\mathbf{x}_{i,\tau}, i \in \mathcal{A}$ to $\mathbf{x}$.
10: **end for**
11: Return $\mathbf{x}$

4.2 Controlling the degree of information sharing

Activation Sharing is governed by two critical factors that determine its impact on local training: (i) which layer's latent information is shared, and (ii) the volume of activation outputs being shared.

4.2.1 Selecting the sharing layer via cost-variance trade-off

The tail round exposes the sub-model $\mathbf{z}$ to latent information from other clients. These remote activations can be utilized to update $\mathbf{z}$, thereby reducing the variance of stochastic gradients. Sharing activations at earlier layers allows a larger portion of the model to benefit from this variance reduction. However, this incurs higher computational overhead, as backpropagation must traverse more layers during the tail rounds. To address this trade-off, we propose a practical layer selection scheme that balances variance reduction and system efficiency.

Step 1. System efficiency curve – Let k denote the partition index that splits the model into two sub-models: $\mathbf{y}$, comprising layers 1 to $k-1$, and $\mathbf{z}$, comprising layers k to L (where L is the total number of layers). The computational cost of the backward pass through layer i is defined as:

$$\mathcal{C}_{\text{comp}}(i) := N_{\text{out}}^i \cdot N_{\text{conn}}^i,$$
$$\hat{\mathcal{C}}_{\text{comp}}(k) = \sum_{i=k}^{L} \mathcal{C}_{\text{comp}}(i),$$

where N_{out}^i is the number of output activations at layer i, and N_{conn}^i is the number of input connections per output unit. Since backpropagation proceeds from layer L down to k, the total cost is accumulated over the layers in $\mathbf{z}$.

The additional communication cost of activation sharing is quantified as:

$$\mathcal{C}_{\text{comm}}(k) := N_{\text{out}}^{k-1}, \tag{4}$$

where N_{out}^{k-1} denotes the number of activations at the boundary layer $k-1$. To integrate these measures, we normalize them separately and compute their product to represent the total system

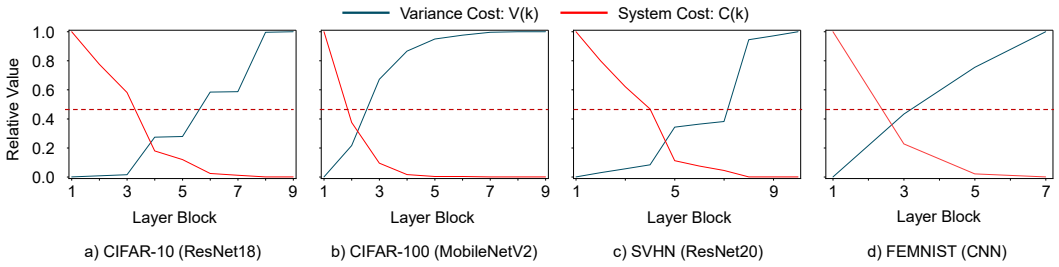


Figure 2: Layer selection results. The green curve represents the cumulative variance of the input-side sub-model y, computed using Eq. 6, while the red curve represents the system efficiency computed using Eq. 5. The red horizontal dashed line indicates the midpoint of the efficiency range. Across various benchmarks, the intersection of the two curves indicates a balanced trade-off between system efficiency and variance reduction.

overhead for a given k:

$$\mathcal{C}(k) = \tilde{\mathcal{C}}_{\text{comp}}(k) \cdot \tilde{\mathcal{C}}_{\text{comm}}(k), \tag{5}$$

where $\tilde{\mathcal{C}}_{\text{comp}}(k)$ and $\tilde{\mathcal{C}}_{\text{comm}}(k)$ are the normalized computation and communication costs, respectively. Notably, $\mathcal{C}(k)$ monotonically decreases as k increases from 1 to L.

Step 2. Variance reduction curve – We quantify the potential benefit of variance reduction by varying k as follows:

$$\mathcal{V}(k) := \sum_{i=1}^{k-1} \text{Var}(g^i), \tag{6}$$

where g^i denotes the mini-batch gradient of the loss function with respect to the parameters of layer i. In practice, we approximate the true gradient variance using the empirical variance of gradients within each mini-batch.

Step 3. Finding the intersection – Finally, we identify the optimal layer k where the normalized $\mathcal{C}(k)$ and $\mathcal{V}(k)$ curves intersect. This "sweet spot" represents a configuration that maximizes the number of layers benefiting from variance reduction while maintaining overall system efficiency. Figure 2 illustrates these $\mathcal{C}(\cdot)$ and $\mathcal{V}(\cdot)$ curves across four popular machine learning benchmarks.

4.2.2 Communication-efficient activation sharing

Depending on the input dimensionality and layer connectivity, intermediate activations can incur substantial communication overhead. To mitigate this cost, we propose a practical scheme that exchanges a minimal subset of activations and reuses them across multiple iterations within each communication round.

In Algorithm 1, the size of I_j is $B \cdot \tau'/2$, where B is the mini-batch size. Then, the tail round updates $\mathbf{z}$ using random mini-batches drawn from $I_i \cup I_j$ (lines 6–7). Instead, we exchange only a small subset of activations (with $|I_i| > |I_j|$) and construct an augmented activation pool $\mathcal{P}$ by randomly sampling $B \cdot \tau'/2$ samples from I_j with replacement such that:

$$\mathcal{P} = I_i \cup \left(\bigcup^{B.\tau'/2} \xi \sim \text{Uniform}(I_j) \right), \tag{7}$$

where ξ denotes the activations randomly drawn from I_j. By sampling mini-batches from $\mathcal{P}$, we effectively oversample I_j, allowing its activations to be reused multiple times within each communication round, thereby improving training efficiency. In addition, we sample ξ for $B \cdot \tau'/2$ times so that the contributions of local and remote latent information become balanced within the augmented pool. This balanced representation enables the model to effectively incorporate newly received remote knowledge without being dominated by the larger local dataset. We conduct an ablation study to evaluate the impact of this activation sharing strategy in terms of both system efficiency and model accuracy.

4.3 CONVERGENCE ANALYSIS

We show that ASFL guarantees convergence for non-convex and smooth optimization problems:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x}),$$

where m denotes the total number of clients, and $F_i(\mathbf{x}) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[f(\mathbf{x};\xi_i)]$ represents the local objective function associated with the data distribution $\mathcal{D}_i$ of client i. Here, $f(\mathbf{x};\xi_i)$ is the loss function evaluated on a sample ξ_i drawn from $\mathcal{D}_i$. In each communication round, the fraction of remote data in the total data consumed is denoted by α.

Our analysis relies on the following assumptions, which hold for all $i \in [m]$.

Assumption 1 (Lipschitz continuity). $\|\nabla F_i(\mathbf{a}) - \nabla F_i(\mathbf{b})\| \leq \mathcal{L}\|\mathbf{a}-\mathbf{b}\|$ *for some* $\mathcal{L} > 0$ *and all* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$.

Assumption 2 (Unbiased local gradients). $\mathbb{E}_{\xi_i\sim D_i}[\nabla f(\mathbf{x},\xi_i)] = \nabla F_i(\mathbf{x})$.

Assumption 3 (Bounded local and global variances). $\mathbb{E}_{\xi_i\sim D_i}\|\nabla f(\mathbf{x},\xi_i) - \nabla F_i(\mathbf{x})\|^2 \leq \sigma_x^2$ *and* $\mathbb{E}\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \sigma_G^2$ *for some* $\sigma_x, \sigma_G > 0$.

Assumption 4 (Bounded local variance w.r.t. a partial model). $\mathbb{E}_{\xi_i\sim D_i}\|\nabla f(\mathbf{z},\xi_i) - \nabla F_i(\mathbf{z})\|^2 \leq \sigma_z^2$ *for some* $\sigma_z > 0$.

Then, we present our convergence analysis results. See the Appendix for proofs.

Lemma 1 (framework). *Under assumption 1 ∼ 4, if the learning rate* $\eta \leq \min\{\frac{1}{\tau\mathcal{L}}, \frac{1}{4\tau'\mathcal{L}}\}$, *the loss reduction of a single communication round is bounded as follows.*

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\ + \frac{3\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ + \frac{2\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2.\end{aligned}$$

Lemma 2 (partial model discrepancy). *For any step-size satisfying* $\eta \leq \frac{1}{2\sqrt{6}L\tau'}$, *we have:*

$$\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] \leq 2\eta^2\tau'\sigma_z^2 + 8\tau'\eta^2\left(\alpha(1-\alpha)+2\tau'\right)\sigma_G^2 + 16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2].$$

Theorem 3. *Under assumption* $1 \sim 4$, *if* $\eta \leq \min\{\frac{1}{12\mathcal{L}\tau}, \frac{1}{8\mathcal{L}\sqrt{\tau/(\tau')^3}}\}$, *the federated optimization with activation sharing guarantees the following condition.*

$$\begin{aligned}\sum_{t=0}^{T-1}\|\nabla F(\mathbf{x}_t)\|^2 \leq \frac{4}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\ + 4\eta\mathcal{L}T\left(3\eta\mathcal{L}+\frac{1}{a}\right)\sigma_x^2 + 4\tau'\eta\mathcal{L}T\left(\eta\mathcal{L}+\frac{4}{\tau}+\frac{1}{a\tau}\right)\sigma_z^2 \\ + \frac{8\eta\tau'\mathcal{L}T}{\tau}\left(2\eta\mathcal{L}\tau'(\alpha(1-\alpha)+2\tau') + \frac{1}{a}(1+\alpha-2\alpha^2)\right)\sigma_G^2.\end{aligned}$$

Remark 1 (Convergence and diminishing step size). *The above result guarantees convergence to a stationary point. In particular, due to the global variance which never goes to 0, the algorithm converges to a neighborhood of a stationary point, where the size of the neighborhood is determined by the variance terms. Furthermore, when a diminishing stepsize is used such that* $\eta \to 0$ *and* $\sum_{t=0}^{T-1}\eta = \infty$, *all variance-dependent terms vanish asymptotically, and we obtain*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{x}_t)\|^2 = 0, \tag{8}$$

which establishes convergence to a stationary point. This result is consistent with the convergence behavior of vanilla FedAvg.

Remark 2 (Trade-off between local and global variance)**.** *In the tail round, stochastic noise in the local update of* $\mathbf{y}$ *is eliminated as its updates are dropped, resulting in a smaller local variance term, i.e.,* $\sigma_z < \sigma_x$*. Moreover, incorporating remote activations may further reduce* σ_z *via an averaging effect. As* τ' *increases under a fixed local computation budget, the local variance contribution decreases while the global variance induced by shared activations increases (maximized at* $\alpha = 1/2$*). Empirically, the reduction in local variance dominates, leading to improved overall performance.*

Remark 3 (Linear speedup)**.** *With a sufficiently small step size and an appropriate number of local updates, ASFL achieves linear speedup with respect to the number of clients* m*. In particular, if*

$$\eta = \Theta\left(\frac{\sqrt{m}}{\sqrt{T}}\right), \quad \tau' = O(\tau^{1/3}), \tag{9}$$

then the convergence rate becomes $O(1/\sqrt{mT})$*, matching the standard linear speedup achieved in distributed optimization. Note that our analysis considers the partial client participation. However, the results can be readily extended to the full participation setting with* m *clients by replacing random client sampling with deterministic full participation.*

4.4 Differential privacy for activation sharing

Although ASFL does not share raw data, exchanging latent representations $I_i = \{\mathbf{y}(\mathcal{B}_{i,t})\}_{t=1}^{\tau'/2}$ between paired clients may still leak sensitive information. To mitigate this risk, one can consider incorporating differential privacy (DP) to ASFL, following prior work Thapa et al. (2022). Specifically, we adopt PixelDP Lecuyer et al. (2018) by inserting a noise layer into $\mathbf{y}$, so that the receiving module $\mathbf{z}$ operates on privatized activations rather than the raw outputs of $\mathbf{y}$. By the *post-processing immunity* property of DP Dwork & Roth (2014), any downstream computation on the perturbed activations preserves the same privacy guarantee.

The sensitivity of the subnetwork $\mathbf{y}$, which governs the required noise magnitude, is defined as:

$$\triangle_{p,q}^{\mathbf{y}} = \max_{\xi \neq \xi'} \frac{\left\|\mathbf{y}(\xi) - \mathbf{y}(\xi')\right\|_q}{\|\xi - \xi'\|_p},$$

where ξ and ξ' are two input samples of the corresponding client, and p and q are the degrees of input and output norms. In addition, R defines the radius of the threat model such that any adversarial perturbation confined within $B_p(R) := \{\delta' \in \mathbb{R}^n : \|\delta'\|_p \leq R\}$ is guaranteed to be masked by the injected noise and provides (ϵ, δ)-DP guarantees.

We consider two standard mechanisms to privatize the shared activations. The Laplace mechanism achieves $(\epsilon, 0)$-DP by adding noise scaled to $\triangle^{\mathbf{y}} p, q/\epsilon$, while the Gaussian mechanism ensures (ϵ, δ)-DP with variance $\sigma^2 \propto (\triangle^{\mathbf{y}} p, q/\epsilon)^2$. In both cases, the transmitted privatized activations satisfy differential privacy w.r.t. the sender's local data.

5 Experiments

Experimental settings – We evaluate ASFL on four benchmarks: CIFAR-10 (ResNet18 He et al. (2016)), CIFAR-100 (MobileNetV2 Sandler et al. (2018)), SVHN (ResNet20), and FEMNIST (CNN). Experiments involve 100 total clients, with 10 randomly activated per round. For non-IID settings, we use a Dirichlet distribution with $\alpha = 0.1$. Our framework is based on PyTorch 2.7.0 and runs on a server with two NVIDIA RTX 4070 GPUs. See Appendix A.1 for detailed settings.

5.1 Comparative study

FL benchmarks – Table 1 compares ASFL with SOTA FL methods, where PixelDP is not applied to the transmitted activations to isolate the effect of activation sharing on optimization efficiency. Drift-mitigation methods, including SCAFFOLD, FedProto, MOON, FedPVR, and FedVARP, improve accuracy over FedAvg on most benchmarks, but SCAFFOLD nearly doubles communication cost through control variate aggregation, and the other four offer relatively limited gains. SplitFed achieves higher accuracy by exchanging activations at every update step, but at the cost of substantially increased communication cost. In contrast, ASFL attains higher accuracy across all benchmarks while introducing only marginal additional communication cost. E.g., on CIFAR-10, ASFL

Table 1: Accuracy and communication cost comparison across datasets.

Method	CIFAR-10		CIFAR-100		SVHN		FEMNIST	
	Acc. (%) ↑	Comm (MB) ↓	Acc. ↑	Comm ↓	Acc. ↑	Comm ↓	Acc. ↑	Comm ↓
FedAvg	67.70 ± 0.33	42.63	48.31 ± 0.68	9.20	87.13 ± 0.16	1.04	83.54 ± 0.19	1.54
FedProx	67.91 ± 0.53	42.63	47.61 ± 0.27	9.20	88.27 ± 0.90	1.04	83.23 ± 0.29	1.54
FedProto	69.57 ± 1.58	42.65	48.08 ± 0.87	9.69	87.99 ± 0.15	1.04	84.11 ± 0.23	1.60
MOON	69.35 ± 0.83	42.63	46.49 ± 0.21	9.20	86.10 ± 0.83	1.04	84.03 ± 0.12	1.54
SCAFFOLD	71.93 ± 1.14	85.26	48.53 ± 0.34	18.40	87.36 ± 0.37	2.08	84.13 ± 0.20	3.08
FedPVR	69.40 ± 0.97	42.65	48.48 ± 0.70	9.69	87.92 ± 0.47	1.04	83.57 ± 0.26	1.60
FedVARP	69.24 ± 1.43	42.63	48.71 ± 0.55	9.20	87.92 ± 0.40	1.04	82.88 ± 0.25	1.54
FedImpro	62.80 ± 0.41	45.13	47.65 ± 0.66	21.70	85.83 ± 1.64	1.67	83.74 ± 0.24	3.02
SplitFed	68.16 ± 0.85	298.63	48.42 ± 0.40	329.2	86.92 ± 0.05	41.04	83.67 ± 0.14	4.41
ASFL (Ours)	$\mathbf{75.27 \pm 0.88}$	50.63	$\mathbf{50.56 \pm 0.32}$	13.20	$\mathbf{89.10 \pm 0.31}$	2.04	$\mathbf{84.85 \pm 0.33}$	1.78

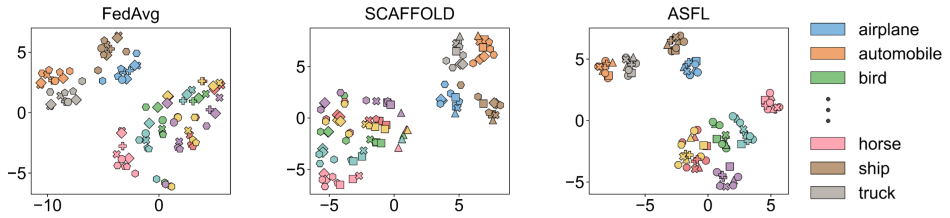


Figure 3: t-SNE visualization of per-client class centroids based on penultimate features from 10 clients on CIFAR-10. Colors denote classes and marker shapes indicate clients.

improves accuracy by approximately 7.5% over FedAvg while exchanging 8 MB more activations per round. Extra experimental analyses supporting these findings are provided in Appendix A.5.

t-SNE analysis – Figure 3 provides a t-SNE visualization of per-client class centroids in the latent space. Under FedAvg, representations of the same class are widely scattered, whereas SCAFFOLD shows only limited concentration. In contrast, ASFL forms noticeably tighter and more coherent clusters across different clients, suggesting superior cross-client representation alignment. Overall, these results demonstrate that activation sharing effectively alleviates data heterogeneity by promoting consistent representations, directly leading to the higher validation accuracy shown in Table 1.

5.2 Per-client round wall-clock time

We compare the per-client round wall-clock time for local computation and report the results in Table 2, normalized to FedAvg on each dataset. All methods use the same total local update budget per communication round (See Appendix A.1.1). ASFL divides this budget equally between tail-round updates and standard local updates ($\tau' = \tau$), whereas methods without a tail round devote the entire budget to standard local training. We synchronize CUDA at both the start and the end of each timed region. For ASFL, the measured local computation time includes both the training loop and the forward passes used to generate shared activations, which we time separately and sum.

Table 2: Per-client round wall-clock time for local computation, normalized to FedAvg on each dataset. Bars share a 0–4× scale, and vertical ticks mark the FedAvg baseline (1×).

Method	CIFAR-10	CIFAR-100	SVHN	FEMNIST
FedProto	1.02×	1.04×	1.04×	1.26×
MOON	1.38×	1.34×	1.32×	1.84×
SCAFFOLD	2.06×	1.77×	1.43×	2.27×
FedPVR	1.00×	1.04×	1.01×	1.10×
FedVARP	1.07×	1.02×	1.03×	1.18×
FedImpro	1.38×	1.92×	1.17×	3.67×
SplitFed	1.00×	1.00×	1.02×	1.19×
ASFL (Ours)	**0.69×**	**0.77×**	**0.67×**	**0.94×**

ASFL achieves the lowest local computation wall-clock time on all four datasets under the same total local update budget, even after including shared activation generation. The tail round contributes to

Table 3: Performance of ASFL with Pixel DP in accuracy (%). The radius of threat model, $R = 0.1$.

Noise type	Noise level	CIFAR-10	CIFAR-100	SVHN	FEMNIST
Laplace	$\epsilon = 2.0$	74.69 ± 0.97	49.09 ± 0.59	88.49 ± 0.34	85.84 ± 0.19
	$\epsilon = 1.0$	74.42 ± 0.97	47.85 ± 0.58	88.88 ± 0.40	85.76 ± 0.19
	$\epsilon = 0.5$	74.14 ± 0.47	47.30 ± 0.82	88.80 ± 0.38	85.29 ± 0.19
	$\epsilon = 0.2$	69.83 ± 0.97	45.11 ± 1.02	87.07 ± 0.41	84.77 ± 0.31
Gaussian	$\delta = 0.05$	73.02 ± 0.84	46.18 ± 0.39	87.88 ± 0.59	85.27 ± 0.08
	$\delta = 0.01$	71.07 ± 0.64	45.56 ± 1.02	86.67 ± 0.80	84.88 ± 0.30
	$\delta = 0.001$	68.55 ± 0.70	45.19 ± 0.05	85.95 ± 0.81	84.96 ± 0.08

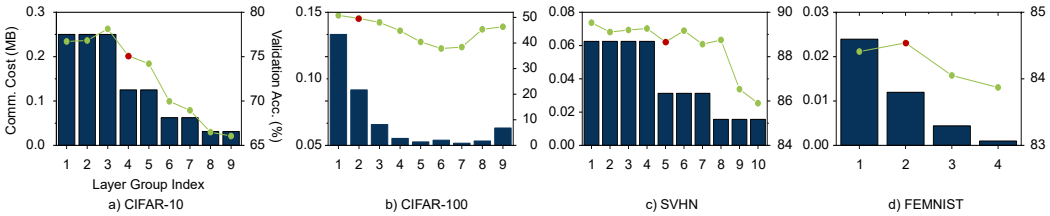


Figure 4: Ablation study on selection of activation sharing layers. The left and right y-axes represent the communication cost and the validation accuracy, respectively. The x-axis denotes the layer index.

this reduction through partial training: its updates train only part of the model, reducing backward-pass computation per update. Thus, allocating part of the fixed local update budget to the tail round provides a computational advantage while preserving the total number of local updates.

5.3 ASFL WITH DIFFERENTIAL PRIVACY

We evaluate the privacy–utility trade-off of ASFL with PixelDP under Laplace and Gaussian noise across varying privacy budgets (Table 3). As expected, stronger privacy guarantees (e.g., smaller ϵ for Laplace or smaller δ for Gaussian) lead to larger injected noise, which in turn degrades model performance. However, even under moderate noise levels, ASFL continues to outperform other FL methods shown in Table 1 that do not even employ differential privacy techniques. This highlights that ASFL provides a superior privacy-utility trade-off, making it a highly practical strategy for privacy-preserving FL. More extensive empirical results are provided in Appendix A.4.

5.4 ABLATION STUDY

Effect of exchange layer selection – Figure 4 shows the communication cost and model accuracy under different exchange layer settings. The communication cost is calculated by adding the size of model parameters and activations. The red dot indicates the layer automatically selected by our proposed method. When activations are shared at layers closer to the output, both the communication cost and accuracy tend to decrease. One exception is CIFAR-100, where both communication cost and accuracy increase at layers 8 and 9. Nevertheless, our method consistently identifies a practical layer that significantly reduces communication cost while marginally degrading accuracy.

Effect of the number of exchanged activations – We conduct extra experiments with varying numbers of exchanged activations. Due to the limited space, we present the results in Appendix A.5.2.

6 CONCLUSION

We propose ASFL, an activation sharing framework for FL that mitigates data heterogeneity across clients. Our theoretical analysis provides insights into its convergence behavior, and extensive experiments demonstrate its effectiveness across diverse FL benchmarks. While activation sharing may raise privacy concerns, we investigate this aspect using PixelDP, a recently proposed representative DP method, and provide empirical insights. Overall, our results suggest that activation sharing is a practical and effective approach for achieving fast and accurate FL across a wide range of applications. As the present study focuses on a practical gossip-based communication pattern, future work could extend this framework by aggregating activations at a central server and exploring alternative strategies for their utilization.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with writing in English and to provide assistance with writing, modifying, and debugging experimental code. We did not use these tools for any other purposes. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted code was manually inspected and tested for correctness.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section A.1 provides experimental settings and implementation details for all experiments reported in the main manuscript. These include the hardware and software environment, client sampling and data partitioning procedures, optimization settings, and allocation of local update budgets across methods. Tables 4 and 5 report the dataset-specific training settings and method-specific hyperparameters, respectively. We plan to publicly release our source code upon publication.

REFERENCES

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Alp Emre Durmus, Zhao Yue, Matas Ramon, Mattina Matthew, Whatmough Paul, and Saligrama Venkatesh. Federated learning based on dynamic regularization. In *International conference on learning representations*, 2021.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3-4):211–487, 2014.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. Fedvarp: Tackling the variance due to partial client participation in federated learning. In *Uncertainty in Artificial Intelligence*, pp. 906–916. PMLR, 2022.

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*, pp. 5132–5143. PMLR, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *arXiv preprint arXiv:1802.03471*, 2018.

Sunwoo Lee, Tuo Zhang, and A Salman Avestimehr. Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8491–8499, 2023.

Bo Li, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. On the effectiveness of partial variance reduction in federated learning with heterogeneous data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3964–3973, 2023.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

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.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

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*, pp. 1273–1282. Pmlr, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Kazusato Oko, Shunta Akiyama, Denny Wu, Tomoya Murata, and Taiji Suzuki. Silver: Single-loop variance reduction and application to federated learning. In *Forty-first International Conference on Machine Learning*, 2024.

Xiong Peng, Bo Han, Fengfei Yu, Tongliang Liu, Feng Liu, and Mingyuan Zhou. Generative model inversion through the lens of the manifold hypothesis. In *Advances in Neural Information Processing Systems*, 2025.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8432–8440, 2022.

Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xinmei Tian, Tongliang Liu, Bo Han, and Xiaowen Chu. Fedimpro: Measuring and improving client update in federated learning. *arXiv preprint arXiv:2402.07011*, 2024.

Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8485–8493, 2022.

Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

Hao Zhang, Chenglin Li, Wenrui Dai, Junni Zou, and Hongkai Xiong. Federated learning based on model discrepancy and variance reduction. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):10407–10421, 2025.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 253–261, 2020.

A APPENDIX

The appendix is structured as follows:

- Section A.1 summarizes experimental settings corresponding to all the experimental results reported in the main manuscript
- Section A.2 provides detailed proofs of the proposed lemma, theorem, and corollary.
- Section A.3 provides additional details on the gossip-style communication pattern used to transfer latent-space information among clients.
- Section A.4 presents additional experimental results and analyses regarding privacy.
- Section A.5 provides supplementary experimental analyses that further support the empirical findings reported in the main manuscript.

A.1 EXPERIMENTAL DETAILS

This section summarizes the experimental details required to reproduce and interpret the results reported in the main manuscript, including training settings, activation-sharing configurations, cost estimation, and gradient-variance measurement.

A.1.1 MAIN COMPARISON PROTOCOL

Table 4: Training settings used for the main comparison in Table 1. Each column corresponds to the dataset–backbone pair used in the main experiments.

Hyperparameter	**CIFAR-10 (ResNet18)**	**CIFAR-100 (MobileNetV2)**	**SVHN (ResNet20)**	**FEMNIST (CNN)**
Learning rate	0.1	0.05	0.1	0.01
Batch size	64	64	32	20
Weight decay	1e-4	5e-4	1e-4	1e-4
LR decay rounds	100, 150	100, 150	100, 150	–
Communication rounds	200	200	200	200
Total local update budget	32	80	40	12
ASFL tail updates (τ')	16	40	20	6
ASFL standard local updates (τ)	16	40	20	6

We use a unified experimental protocol for the main comparison in Table 1. Experiments are run on two NVIDIA RTX 4070 GPUs using PyTorch 2.7.0 with CUDA 11.8 and MPI-based distributed execution. Each run uses a total of 100 clients, with 10 clients sampled uniformly at random in every communication round. Client data are partitioned using a Dirichlet distribution with concentration parameter $\alpha = 0.1$, which creates a strongly non-IID label distribution across clients. All methods are optimized with mini-batch SGD with momentum 0.9. Table 4 reports the training settings for each dataset–backbone pair. For ASFL, the local computation budget in each communication round is divided equally between the tail round and the subsequent standard local training. Specifically, we set $\tau' = \tau$, where τ' denotes the number of tail-round updates and τ denotes the number of standard local updates. For methods without a tail round, the same total local update budget is used entirely for standard local training. In each communication round, ASFL shares activations corresponding to one mini-batch, regardless of the local update budget. For the baseline methods, we use the same settings reported in Table 4, except that the learning rate is selected by grid search.

Method-specific hyperparameters for FedProx, FedProto, and MOON are summarized in Table 5. FedProx uses the proximal coefficient μ, FedProto uses the prototype regularization coefficient λ, and MOON uses the contrastive coefficient μ and temperature T. For FedProto, λ controls the strength of the regularization term that encourages local prototypes to align with their corresponding global prototypes.

Table 5: Method-specific hyperparameters used in the main comparison.

Method	Hyperparameter	CIFAR-10 (ResNet18)	CIFAR-100 (MobileNetV2)	SVHN (ResNet20)	FEMNIST (CNN)
FedProx	μ (proximal coefficient)	0.001	0.001	0.001	0.001
FedProto	λ (prototype regularization coefficient)	1.0	1.0	1.0	1.0
MOON	μ (contrastive coefficient), T (temperature)	1.0, 0.5	1.0, 0.5	0.1, 0.5	1.0, 0.5

A.1.2 Activation-sharing positions

ASFL defines candidate activation-sharing positions at the level of backbone-specific layer blocks, using the same indexing convention as the x-axis of Figure 2. Each layer block is defined as the smallest architectural unit that preserves the backbone's block structure and ends after a nonlinearity. The candidate position is indexed by the layer block at which the tail-side partial update starts. For a selected layer block b, ASFL forwards a mini-batch through the preceding blocks and exchanges the activation at the input of block b. The selected block is therefore included in the partial training range. The input boundary is excluded from the candidate set because it corresponds to exchanging the original mini-batch input rather than an intermediate activation.

Table 6: Candidate activation-sharing positions for each dataset–backbone pair. The number of candidate positions excludes the input boundary but includes the final classifier position.

Dataset	Backbone	No. of candidate positions	Selected layer block
CIFAR-10	ResNet18	9	4
CIFAR-100	MobileNetV2	9	2
SVHN	ResNet20	10	5
FEMNIST	CNN	4	2

Table 6 summarizes the candidate positions and selected layer blocks for each dataset–backbone pair. The selected layer block is chosen according to the cost–variance trade-off shown in Figure 2.

A.1.3 Activation-sharing quantity

The number of shared batches controls the size of the peer activation pool exchanged during the tail round. Table 12 summarizes the effect of this quantity on accuracy and communication cost. In this ablation, the activation-sharing position is fixed to the selected layer block for each dataset–backbone pair, and only the number of shared batches is varied. One shared batch denotes the activations generated from one mini-batch, with the mini-batch size set according to Table 4.

As described in the communication-efficient activation sharing scheme in Algorithm 1, the received peer activation information I_j is reused when constructing the random mini-batches for the tail round. Thus, the number of shared batches in Table 12 controls the amount of peer activation information exchanged through the directed gossip communication pattern, rather than the total number of tail-round mini-batches used to update the tail subnetwork.

A.1.4 Communication and computational cost

For each candidate activation-sharing position, we compute the communication cost as the number of activation elements exchanged at that position. Specifically, for a candidate layer block b, the transmitted activation is the input to block b, and its communication cost is measured as the product of its tensor dimensions.

For the computational cost, we estimate the backpropagation cost of the tail subnetwork starting from each candidate layer block. The estimate focuses on the convolutional and linear layers included in the tail subnetwork, which are the main parameterized computational components of the backbone. We use this layer-wise estimate to compare the relative cost of tail-side partial training

across candidate positions. Figure 2 visualizes the resulting normalized cost curve together with the cumulative gradient-variance curve for layer-block selection.

A.1.5 GRADIENT VARIANCE MEASUREMENT

This subsection explains how we estimate the gradient-variance term in the cost–variance trade-off of Section 4.2.1. For the variance-reduction curve $\mathcal{V}(k) = \sum_{l=1}^{k-1} \mathrm{Var}(g^l)$, we estimate each layer-block term under the client-sampling setting in Section A.1.1, using one local update step on each active client $i \in \mathcal{A}$.

For each active client, we compute per-sample gradients of the cross-entropy loss with respect to the trainable parameters of each layer block. The parameters in a block are vectorized into a single gradient vector, and the block-wise variance is measured as the average squared deviation of the per-sample gradients from their mini-batch mean:

$$\mathrm{Var}_i(g^l) = \frac{1}{B} \sum_{j=1}^{B} \left\| g_{i,j}^l - \bar{g}_i^l \right\|_2^2 .$$

This corresponds to the trace of the per-sample gradient covariance for layer block l on client i. We average this quantity over the active clients to obtain $\mathrm{Var}(g^l)$. We then construct $\mathcal{V}(k)$ as a prefix sum over the candidate layer blocks before block k, excluding the input boundary, and apply min–max normalization for the layer selection shown in Figure 2.

A.2 CONVERGENCE ANALYSIS OF PARTIAL MODEL UPDATES WITH ACTIVATION SHARING

To make this appendix self-contained, we provide the problem definition and assumptions that are provided in the main text. We consider non-convex and smooth optimization problems:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x}),$$

where $F_i(\mathbf{x}) = \mathbb{E}_{\xi_i\sim D_i}[f(\mathbf{x},\xi_i)]$ is the local loss function associated with the local data distribution D_i of client i and m is the number of clients. Our analysis is based on the following assumptions.

Assumption 1 (Lipschitz continuity). *There exists a constant $\mathcal{L} > 0$, such that $\|\nabla F_i(\mathbf{a}) - \nabla F_i(\mathbf{b})\| \leq \mathcal{L}\|\mathbf{a}-\mathbf{b}\|, \forall \mathbf{a},\mathbf{b}\in\mathbb{R}^d$, and $i \in [m]$.*

Assumption 2 (Unbiased local gradients). *The local gradient estimator is unbiased such that $\mathbb{E}_{\xi_i\sim D_i}[\nabla f(\mathbf{a},\xi_i)] = \nabla F_i(\mathbf{a}), \forall i \in [m]$.*

Assumption 3 (Bounded local and global variances). *There exist two constants $\sigma_x > 0$ and $\sigma_G > 0$, such that the local gradient variance is bounded by $\mathbb{E}[\|\nabla f(\mathbf{x},\xi_i) - \nabla F_i(\mathbf{x})\|^2] \leq \sigma_x^2, \forall i \in [m]$, and the global variability is bounded by $\mathbb{E}\left[\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2\right] \leq \sigma_G^2, \forall i \in [m]$.*

Assumption 4 (Bounded local variance of a partial model). *There exists a constant $\sigma_z > 0$ such that the local gradient variance is bounded by $\mathbb{E}[\|\nabla f(\mathbf{z},\xi_i) - \nabla F_i(\mathbf{z})\|^2] \leq \sigma_z^2, \forall i \in [m]$.*

Here, we consider client i's τ' local updates in a tail round use not only its local activations but also remote activations received from a random client k. Without loss of generality, in the tail round, a fraction α of the data consists of activations from local data, while the remaining fraction $1-\alpha$ consists of activations from remote data. Due to the linearity of expectation over the mini-batch sampling, the expected gradient with respect to z, computed from random mini-batches of these activations, is given by

$$\mathbb{E}[\tilde{p}_{t,j}^i] = \alpha\nabla_z F_i(\mathbf{x}_{t,j}^i) + (1-\alpha)\nabla_z F_k(\mathbf{x}_{t,j}^i), \tag{10}$$

where the expectation is taken over the sampled client k and the mini-batch sampling. Thus, the global update at round t across a randomly activated clients is defined as

$$\tilde{\Delta}_t := \frac{1}{a}\sum_{i\in\mathcal{A}}\left(\sum_{j=0}^{\tau'-1}\tilde{p}_{t,j}^i + \sum_{j=0}^{\tau-1} g_{t,j}^i\right), \tag{11}$$

where the first summation denotes the local τ' updates in the tail round while the second summation denotes the local τ updates in the full round.

Lemma 4 (local full-batch gradient bound)**.** *Under Assumption $1 \sim 4$, the following condition holds:*

$$\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \leq 4a\tau'\mathcal{L}^2\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2\right] + 4a^2(\tau')^2\sigma_z^2 + 2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right]. \tag{12}$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] &= \mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)-\nabla_z F(\mathbf{x}_t)+\nabla_z F(\mathbf{x}_t)\right\|^2\right] \\
&\le 2\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)-\nabla_z F(\mathbf{x}_t)\right\|^2\right]+2\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F(\mathbf{x}_t)\right\|^2\right] \\
&\le 2a\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left\|\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)-\nabla_z F(\mathbf{x}_t)\right\|^2\right]+2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \qquad (13) \\
&= 2a\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left\|\sum_{j=0}^{\tau'-1}\left(\nabla_z F_i(\mathbf{x}_{t,j}^i)-\nabla_z F_i(\mathbf{x}_t)+\nabla_z F_i(\mathbf{x}_t)-\nabla_z F(\mathbf{x}_t)\right)\right\|^2\right] \\
&\quad+2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \\
&= 4a\tau'\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\left\|\nabla_z F_i(\mathbf{x}_{t,j}^i)-\nabla_z F_i(\mathbf{x}_t)\right\|^2\right] \\
&\quad+4a\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\|\nabla_z F_i(\mathbf{x}_t)-\nabla_z F(\mathbf{x}_t)\|^2\right] \qquad (14) \\
&\quad+2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \\
&= 4a\tau'\mathcal{L}^2\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\left\|\mathbf{x}_{t,j}^i-\mathbf{x}_t\right\|^2\right] \\
&\quad+4a^2\tau'\sigma_z^2+2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right],
\end{aligned}$$

where (14) holds because $\nabla_z F_i(\mathbf{x}_t) - \nabla_z F(\mathbf{x})_t)$ has a mean of zero and is independent across i. □

Lemma 5 (Variance of Mixed Gradient Estimator)**.** *Let $\tilde{p}_i$ be a stochastic gradient defined as*

$$\tilde{p}^i = \begin{cases} p^i & \text{with probability } \alpha, \\ p^k & \text{with probability } 1-\alpha, \end{cases}$$

where p^i and p^k are stochastic gradients satisfying

$$\mathbb{E}[p^i] = \nabla_z F_i(\mathbf{x}), \quad \mathbb{E}[p^k] = \nabla_z F_k(\mathbf{x}). \qquad (15)$$

Assume that $\mathrm{Var}(p^i), \mathrm{Var}(p^k) \le \sigma_z^2$. Then, the variance of $\tilde{p}_i$ satisfies

$$\mathrm{Var}(\tilde{p}_i) \le \sigma_z^2 + 4\alpha(1-\alpha)\sigma_G^2.$$

Proof.

$$\begin{aligned}
\mathrm{Var}(\tilde{p}^i) &= \mathbb{E}[\|\tilde{p}^i - \mathbb{E}[\tilde{p}^i]\|^2] \\
&= \alpha\mathbb{E}[\|p^i - \mathbb{E}[\tilde{p}^i]\|^2] + (1-\alpha)\mathbb{E}[\|p^k - \mathbb{E}[\tilde{p}^i]\|^2] \\
&= \alpha\mathbb{E}[\|p^i - \mathbb{E}[p^i] + \mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i]\|^2] \\
&\quad + (1-\alpha)\mathbb{E}[\|p^k - \mathbb{E}[p^k] + \mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i]\|^2] \\
&= \alpha\mathbb{E}[\|p^i - \mathbb{E}[p^i]\|^2] + \alpha\|\mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i]\|^2 + 2\alpha\mathbb{E}[\langle p^i - \mathbb{E}[p^i], \mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i]\rangle] \\
&\quad + (1-\alpha)\mathbb{E}[\|p^k - \mathbb{E}[p^k]\|^2] + (1-\alpha)\|\mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i]\|^2 + 2(1-\alpha)\mathbb{E}[\langle p^k - \mathbb{E}[p^k], \mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i]\rangle]
\end{aligned}$$

Since $\mathbb{E}[p^i - \mathbb{E}[p^i]] = 0$ for all $i \in [N]$, the above variance is simplified as follows.

$$\begin{aligned}\mathrm{Var}(\tilde{p}^i) &= \alpha\mathbb{E}[\|p^i - \mathbb{E}[p^i]\|^2] + \alpha\|\mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i]\|^2 \\ &\quad + (1-\alpha)\mathbb{E}[\|p^k - \mathbb{E}[p^k]\|^2] + (1-\alpha)\|\mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i]\|^2 \\ &= \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) \\ &\quad + \alpha\|\mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i]\|^2 + (1-\alpha)\|\mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i]\|^2. \end{aligned} \tag{16}$$

By the definition of $\mathbb{E}[\tilde{p}^i] = \alpha\mathbb{E}[p^i] + (1-\alpha)\mathbb{E}[p^k]$, the following equations hold:

$$\begin{aligned}\mathbb{E}[p^i] - \mathbb{E}[\tilde{p}^i] &= (1-\alpha)(\mathbb{E}[p^i] - \mathbb{E}[p^k]) \\ \mathbb{E}[p^k] - \mathbb{E}[\tilde{p}^i] &= -\alpha(\mathbb{E}[p^i] - \mathbb{E}[p^k]).\end{aligned}$$

By plugging in the above terms, (16) can be re-written as follows:

$$\begin{aligned}\mathrm{Var}(\tilde{p}^i) &= \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) \\ &\quad + \alpha(1-\alpha)^2\|\mathbb{E}[p^i] - \mathbb{E}[p^k]\|^2 + (1-\alpha)\alpha^2\|\mathbb{E}[p^i] - \mathbb{E}[p^k]\|^2 \\ &= \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) + \alpha(1-\alpha)\|\mathbb{E}[p^i] - \mathbb{E}[p^k]\|^2 \\ &= \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) + \alpha(1-\alpha)\|\mathbb{E}[p^i] - \nabla_z F(\mathbf{x}) + \nabla_z F(\mathbf{x}) - \mathbb{E}[p^k]\|^2 \\ &\leq \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) + 2\alpha(1-\alpha)\|\mathbb{E}[p^i] - \nabla_z F(\mathbf{x})\|^2 + 2\alpha(1-\alpha)\|\mathbb{E}[p^k] - \nabla_z F(\mathbf{x})\|^2 \\ &\leq \alpha\mathrm{Var}(p^i) + (1-\alpha)\mathrm{Var}(p^k) + 4\alpha(1-\alpha)\sigma_G^2 \\ &\leq \sigma_z^2 + 4\alpha(1-\alpha)\sigma_G^2\end{aligned}$$

□

Lemma 6 (framework)**.** *Under assumption 1 ∼ 3, if the learning rate $\eta \leq \min\{\frac{1}{\tau\mathcal{L}}, \frac{1}{4\tau'\mathcal{L}}\}$, the loss reduction of a single communication round is bounded as follows.*

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] &\leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\ &\quad + \frac{3\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{2\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4 + \frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}T}{a\tau}(1 + \alpha - 2\alpha^2)\sigma_G^2.\end{aligned}$$

Proof. Based on Assumption 1, taking expectation of $F(\mathbf{x}_{t+1})$, we have:

$$
\begin{aligned}
\mathbb{E}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) + \langle \nabla F(\mathbf{x}_t), \mathbb{E}[\mathbf{x}_{t+1} - \mathbf{x}_t] \rangle + \frac{\mathcal{L}}{2}\mathbb{E}[\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2] \\
&= F(\mathbf{x}_t) + \langle \nabla F(\mathbf{x}_t), \mathbb{E}[-\eta\tilde{\Delta}_t] \rangle + \frac{\mathcal{L}}{2}\mathbb{E}[\|\eta\tilde{\Delta}_t\|^2] \\
&= F(\mathbf{x}_t) + \frac{\mathcal{L}}{2}\mathbb{E}[\|\eta\tilde{\Delta}_t\|^2] \\
&\quad + \langle \nabla F(\mathbf{x}_t), \mathbb{E}[-\eta\tilde{\Delta}_t + \eta\tau\nabla F(\mathbf{x}_t) + \eta\tau'\nabla_z F(\mathbf{x}_t) - \eta\tau\nabla F(\mathbf{x}_t) - \eta\tau'\nabla_z F(\mathbf{x}_t)] \rangle \\
&= F(\mathbf{x}_t) + \frac{\mathcal{L}}{2}\mathbb{E}[\|\eta\tilde{\Delta}_t\|^2] \\
&\quad - \eta\tau\|\nabla F(\mathbf{x}_t)\|^2 + \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1} g_{t,j}^i + \eta\tau\nabla F(\mathbf{x}_t)\right] \rangle \\
&\quad + \langle \nabla F(\mathbf{x}_t), -\eta\tau'\nabla_z F(\mathbf{x}_t) \rangle + \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1} \tilde{p}_{t,j}^i + \eta\tau'\nabla_z F(\mathbf{x}_t)\right] \rangle \\
&= F(\mathbf{x}_t) - \eta\tau\|\nabla F(\mathbf{x}_t)\|^2 - \eta\tau'\|\nabla_z F(\mathbf{x}_t)\|^2 + \frac{\mathcal{L}}{2}\underbrace{\mathbb{E}[\|\eta\tilde{\Delta}_t\|^2]}_{T_3} \\
&\quad \underbrace{+ \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left(g_{t,j}^i - \nabla F_i(\mathbf{x}_t)\right)\right] \rangle}_{T_1} \\
&\quad \underbrace{+ \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_t)\right)\right] \rangle}_{T_2} \qquad (17)
\end{aligned}
$$

Now, we will bound T_1, T_2, and T_3, separately.

Bounding T_1.

$$
\begin{aligned}
T_1 &= \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left(g_{t,j}^i - \nabla F_i(\mathbf{x}_t)\right)\right] \rangle \\
&= \langle \nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t)\right)\right] \rangle \\
&= \langle \sqrt{\eta\tau}\nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\sqrt{\eta}}{a\sqrt{\tau}}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t)\right)\right] \rangle
\end{aligned}
$$

$$= \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta}{2a^2\tau}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t)\right)\right\|^2\right] - \frac{\eta}{2a^2\tau}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \tag{18}$$

$$\leq \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\mathbb{E}\left[\left\|\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t)\right\|^2\right] - \frac{\eta}{2a^2\tau}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right]$$

$$\leq \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\eta\mathcal{L}^2}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\mathbb{E}\left[\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2\right] - \frac{\eta}{2a^2\tau}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right], \tag{19}$$

where (18) holds due to the fact that $ab = \frac{a^2}{2} + \frac{b^2}{2} - \frac{(a-b)^2}{2}$, and (19) holds based on Assumption 1.

Bounding T_2.

$$\begin{aligned}
T_2 &= \langle\nabla F(\mathbf{x}_t), \mathbb{E}\left[-\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_t)\right)\right]\rangle \\
&= \langle\nabla F(\mathbf{x}_t), -\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\mathbb{E}\left[\tilde{p}_{t,j}^i\right] - \nabla_z F_i(\mathbf{x}_t)\right)\rangle \\
&= \langle\nabla F(\mathbf{x}_t), -\frac{\eta}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t) + (1-\alpha)\left(\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right)\right)\rangle \\
&\leq \frac{\eta\tau'}{2}\|\nabla F(\mathbf{x}_t)\|^2 \\
&\quad + \frac{\eta}{2\tau'}\left\|\frac{1}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t) + (1-\alpha)\left(\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right)\right)\right\|^2 \qquad (20) \\
&\leq \frac{\eta\tau'}{2}\|\nabla F(\mathbf{x}_t)\|^2 \\
&\quad + \frac{\eta}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\underbrace{\left\|\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t) + (1-\alpha)\left(\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right)\right\|^2}_{T_4}, \qquad (21)
\end{aligned}$$

where (20) is based on Young's inequality.

The T_4 can be bounded as follows.

$$\begin{aligned}
T_4 &= \left\|\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t) + (1-\alpha)\left(\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right)\right\|^2 \\
&\leq 2\left\|\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t)\right\|^2 + 2(1-\alpha)^2\left\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2 \\
&\leq 2\mathcal{L}^2\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 + 2(1-\alpha)^2\left\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2 \qquad (22) \\
&= 2\mathcal{L}^2\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 + 2(1-\alpha)^2\left\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F(\mathbf{x}_{t,j}^i) - \left(\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F(\mathbf{x}_{t,j}^i)\right)\right\|^2 \\
&\leq 2\mathcal{L}^2\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 + 8(1-\alpha)^2\sigma_G^2, \qquad (23)
\end{aligned}$$

where (22) and (23) hold based on Assumption 1 and 3, respectively. Now plugging in (23) into (21), we have

$$T_2 \leq \frac{\eta\tau'}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{\eta}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(2\mathcal{L}^2\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 + 8(1-\alpha)\sigma_G^2\right)$$
$$= \frac{\eta\tau'}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{\eta\mathcal{L}^2}{a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 + 4\eta\tau'(1-\alpha)\sigma_G^2. \tag{24}$$

Bounding T_3.

$$\begin{aligned}
T_3 &= \mathbb{E}[\|\eta\tilde{\Delta}_t\|^2] \\
&= \frac{\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\left(\sum_{j=0}^{\tau'-1}\tilde{p}_{t,j}^i + \sum_{j=0}^{\tau-1}g_{t,j}^i\right)\right\|^2\right] \\
&= \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}g_{t,j}^i - \nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\quad + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\leq \frac{2\eta^2\tau}{a}\sigma_x^2 + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\quad + \frac{2\eta^2}{a^2}\underbrace{\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right]}_{T_5}.
\end{aligned} \tag{25}$$

The T_5 can be bounded as follows:

$$\begin{aligned}
T_5 &= \mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&= \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\left\|\tilde{p}_{t,j}^i - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \qquad (26) \\
&= \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\mathbb{E}\left[\left\|\tilde{p}_{t,j}^i - \mathbb{E}[\tilde{p}_{t,j}^i]\right\|^2\right] + \mathbb{E}\left[\left\|\mathbb{E}[\tilde{p}_{t,j}^i] - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right]\right) \\
&= \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\mathbb{E}\left[\left\|\tilde{p}_{t,j}^i - \mathbb{E}[\tilde{p}_{t,j}^i]\right\|^2\right] + (1-\alpha)^2\left\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right) \\
&\leq \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\mathbb{E}\left[\left\|\tilde{p}_{t,j}^i - \mathbb{E}[\tilde{p}_{t,j}^i]\right\|^2\right] + 2(1-\alpha)^2\sigma_G^2\right) \\
&\leq \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left(\sigma_z^2 + 4\alpha(1-\alpha)\sigma_G^2 + 2(1-\alpha)^2\sigma_G^2\right) \\
&= a\tau'\left(\sigma_z^2 + 2(1+\alpha-2\alpha^2)\sigma_G^2\right). \qquad (27)
\end{aligned}$$

where (26) holds because the term is independent across client sampling and mini-batch sampling. By plugging in (27) into (25), we have:

$$T_3 \leq \frac{2\eta^2\tau}{a}\sigma_x^2 + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] + \frac{2\eta^2}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] + \frac{2\eta^2\tau'}{a}\left(\sigma_z^2 + 2(1+\alpha-2\alpha^2)\sigma_G^2\right). \tag{28}$$

Finally, we can plug in (19), (24), and (28) into (17), and we have:

$$\begin{aligned}
\mathbb{E}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \eta\tau\|\nabla F(\mathbf{x}_t)\|^2 - \eta\tau'\|\nabla_z F(\mathbf{x}_t)\|^2 + \frac{\mathcal{L}}{2}T_3 + T_1 + T_2 \\
&= F(\mathbf{x}_t) - \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 - \frac{\eta\tau'}{2}\|\nabla_z F(\mathbf{x}_t)\|^2 \\
&\quad + \left(\frac{\eta^2\mathcal{L}}{a^2} - \frac{\eta}{2a^2\tau}\right)\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\quad + \frac{\eta^2\mathcal{L}}{a^2}\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\nabla_z F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\quad + \frac{\eta\mathcal{L}^2}{2a}\left(\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2\right) \\
&\quad + \frac{\eta^2\tau\mathcal{L}}{a}\sigma_x^2 + \frac{\eta^2\tau'\mathcal{L}}{a}\sigma_z^2 + \frac{2\eta^2\tau'\mathcal{L}}{a}(1+\alpha-2\alpha^2)\sigma_G^2.
\end{aligned}$$

Based on lemma 4, the following bound can be obtained:

$$\begin{aligned}
\mathbb{E}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 - \frac{\eta\tau'}{2}\|\nabla_z F(\mathbf{x}_t)\|^2 \\
&\quad + \left(\frac{\eta^2\mathcal{L}}{a^2} - \frac{\eta}{2a^2\tau}\right)\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\
&\quad + \frac{\eta^2\mathcal{L}}{a^2}\left(4a\tau'\mathcal{L}^2\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\mathbb{E}\left[\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2\right] + 4a^2(\tau')^2\sigma_z^2 + 2a^2(\tau')^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right]\right) \\
&\quad + \frac{\eta\mathcal{L}^2}{2a}\left(\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2\right) \\
&\quad + \frac{\eta^2\tau\mathcal{L}}{a}\sigma_x^2 + \frac{\eta^2\tau'\mathcal{L}}{a}\sigma_z^2 + \frac{2\eta^2\tau'\mathcal{L}}{a}(1+\alpha-2\alpha^2)\sigma_G^2.
\end{aligned}$$

Then, by rearranging the terms, we have:

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \left(2\eta^2(\tau')^2\mathcal{L} - \frac{\eta\tau'}{2}\right)\|\nabla_z F(\mathbf{x}_t)\|^2 \\ &\quad + \left(\frac{\eta^2\mathcal{L}}{a^2} - \frac{\eta}{2a^2\tau}\right)\mathbb{E}\left[\left\|\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\nabla F_i(\mathbf{x}_{t,j}^i)\right\|^2\right] \\ &\quad + \left(\frac{4\eta^2\tau'\mathcal{L}^3}{a} + \frac{\eta\mathcal{L}^2}{2a}\right)\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\eta\mathcal{L}^2}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{\eta^2\tau\mathcal{L}}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\eta^2\tau'\mathcal{L}\sigma_z^2 + \frac{2\eta^2\tau'\mathcal{L}}{a}(1+\alpha-2\alpha^2)\sigma_G^2. \\ &\leq F(\mathbf{x}_t) - \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \left(2\eta^2(\tau')^2\mathcal{L} - \frac{\eta\tau'}{2}\right)\|\nabla_z F(\mathbf{x}_t)\|^2 \\ &\quad + \left(\frac{4\eta^2\tau'\mathcal{L}^3}{a} + \frac{\eta\mathcal{L}^2}{2a}\right)\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\eta\mathcal{L}^2}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{\eta^2\tau\mathcal{L}}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\eta^2\tau'\mathcal{L}\sigma_z^2 + \frac{2\eta^2\tau'\mathcal{L}}{a}(1+\alpha-2\alpha^2)\sigma_G^2, \qquad (29)\end{aligned}$$

where (29) holds if $\eta \leq \frac{1}{2\tau'\mathcal{L}}$.

Then, if $\eta \leq \frac{1}{4\tau'\mathcal{L}}$, the right-hand side can be simplified as follows:

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}_{t+1})] &\leq F(\mathbf{x}_t) - \frac{\eta\tau}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{3\eta\mathcal{L}^2}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\eta\mathcal{L}^2}{2a}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{\eta^2\tau\mathcal{L}}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\eta^2\tau'\mathcal{L}\sigma_z^2 + \frac{2\eta^2\tau'\mathcal{L}}{a}(1+\alpha-2\alpha^2)\sigma_G^2. \qquad (30)\end{aligned}$$

Then, by rearranging the terms, we have:

$$\begin{aligned}\|\nabla F(\mathbf{x}_t)\|^2 &\leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_t) - \mathbb{E}[F(\mathbf{x}_{t+1})]\right) + \frac{3\mathcal{L}^2}{a\tau}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\mathcal{L}^2}{a\tau}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{2\eta\mathcal{L}}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2\end{aligned}$$

Finally, by summing up across T communication rounds and unrolling the telescoping sum, we have:

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] &\leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\ &\quad + \frac{3\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau-1}\left\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\right\|^2 + \frac{\mathcal{L}^2}{a\tau}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{A}}\sum_{j=0}^{\tau'-1}\left\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\right\|^2 \\ &\quad + \frac{2\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2.\end{aligned}$$

□

Lemma 7 (full model discrepancy)**.** *For any learning rate satisfying $\eta \leq \frac{1}{2\sqrt{3}\tau\mathcal{L}}$, we have the following result:*

$$\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] \leq 2\tau\eta^2\sigma_x^2 + 12\tau^2\eta^2\sigma_G^2 + 12\tau^2\eta^2\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2].$$

Proof. For any client $i \in [m]$ and $j \in [\tau]$, we have

$$
\begin{aligned}
&\mathbb{E}[\|\mathbf{x}_{t,j+1}^i - \mathbf{x}_t\|^2] = \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta g_{t,j}^i\|^2] \\
&= \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta \left(g_{t,j}^i - \nabla F_i(\mathbf{x}_{t,j}^i) + \nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\right) \|^2] \\
&= \mathbb{E}[\|\eta \left(g_{t,j}^i - \nabla F_i(\mathbf{x}_{t,j}^i)\right) \|^2] \\
&\quad + \mathbb{E}[\langle \eta \left(g_{t,j}^i - \nabla F_i(\mathbf{x}_{t,j}^i)\right), \mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta \left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\right)\rangle] \\
&\quad + \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta \left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\right) \|^2] \\
&= \mathbb{E}[\|\eta \left(g_{t,j}^i - \nabla F_i(\mathbf{x}_{t,j}^i)\right) \|^2] \qquad (31) \\
&\quad + \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta \left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\right) \|^2] \\
&\leq \eta^2\sigma_x^2 + \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t - \eta \left(\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\right) \|^2] \\
&\leq \eta^2\sigma_x^2 + \left(1 + \frac{1}{2\tau - 1}\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] \qquad (32) \\
&\quad + 2\tau\eta^2 \mathbb{E}[\|\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t) + \nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\|^2] \\
&\leq \eta^2\sigma_x^2 + \left(1 + \frac{1}{2\tau - 1}\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] \\
&\quad + 6\tau\eta^2 \left(\mathbb{E}[\|\nabla F_i(\mathbf{x}_{t,j}^i) - \nabla F_i(\mathbf{x}_t)\|^2] + \mathbb{E}[\|\nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\|^2] + \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\right) \\
&\leq \eta^2\sigma_x^2 + \left(1 + \frac{1}{2\tau - 1}\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] \\
&\quad + 6\tau\eta^2\sigma_G^2 + 6\tau\eta^2\mathcal{L}^2 \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
&= \eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + \left(1 + \frac{1}{2\tau - 1} + 6\tau\eta^2 L^2\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2],
\end{aligned}
$$

where (31) is because $\mathbb{E}[g_{t,j}^i] = \nabla F_i(\mathbf{x}_{t,j}^i)$. The (32) is based on the fact that, for any $\alpha > 0$,

$$
\|\mathbf{a} + \mathbf{b}\|^2 \leq (1 + \frac{1}{\alpha})\|\mathbf{a}\|^2 + (1 + \alpha)\|\mathbf{b}\|^2.
$$

Next, if $\eta \leq \frac{1}{2\sqrt{3}\tau\mathcal{L}}$, the above bound can be simplified as follows.

$$
\begin{aligned}
&\mathbb{E}[\|\mathbf{x}_{t,j+1}^i - \mathbf{x}_t\|^2] \\
&\leq \eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + \left(1 + \frac{1}{2\tau - 1} + 6\tau\eta^2\mathcal{L}^2\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
&\leq \eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + \left(1 + \frac{1}{\tau - 1}\right) \mathbb{E}[\|\mathbf{x}_{t,j}^i - \mathbf{x}_t\|^2] + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2].
\end{aligned}
$$

Then, by unrolling the recursion until j goes to 0, we have

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{x}_{t,j+1}^i - \mathbf{x}_t\|^2] &\leq \sum_{k=0}^{j} \left(1 + \frac{1}{\tau - 1}\right)^k \left(\eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\right) \\
&\leq (\tau - 1)((1 + \frac{1}{\tau - 1})^k - 1) \left(\eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\right) \\
&\leq (\tau - 1)((1 + \frac{1}{\tau - 1})^\tau - 1) \left(\eta^2\sigma_x^2 + 6\tau\eta^2\sigma_G^2 + 6\tau\eta^2 \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\right)
\end{aligned}
$$

Here, the $(\tau-1)\left((1+\frac{1}{\tau-1})^{\tau}-1\right)$ can be simplified as follows:

$$\begin{aligned}(\tau-1)\left(\left(1+\frac{1}{\tau-1}\right)^{\tau}-1\right) &= (\tau-1)\left(\left(1+\frac{1}{\tau-1}\right)^{\tau-1}\cdot\left(1+\frac{1}{\tau-1}\right)-1\right)\\ &\le (\tau-1)\left(e\left(1+\frac{1}{\tau-1}\right)-1\right) \qquad (33)\\ &= \tau(e-1)+1\\ &\le 2\tau,\end{aligned}$$

where (33) holds because $(1+1/\beta)^{\beta}\le e$. Therefore, we have

$$\mathbb{E}[\|\mathbf{x}_{t,j+1}^i-\mathbf{x}_t\|^2]\le 2\tau\eta^2\sigma_x^2+12\tau^2\eta^2\sigma_G^2+12\tau^2\eta^2\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]. \qquad (34)$$

Finally, because the right-hand side is independent of a, we have

$$\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{x}_{t,j+1}^i-\mathbf{x}_t\|^2]\le 2\tau\eta^2\sigma_x^2+12\tau^2\eta^2\sigma_G^2+12\tau^2\eta^2\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]$$

By replacing $j+1$ with j, we have the final bound:

$$\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{x}_{t,j}^i-\mathbf{x}_t\|^2]\le 2\tau\eta^2\sigma_x^2+12\tau^2\eta^2\sigma_G^2+12\tau^2\eta^2\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]. \qquad (35)$$

□

Lemma 8 (partial model discrepancy)**.** *For any step-size satisfying $\eta\le\frac{1}{2\sqrt{6}L\tau'}$, we have the following result:*

$$\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t\|^2]\le 2\eta^2\tau'\sigma_z^2+8\tau'\eta^2\left(\alpha(1-\alpha)+2\tau'\right)\sigma_G^2+16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2].$$

Proof. Here, we focus on the tail round only because the analysis of the full model update is identical to lemma 3 in Reddi et al. (2020). For any client $i\in[m]$ and $j\in[\tau']$, we have

$$\begin{aligned}\mathbb{E}[\|\mathbf{z}_{t,j+1}^i-\mathbf{z}_t\|^2] &= \mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\tilde{p}_{t,j}^i\|^2]\\ &= \mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\left(\tilde{p}_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]+\mathbb{E}[\tilde{p}_{t,j}^i]\right)\|^2]\\ &= \mathbb{E}[\|\eta\left(\tilde{p}_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]\right)\|^2]+\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\mathbb{E}[\tilde{p}_{t,j}^i]\|^2]\\ &\quad+\mathbb{E}[\langle 2\eta\left(\tilde{p}_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]\right),\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\mathbb{E}[\tilde{p}_{t,j}^i]\rangle]\\ &= \mathbb{E}[\|\eta\left(\tilde{p}_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]\right)\|^2]+\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\mathbb{E}[\tilde{p}_{t,j}^i]\|^2], \qquad (36)\end{aligned}$$

where (36) holds since $\mathbb{E}[p_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]]=0$.

We then further bound the right-hand side of (36) as follows:

$$\begin{aligned}\mathbb{E}[\|\mathbf{z}_{t,j+1}^i-\mathbf{z}_t\|^2] &= \mathbb{E}[\|\eta\left(\tilde{p}_{t,j}^i-\mathbb{E}[\tilde{p}_{t,j}^i]\right)\|^2]+\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\mathbb{E}[\tilde{p}_{t,j}^i]\|^2]\\ &\le \eta^2\left(\sigma_z^2+4\alpha(1-\alpha)\sigma_G^2\right)+\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t-\eta\mathbb{E}[\tilde{p}_{t,j}^i]\|^2]\\ &\le \eta^2\left(\sigma_z^2+4\alpha(1-\alpha)\sigma_G^2\right)\\ &\quad+\left(1+\frac{1}{2\tau'-1}\right)\mathbb{E}[\|\mathbf{z}_{t,j}^i-\mathbf{z}_t\|^2]+\underbrace{2\tau'\eta^2\mathbb{E}\left[\left\|\tilde{p}_{t,j}^i\right\|^2\right]}_{T_6} \qquad (37)\end{aligned}$$

where (37) holds due to the fact that, for any $\alpha>0$,

$$\|\mathbf{a}+\mathbf{b}\|^2\le(1+\frac{1}{\alpha})\|\mathbf{a}\|^2+(1+\alpha)\|\mathbf{b}\|^2.$$

The T_6 can be bounded as follows:

$$\begin{aligned} T_6 &= 2\tau'\eta^2\mathbb{E}[\|\tilde{p}_{t,j}^i\|^2] \\ &= 2\tau'\eta^2\mathbb{E}\left[\left\|\alpha\nabla_z F_i(\mathbf{x}_{t,j}^i) + (1-\alpha)\nabla_z F_k(\mathbf{x}_{t,j}^i)\right\|^2\right] \\ &\leq 2\tau'\eta^2\alpha\mathbb{E}[\|\nabla_z F_i(\mathbf{x}_{t,j}^i)\|^2] + 2\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_{t,j}^i)\|^2] \end{aligned} \tag{38}$$

where (38) holds based on the convexity of ℓ_2 norm and Jensen's inequality. Then, we can further bound (38) as follows:

$$\begin{aligned} T_6 &\leq 2\tau'\eta^2\alpha\mathbb{E}[\|\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t) + \nabla_z F_i(\mathbf{x}_t)\|^2] \\ &\quad + 2\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_k(\mathbf{x}_t) + \nabla_z F_k(\mathbf{x}_t)\|^2] \\ &\leq 4\tau'\eta^2\alpha\mathbb{E}[\|\nabla_z F_i(\mathbf{x}_{t,j}^i) - \nabla_z F_i(\mathbf{x}_t)\|^2] + 4\tau'\eta^2\alpha\mathbb{E}[\|\nabla_z F_i(\mathbf{x}_t)\|^2] \\ &\quad + 4\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_{t,j}^i) - \nabla_z F_k(\mathbf{x}_t)\|^2] + 4\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_t)\|^2] \end{aligned}$$

Based on Assumption 4, we have:

$$\begin{aligned} T_6 &\leq 4\tau'\eta^2\mathcal{L}^2\alpha\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] + 4\tau'\eta^2\alpha\mathbb{E}\left[\|\nabla_z F_i(\mathbf{x}_t) - \nabla_z F(\mathbf{x}_t) + \nabla_z F(\mathbf{x}_t)\|^2\right] \\ &\quad + 4\tau'\eta^2\mathcal{L}^2(1-\alpha)\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] + 4\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_t) - \nabla_z F(\mathbf{x}_t) + \nabla_z F(\mathbf{x}_t)\|^2] \\ &= 4\tau'\eta^2\mathcal{L}^2\mathbb{E}\left[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2\right] + 8\tau'\eta^2\alpha\mathbb{E}\left[\|\nabla_z F_i(\mathbf{x}_t) - \nabla_z F(\mathbf{x}_t)\|^2\right] + 8\tau'\eta^2\alpha\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \\ &\quad + 8\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F_k(\mathbf{x}_t) - \nabla_z F(\mathbf{x}_t)\|^2] + 8\tau'\eta^2(1-\alpha)\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2] \\ &\leq 4\tau'\eta^2\mathcal{L}^2\mathbb{E}\left[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2\right] + 8\tau'\eta^2\sigma_G^2 + 8\tau'\eta^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right]. \end{aligned} \tag{39}$$

By plugging in (39) into (37), we have:

$$\begin{aligned} \mathbb{E}[\|\mathbf{z}_{t,j+1}^i - \mathbf{z}_t\|^2] \leq \eta^2 &\left(\sigma_z^2 + 4\alpha(1-\alpha)\sigma_G^2\right) \\ &+ \left(1 + \frac{1}{2\tau'-1} + 4\tau'\eta^2\mathcal{L}^2\right)\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] \\ &+ 8\tau'\eta^2\sigma_G^2 + 8\tau'\eta^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \end{aligned}$$

If $\eta \leq \frac{1}{2\sqrt{2}\mathcal{L}\tau'}$, the bound can be simplified as follows:

$$\begin{aligned} \mathbb{E}[\|\mathbf{z}_{t,j+1}^i - \mathbf{z}_t\|^2] \leq \eta^2 &\left(\sigma_z^2 + 4\alpha(1-\alpha)\sigma_G^2\right) \\ &+ \left(1 + \frac{1}{\tau'-1}\right)\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] \\ &+ 8\tau'\eta^2\sigma_G^2 + 8\tau'\eta^2\mathbb{E}\left[\|\nabla_z F(\mathbf{x}_t)\|^2\right] \end{aligned}$$

Next, by unrolling the recursion until j goes to 0 within the tail round, we have a telescoping sum as follows:

$$\begin{aligned} &\mathbb{E}[\|\mathbf{z}_{t,j+1}^i - \mathbf{z}_t\|^2] \\ &\quad\leq \sum_{i=0}^{j}\left(1 + \frac{1}{\tau'-1}\right)^i\left(\mathbb{E}[\|\mathbf{z}_{t,j}^i - \mathbf{z}_t\|^2] + \eta^2\sigma_z^2 + 4\eta^2\left(\alpha(1-\alpha) + 2\tau'\right)\sigma_G^2 + 8\tau'\eta^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2]\right) \\ &\quad= (\tau'-1)\left(\left(1 + \frac{1}{\tau'-1}\right)^j - 1\right)\left(\eta^2\sigma_z^2 + 4\eta^2\left(\alpha(1-\alpha) + 2\tau'\right)\sigma_G^2 + 8\tau'\eta^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2]\right) \\ &\quad\leq (\tau'-1)\left(\left(1 + \frac{1}{\tau'-1}\right)^{\tau'} - 1\right)\left(\eta^2\sigma_z^2 + 4\eta^2\left(\alpha(1-\alpha) + 2\tau'\right)\sigma_G^2 + 8\tau'\eta^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2]\right) \\ &\quad\leq 2\eta^2\tau'\sigma_z^2 + 8\tau'\eta^2\left(\alpha(1-\alpha) + 2\tau'\right)\sigma_G^2 + 16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2], \end{aligned} \tag{40}$$

where (40) holds because:

$$
\begin{aligned}
(\tau-1)\left(\left(1+\frac{1}{\tau-1}\right)^{\tau}-1\right) &= (\tau-1)\left(\left(1+\frac{1}{\tau-1}\right)^{\tau-1}\cdot\left(1+\frac{1}{\tau-1}\right)-1\right)\\
&\le (\tau-1)\left(e\left(1+\frac{1}{\tau-1}\right)-1\right) \qquad (41)\\
&= \tau(e-1)+1\\
&\le 2\tau,
\end{aligned}
$$

where (41) holds because $(1+1/\beta)^{\beta}\le e$.

Finally, because the right-hand side is independent of a, we have

$$
\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{z}_{t,j+1}^{i}-\mathbf{z}_t\|^2]\le 2\eta^2\tau'\sigma_z^2+8\tau'\eta^2\left(\alpha(1-\alpha)+2\tau'\right)\sigma_G^2+16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2].
$$

By replacing $j+1$ with j for simplicity, we have:

$$
\frac{1}{a}\sum_{i\in\mathcal{A}}\mathbb{E}[\|\mathbf{z}_{t,j}^{i}-\mathbf{z}_t\|^2]\le 2\eta^2\tau'\sigma_z^2+8\tau'\eta^2\left(\alpha(1-\alpha)+2\tau'\right)\sigma_G^2+16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2].
$$

□

Theorem 9. *Under Assumption* $1\sim 3$*, if* $\eta\le\min\{\frac{1}{12\mathcal{L}\tau},\frac{1}{8\mathcal{L}\sqrt{\tau/(\tau')^3}}\}$*, the federated optimization with activation sharing guarantees the following condition.*

$$
\begin{aligned}
\sum_{t=0}^{T-1}\|\nabla F(\mathbf{x}_t)\|^2 \le{}& \frac{4}{\eta\tau}\left(F(\mathbf{x}_0)-\mathbb{E}[F(\mathbf{x}_T)]\right)\\
&+4\eta\mathcal{L}T\left(3\eta\mathcal{L}+\frac{1}{a}\right)\sigma_x^2+4\tau'\eta\mathcal{L}T\left(\eta\mathcal{L}+\frac{4}{\tau}+\frac{1}{a\tau}\right)\sigma_z^2\\
&+\frac{8\eta\tau'\mathcal{L}T}{\tau}\left(2\eta\mathcal{L}\tau'(\alpha(1-\alpha)+2\tau')+\frac{1}{a}(1+\alpha-2\alpha^2)\right)\sigma_G^2.
\end{aligned}
$$

Proof. Based on lemma 1, 7, and 2, we have:

$$
\begin{aligned}
\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \le{}& \frac{2}{\eta\tau}\left(F(\mathbf{x}_0)-\mathbb{E}[F(\mathbf{x}_T)]\right)\\
&+\frac{3\mathcal{L}^2}{\tau}\sum_{t=0}^{T-1}\sum_{j=0}^{\tau-1}\left(2\tau\eta^2\sigma_x^2+12\tau^2\eta^2\sigma_G^2+12\tau^2\eta^2\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\right)\\
&+\frac{\mathcal{L}^2}{\tau}\sum_{t=0}^{T-1}\sum_{j=0}^{\tau'-1}\left(2\eta^2\tau'\sigma_z^2+8\tau'\eta^2\left(\alpha(1-\alpha)+2\tau'\right)\sigma_G^2+16\eta^2(\tau')^2\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2]\right)\\
&+\frac{2\eta\mathcal{L}T}{a}\sigma_x^2+\left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2+\frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2\\
\le{}& \frac{2}{\eta\tau}\left(F(\mathbf{x}_0)-\mathbb{E}[F(\mathbf{x}_T)]\right)\\
&+6\tau\eta^2\mathcal{L}^2T\sigma_x^2+36\tau^2\eta^2\mathcal{L}^2T\sigma_G^2+36\tau^2\eta^2\mathcal{L}^2\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2]\\
&+2\tau'\eta^2\mathcal{L}^2T\sigma_z^2+8\eta^2(\alpha(1-\alpha)+2\tau')\mathcal{L}^2T\frac{(\tau')^2}{\tau}\sigma_G^2+16\eta^2\mathcal{L}^2\frac{(\tau')^3}{\tau}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2]\\
&+\frac{2\eta\mathcal{L}T}{a}\sigma_x^2+\left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2+\frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2.
\end{aligned}
$$

If $\eta \leq \min\{\frac{1}{12\mathcal{L}\tau}, \frac{1}{8\mathcal{L}\sqrt{\tau/(\tau')^3}}\}$, the right-hand side can be bounded as follows:

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\
+ 6\tau\eta^2\mathcal{L}^2 T\sigma_x^2 + \frac{1}{4}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
+ 2\tau'\eta^2\mathcal{L}^2 T\sigma_z^2 + 8\eta^2(\alpha(1-\alpha)+2\tau')\mathcal{L}^2 T\frac{(\tau')^2}{\tau}\sigma_G^2 + \frac{1}{4}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla_z F(\mathbf{x}_t)\|^2] \\
+ \frac{2\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2.
\end{aligned}$$

By the definition, $\|\nabla_z F(\mathbf{x}_t)\|^2 \leq \|\nabla F(\mathbf{x}_t)\|^2$, and we have:

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \leq \frac{2}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\
+ 6\tau\eta^2\mathcal{L}^2 T\sigma_x^2 + \frac{1}{2}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \\
+ 2\tau'\eta^2\mathcal{L}^2 T\sigma_z^2 + 8\eta^2(\alpha(1-\alpha)+2\tau')\mathcal{L}^2 T\frac{(\tau')^2}{\tau}\sigma_G^2 \\
+ \frac{2\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{2\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{4\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2.
\end{aligned}$$

Finally, by rearranging the terms, we have the result:

$$\begin{aligned}
\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] &\leq \frac{4}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\
&\quad + 12\tau\eta^2\mathcal{L}^2 T\sigma_x^2 \\
&\quad + 4\tau'\eta^2\mathcal{L}^2 T\sigma_z^2 + 16\eta^2(\alpha(1-\alpha)+2\tau')\mathcal{L}^2 T\frac{(\tau')^2}{\tau}\sigma_G^2 \\
&\quad + \frac{4\eta\mathcal{L}T}{a}\sigma_x^2 + \left(4+\frac{1}{a}\right)\frac{4\eta\tau'\mathcal{L}T}{\tau}\sigma_z^2 + \frac{8\eta\tau'\mathcal{L}T}{a\tau}(1+\alpha-2\alpha^2)\sigma_G^2 \\
&= \frac{4}{\eta\tau}\left(F(\mathbf{x}_0) - \mathbb{E}[F(\mathbf{x}_T)]\right) \\
&\quad + 4\eta\mathcal{L}T\left(3\eta\mathcal{L}+\frac{1}{a}\right)\sigma_x^2 + 4\tau'\eta\mathcal{L}T\left(\eta\mathcal{L}+\frac{4}{\tau}+\frac{1}{a\tau}\right)\sigma_z^2 \\
&\quad + \frac{8\eta\tau'\mathcal{L}T}{\tau}\left(2\eta\mathcal{L}\tau'(\alpha(1-\alpha)+2\tau') + \frac{1}{a}(1+\alpha-2\alpha^2)\right)\sigma_G^2.
\end{aligned}$$

□

A.3 COMMUNICATION PATTERN

A.3.1 DISCUSSION OF GOSSIP-STYLE COMMUNICATION

We implement latent-space information sharing through a gossip-style communication pattern Boyd et al. (2006). Rather than relying on a centralized collector or broadcasting latent information to all clients, each active client receives latent-space information from one randomly selected active client at each communication round. Specifically, at round r, each active client i uniformly samples one source client j, where $j \neq i$, and receives the latent-space information $I_j^{(r)}$ generated by client j. Here, the directed communication link $j \rightarrow i$ indicates that latent-space information flows from the sampled source client j to the receiving client i.

This communication pattern is particularly suitable for federated learning scenarios with a large number of edge devices. In such environments, clients are naturally distributed and can transfer latent-space information through local client-level connections rather than persistent global connectivity. Our method organizes the transfer of I_i over randomly formed directed communication links while preserving the standard federated training procedure. Thus, the proposed method should be understood as latent-space information sharing over a directed gossip-style communication graph within the standard federated training procedure.

This gossip-style communication pattern has two practical advantages over centralized communication. First, it does not incur blocking. In contrast, when activations are aggregated at a central server and redistributed to clients, blocking is inevitable and increases with the number of clients. Therefore, our design offers better scalability than typical centralized approaches. Second, it enables more robust communication in FL environments. Even if a client is disconnected, gossip-style latent information sharing does not stall the entire system, resulting in a more robust training process. Nevertheless, exploring alternative communication patterns for latent information sharing remains a promising direction for future work.

A.4 ADDITIONAL DISCUSSION ON PRIVACY

While this paper primarily focuses on the proposed activation sharing mechanism for accelerating federated learning, privacy leakage remains a critical concern for practical deployment. In this appendix, we provide additional experimental results and analysis on the differential privacy aspects of ASFL.

The sensitivity of the input-side subnetwork $\mathbf{y}$, which governs the required noise magnitude, is defined as:

$$\triangle_{p,q}^{\mathbf{y}} \;=\; \max_{\xi \neq \xi'} \frac{\big\|\mathbf{y}(\xi) - \mathbf{y}(\xi')\big\|_q}{\|\xi - \xi'\|_p},$$

where ξ and ξ' are two input samples of the corresponding client, and p and q are the degrees of input and output norms. In addition, R defines the radius of the threat model such that any adversarial perturbation confined within $B_p(R) := \{\delta' \in \mathbb{R}^n : \|\delta'\|_p \leq R\}$ is guaranteed to be masked by the injected noise and provides (ϵ, δ)-DP guarantees.

We consider two noise mechanisms for privatizing the shared activations I_i, each offering a distinct privacy-utility trade-off.

- **Laplace mechanism:** The Laplace mechanism adds independent noise drawn from a Laplace distribution to the activation vector: $I_i^P \;:=\; \mathbf{y}(\mathcal{B}_{i,t}) \;+\; \mathrm{Lap}\Big(\frac{\triangle_{p,q}^{\mathbf{y}} L}{\epsilon}\Big)$, where $\epsilon > 0$ is the privacy budget, and I_i^P denotes the privatized activation vector. The Laplace mechanism satisfies *pure* $(\epsilon, 0)$-DP.
- **Gaussian mechanism:** The Gaussian mechanism adds isotropic Gaussian noise to the activation vector: $I_i^P \;:=\; \mathbf{y}(\mathcal{B}_{i,t}) \;+\; \mathcal{N}\big(0,\, \sigma^2 \mathbf{I}\big)\,, \quad \sigma \;=\; \frac{\triangle_{p,q}^{\mathbf{y}} L \sqrt{2\ln(1.25/\delta)}}{\epsilon}$. The Gaussian mechanism satisfies (ϵ, δ)-DP, where δ represents a small probability of the privacy guarantee failing.

In both cases, by the post-processing immunity, the privatized activations I_i^P forwarded to the paired client yield a received representation that is (ϵ, δ)-DP (or $(\epsilon, 0)$-DP for Laplace) with respect to the raw local data of the sending client.

Impact of PixelDP on privacy leakage in ASFL – The following lemma, established in Lecuyer et al. (2018), formalizes the robustness and privacy guarantee provided by PixelDP.

Lemma 10 (PixelDP Stability Bound Lecuyer et al. (2018))**.** *Suppose a randomized function $\mathcal{M}$ with bounded output $\mathcal{M} \in [0, b]$, $b \in \mathbb{R}^+$, satisfies (ϵ, δ)-DP. Then the expected value of its output satisfies:*

$$\forall\, \delta' \in B_p(1): \quad \mathbb{E}\big[\mathcal{M}(x)\big] \;\leq\; e^{\epsilon} \cdot \mathbb{E}\big[\mathcal{M}(x + \delta')\big] \;+\; b\delta, \tag{42}$$

where the expectation is taken over the randomness of $\mathcal{M}$.

Lemma 10 implies that the expected scoring function of the model is stable under bounded input perturbations, providing a certified robustness guarantee against adversarial examples in addition to privacy protection. Together, these properties make PixelDP a principled choice for protecting the activation sharing step in ASFL.

A.4.1 IMPACT OF ACTIVATION SHARING LAYER ON PERFORMANCE

Beyond the choice of noise mechanism and budget, the selection of the network layer at which activations are extracted and shared has a significant impact on the utility of ASFL. To investigate this, we vary the partition point across layers of the network and evaluate three configurations: (i) ASFL (no noise), (ii) ASFL+Laplace ($\epsilon = 0.2$, highest privacy), and (iii) ASFL+Gaussian ($\epsilon = 0.5$, $\delta = 0.001$). The results across all datasets are reported in Tables 7–10. For each configuration, we additionally report the Laplace Drop and Gaussian Drop, defined as the absolute accuracy degradation incurred by adding Laplace or Gaussian noise relative to the noise-free ASFL baseline, respectively.

As the partition layer is moved deeper into the network, the performance of ASFL generally degrades. For example, on CIFAR-10, the accuracy of ASFL decreases from 79.6% at the earlier

partition point to 66.0% at a deeper layer. This behavior can be attributed to two complementary factors. First, deeper layers produce more task-specific, lower-dimensional representations that carry less transferable information across heterogeneous clients. Second, the activations of deeper layers lie in a lower-dimensional manifold, reducing the diversity benefit of activation sharing from a paired client. In contrast, earlier layers produce richer representations that are more amenable to cross-client transfer, thereby better alleviating data heterogeneity.

Moreover, the performance drop induced by noise injection is reported in Tables 7–10. Both the Laplace Drop and Gaussian Drop decrease as the partition point moves deeper into the network. On CIFAR-10, the Laplace Drop reduces from 11.47% at the earliest layer to 0.66% at the deepest layer, and the Gaussian Drop similarly narrows from 12.47% to 0.73%. A consistent pattern is observed on CIFAR-100, where the Gaussian Drop decreases from 5.04% at the input layer to 0.89% at the final layer. This behavior can be explained by the reduction of the progressive compression of representations across network depth as representations become more compressed and task-specific. The injected noise has a proportionally smaller disruptive effect on the downstream task. In other words, activations in deeper layers exhibit reduced sensitivity to privacy perturbations. *This finding suggests a practical trade-off that earlier layers offer higher utility for activation sharing but require more careful noise calibration to preserve privacy, while deeper layers are inherently more noise-tolerant but yield diminishing collaboration benefits.*

Table 7: Impact of activation sharing layer on performance of CIFAR-10 with Laplace ($\epsilon = 0.2$) and Gaussian ($\delta = 0.001, \epsilon = 0.5$). Laplace Drop and Gaussian Drop denote the accuracy reduction (%) of ASFL + Laplace and ASFL + Gaussian relative to the ASFL baseline, respectively.

	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=7$	$l=8$	$l=9$
ASFL	79.6 ± 0.71	76.74 ± 0.47	76.87 ± 0.38	77.38 ± 0.26	75.24 ± 0.03	73.85 ± 0.31	70.14 ± 0.52	69.38 ± 0.44	65.68 ± 0.21	66.00 ± 0.58
ASFL + Laplace	67.59 ± 0.95	69.77 ± 1.32	67.05 ± 0.87	70.53 ± 0.95	69.83 ± 0.97	70.86 ± 0.40	67.93 ± 0.38	66.81 ± 0.95	65.09 ± 0.37	65.34 ± 0.95
ASFL + Gaussian	66.59 ± 1.20	68.54 ± 0.18	68.91 ± 1.31	69.63 ± 0.74	68.55 ± 0.70	70.16 ± 0.79	68.10 ± 0.46	67.60 ± 0.48	65.48 ± 0.29	65.27 ± 0.94
Laplace Drop	11.47 ↓	6.97 ↓	9.82 ↓	6.85 ↓	5.41 ↓	2.99 ↓	2.21 ↓	2.57 ↓	0.59 ↓	0.66 ↓
Gaussian Drop	12.47 ↓	8.20 ↓	7.96 ↓	7.75 ↓	6.69 ↓	3.69 ↓	2.04 ↓	1.78 ↓	0.20 ↓	0.73 ↓

Table 8: Impact of activation sharing layer on performance of CIFAR-100 with Laplace ($\epsilon = 0.2$) and Gaussian ($\delta = 0.001, \epsilon = 0.5$). Laplace Drop and Gaussian Drop denote the accuracy reduction (%) of ASFL + Laplace and ASFL + Gaussian relative to the ASFL baseline, respectively.

	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=7$	$l=8$	$l=9$
ASFL	50.88 ± 0.78	49.91 ± 0.40	49.12 ± 0.73	47.81 ± 0.24	40.95 ± 0.95	31.82 ± 1.13	32.44 ± 2.18	39.21 ± 0.85	44.08 ± 1.22	47.23 ± 0.58
ASFL + Laplace	47.16 ± 0.59	46.20 ± 0.54	45.11 ± 1.02	43.91 ± 0.92	37.48 ± 0.91	29.92 ± 0.57	28.60 ± 1.45	38.06 ± 0.83	41.50 ± 0.80	44.80 ± 0.48
ASFL + Gaussian	45.84 ± 1.00	46.40 ± 0.24	45.19 ± 0.05	43.48 ± 0.94	39.61 ± 2.41	30.90 ± 0.57	29.75 ± 0.75	38.03 ± 0.05	41.17 ± 0.48	46.34 ± 0.17
Laplace Drop	3.72 ↓	3.71 ↓	4.01 ↓	3.9 ↓	3.47 ↓	1.90 ↓	3.84 ↓	1.15 ↓	2.58 ↓	2.43 ↓
Gaussian Drop	5.04 ↓	3.71 ↓	3.93 ↓	4.33 ↓	1.34 ↓	0.92 ↓	2.69 ↓	1.18 ↓	2.91 ↓	0.89 ↓

Table 9: Impact of activation sharing layer on performance of SVHN with Laplace ($\epsilon = 0.2$) and Gaussian ($\delta = 0.001, \epsilon = 0.5$). Laplace Drop and Gaussian Drop denote the accuracy reduction (%) of ASFL + Laplace and ASFL + Gaussian relative to the ASFL baseline, respectively.

	$l=0$	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$	$l=7$	$l=8$	$l=9$	$l=10$
ASFL	89.20 ± 0.16	89.37 ± 0.10	89.11 ± 0.03	89.40 ± 0.18	89.34 ± 0.11	89.12 ± 0.10	88.62 ± 0.28	89.43 ± 0.55	88.39 ± 0.15	86.61 ± 0.41	86.02 ± 0.59
ASFL + Laplace	19.66 ± 0.09	51.61 ± 24.99	19.61 ± 0.02	41.58 ± 31.07	64.19 ± 31.53	87.07 ± 0.41	87.56 ± 0.40	87.71 ± 0.88	86.76 ± 0.65	85.97 ± 0.30	85.31 ± 0.90
ASFL + Gaussian	22.73 ± 4.44	32.99 ± 18.77	19.60 ± 0.01	54.13 ± 25.37	64.76 ± 31.94	85.95 ± 0.81	85.82 ± 0.92	86.67 ± 0.90	88.00 ± 0.25	85.73 ± 0.73	85.36 ± 1.23
Laplace Drop	69.54 ↓	37.76 ↓	69.50 ↓	47.82 ↓	25.15 ↓	2.05 ↓	1.06 ↓	1.72 ↓	1.63 ↓	0.64 ↓	0.71 ↓
Gaussian Drop	66.47 ↓	56.38 ↓	69.51 ↓	35.27 ↓	25.48 ↓	3.17 ↓	2.80 ↓	2.76 ↓	0.39 ↓	0.88 ↓	0.66 ↓

A.4.2 Model inversion attack on shared activations

We examine the privacy behavior of ASFL under an optimization-based model inversion attack to assess whether Gaussian perturbation reduces visually recoverable information in shared latent representations. An adversary observing these representations may attempt to reconstruct the corresponding private inputs.

Table 10: Impact of activation sharing layer on performance of FEMNIST with Laplace ($\epsilon = 0.2$) and Gaussian ($\delta = 0.001, \epsilon = 0.5$). Laplace Drop and Gaussian Drop denote the accuracy reduction (%) of ASFL + Laplace and ASFL + Gaussian relative to the ASFL baseline, respectively.

	$l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = 4$
ASFL	86.38 ± 0.17	86.27 ± 0.57	86.01 ± 0.20	85.56 ± 0.35	85.76 ± 0.24
ASFL + Laplace	83.66 ± 0.14	83.25 ± 0.51	84.77 ± 0.31	85.03 ± 0.08	85.18 ± 0.25
ASFL + Gaussian	83.66 ± 0.19	83.55 ± 0.40	84.96 ± 0.08	85.05 ± 0.24	85.18 ± 0.16
Laplace Drop	$2.72 \downarrow$	$3.02 \downarrow$	$1.24 \downarrow$	$0.53 \downarrow$	$0.58 \downarrow$
Gaussian Drop	$2.72 \downarrow$	$2.72 \downarrow$	$1.05 \downarrow$	$0.51 \downarrow$	$0.58 \downarrow$

Model inversion attacks in the image domain are commonly categorized into optimization-based and generative approaches. Optimization-based methods directly optimize an input image so that its model response matches either a target intermediate representation Mahendran & Vedaldi (2015) or a target model output, such as a class confidence score Fredrikson et al. (2015). In contrast, generative methods introduce a learned image prior and perform inversion in the latent space of a generator, thereby constraining the reconstruction toward the natural image manifold Zhang et al. (2020); Peng et al. (2025).

In this experiment, we consider an optimization-based model inversion attack that reconstructs inputs by matching shared intermediate features. Given an observed shared activation $\tilde{I}$, the attacker reconstructs an input image by solving:

$$x^* = \arg\min_{x'} \left\| \mathbf{y}(x') - \tilde{I} \right\|_2^2 + \lambda R_{\mathrm{TV}}(x')$$

where $\mathbf{y}(x')$ denotes the intermediate representation at the shared activation layer, $R_{\mathrm{TV}}(\cdot)$ is the total variation regularizer that suppresses high-frequency artifacts, and λ controls its strength.

We evaluate the attack with Gaussian perturbation using $(\epsilon, \delta) = (0.5, 0.05)$. Figure 5 shows the shared activations used for inversion and the corresponding reconstructed images.

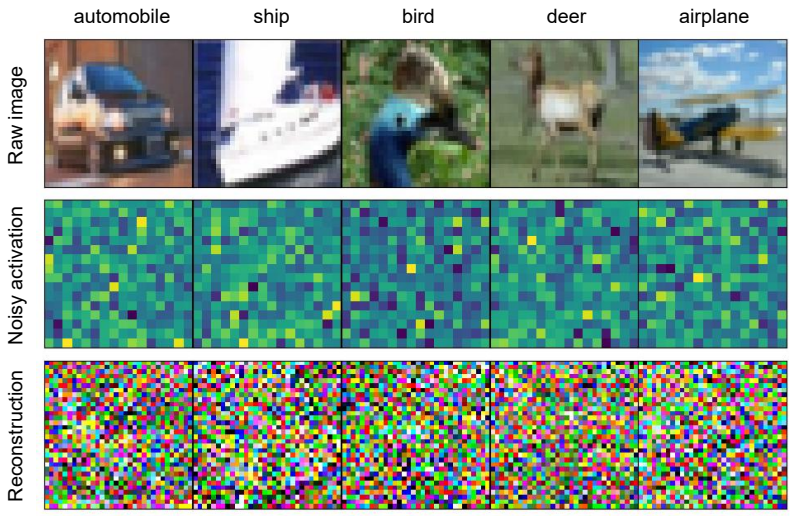


Figure 5: Optimization-based feature inversion attack on shared activations under Gaussian perturbation. The top row shows the original private input images for reference, the middle row visualizes the corresponding perturbed shared activations used for inversion, and the bottom row shows the reconstructed images.

As shown in Figure 5, the reconstructed images do not visually reveal the corresponding private inputs under this attack setting.

A.5 Additional experimental results

A.5.1 Ablation on activation pool construction

We provide an additional ablation study on the construction of the augmented activation pool used in communication-efficient activation sharing. As described in Section 4.2.2, ASFL exchanges only a small subset of remote activations and reuses them within each communication round to reduce communication overhead. Since the number of received remote activations can be smaller than the desired number of remote samples in the augmented pool, the construction strategy influences how effectively the model incorporates newly received remote information without being dominated by local activations.

We compare three strategies for handling the received remote activations. The first strategy, denoted as *Random*, samples remote activations uniformly at random with replacement until the desired number of remote samples is reached. This is the strategy adopted in ASFL. The second strategy, denoted as *Replication*, repeats the received remote activation pool until it reaches the required size and then uses the leading portion. The third strategy, denoted as *Over-sampling*, directly uses the received remote activation pool without additional resampling, even when the number of remote activations is smaller than the target size.

Table 11: Ablation study on activation pool construction. We report validation accuracy (%) for three strategies for handling received remote activations: Random, Replication, and Over-sampling.

Strategy	CIFAR-10 (ResNet18)	FEMNIST (CNN)
Random	$\mathbf{75.25} \pm 0.32$	$\mathbf{84.49} \pm 0.59$
Replication	74.34 ± 0.76	84.11 ± 0.38
Over-sampling	71.95 ± 0.83	84.39 ± 0.30

Table 11 reports the results on CIFAR-10 with ResNet18 and FEMNIST with a CNN. On CIFAR-10, Random achieves the best accuracy, improving over Replication and Default by 0.91 and 3.30 percentage points, respectively. This result indicates that randomly reusing remote activations helps preserve the intended balance between local and remote latent information when communication-efficient sharing substantially reduces the number of exchanged activations. On FEMNIST, the gap among the three strategies is relatively small, with Random still achieving the highest average accuracy. Overall, these results suggest that Random is a practical and robust choice for constructing the augmented activation pool.

A.5.2 Effect of the number of exchanged activations

Table 12 presents ablation results with varying numbers of shared batches across clients. The accuracy generally improves as more batches are exchanged, although communication overhead also increases accordingly. However, the gains are not always consistent, possibly due to randomness in activation sampling. If the remote data used to compute activations are similar to the local data, activation sharing may fail to reduce data heterogeneity while still incurring additional communication cost, and less informative shared activations may even degrade performance. These results suggest that exchanging one or two high-loss batches is sufficient to mitigate data heterogeneity with only marginal communication overhead.

Table 12: Ablation study on activation sharing quantity (the number of shared activation batches).

Shared	**CIFAR-10**		**CIFAR-100**		**SVHN**		**FEMNIST**	
Batches	**Acc (%)** ↑	**Comm (MB)** ↓	**Acc (%)** ↑	**Comm (MB)** ↓	**Acc (%)** ↑	**Comm (MB)** ↓	**Acc (%)** ↑	**Comm (MB)** ↓
8×	74.15 ± 1.17	64	53.27 ± 0.86	32	89.06 ± 0.27	8	84.82 ± 0.28	1.92
4×	76.01 ± 0.50	32	52.34 ± 1.12	16	88.99 ± 0.49	4	84.59 ± 0.21	0.96
2×	75.81 ± 0.70	16	51.18 ± 0.69	8	89.26 ± 0.65	2	84.61 ± 0.07	0.48
1×	75.80 ± 0.55	8	50.24 ± 0.54	4	89.29 ± 0.31	1	84.62 ± 0.28	0.24

A.5.3 Effect of the Dirichlet Concentration Parameter

We conduct additional experiments on CIFAR-10 to evaluate ASFL under different degrees of data heterogeneity. Specifically, we use Dirichlet concentration parameters $\alpha = 0.05$ and $\alpha = 0.5$, corresponding to stronger and weaker non-IID conditions, respectively, relative to the original setting of $\alpha = 0.1$.

Table 13: Validation accuracy (%) on CIFAR-10 with different Dirichlet concentration parameters. Smaller α corresponds to stronger data heterogeneity.

Method	$\alpha = 0.05$	$\alpha = 0.5$
FedAvg	58.93 ± 0.3	85.77 ± 0.3
FedProx	56.16 ± 0.1	86.20 ± 0.1
FedProto	59.44 ± 1.0	86.86 ± 0.2
MOON	58.69 ± 1.1	86.58 ± 0.1
SCAFFOLD	55.37 ± 1.8	87.59 ± 0.1
FedPVR	58.45 ± 0.9	86.27 ± 0.1
FedVARP	61.14 ± 0.4	85.93 ± 0.1
FedImpro	50.88 ± 0.7	83.11 ± 0.5
SplitFed	58.21 ± 0.5	85.75 ± 0.2
ASFL (Ours)	$\mathbf{67.14} \pm 0.5$	$\mathbf{87.60} \pm 0.1$

As shown in Table 13, ASFL still achieves strong validation accuracy under more severe non-IID conditions ($\alpha = 0.05$). It achieves 67.14% accuracy, exceeding FedAvg and the strongest baseline, FedVARP, by 8.21 and 6.00 percentage points, respectively. These results are consistent with our original findings and support the effectiveness of ASFL under strong data heterogeneity.

Under weaker non-IID conditions ($\alpha = 0.5$), all methods achieve higher validation accuracy, and the performance gaps across methods are substantially smaller, leaving limited additional benefit from ASFL in this setting. Nevertheless, ASFL retains the highest validation accuracy at 87.60%. This pattern is consistent with the motivation of ASFL's activation-sharing mechanism, which is designed to mitigate client drift caused by data heterogeneity: its benefit is more pronounced when heterogeneity is severe and diminishes as client data distributions become more similar.

A.5.4 Isolated performance gain from activation sharing

Attributing ASFL's accuracy gains to activation sharing requires separating its contribution from that of partial training. To make this distinction, we evaluate a variant on CIFAR-10 that retains ASFL's partial training procedure, including the same 16 tail update steps in each tail round, but replaces the received remote activations with the client's own local activations.

Table 14: Ablation study on activation sharing on CIFAR-10. Both ASFL variants use the same partial training procedure with 16 tail update steps per tail round.

Method	Validation accuracy (%)
FedAvg	67.70 ± 0.33
ASFL (without remote activations)	66.23 ± 0.41
ASFL (with remote activations)	75.27 ± 0.88

Replacing remote activations with local ones reduces validation accuracy from 75.27% to 66.23%, a drop of 9.04 percentage points despite using the same partial training procedure and tail update budget (Table 14). The local-only variant also falls below FedAvg (67.70%) by 1.47 percentage points. Thus, partial training alone does not explain ASFL's improvement over FedAvg in this setting. the gain depends on incorporating remote activations.