ADAPT-FED: Adaptive Federated Optimization with Learning Stability

Khotso Selialia [1] **Yasra Chandio** [1] **Jimi Oke** [1] **Fatima Anwar** [1]

Abstract

Conventional approaches to Federated Learning (FL), which typically involve gradient descent (GD), pose significant challenges of poor generalization due to training instability. To address these challenges, we introduce `ADAPT-FED`, a framework that refines adaptive optimization by dynamically adjusting learning rates based on the stability observed in GD trajectories. In doing so, it mitigates the adverse effects of training instability and ensures consistent model training across diverse clients. Our empirical results demonstrate the superior performance and generalization capability of `ADAPT-FED` over conventional FL algorithms.

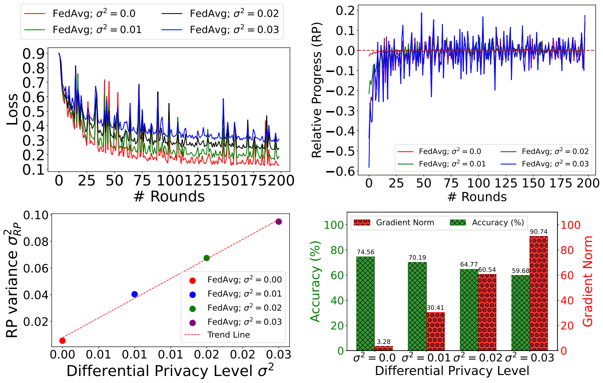

Figure 1. Correlation of DP noise with training instability in a 10-client CIFAR10 setup ($\alpha = 0.3$ non-iid). The variance in relative progress (RP) value shows that increased DP noise elevates instability, leading to larger gradient norms and lower accuracy.

1. INTRODUCTION

Federated learning (FL) enables decentralized model training while preserving data privacy (Li et al., 2019a; Wang et al., 2020b). However, FL implementation faces challenges due to the heterogeneity of clients' data distributions (Hsieh et al., 2020), which complicates the aggregation of global model parameters, leading to poor generalization performance (Li et al., 2019b).

Generalization is the model's ability to perform well on new, unseen data beyond the training dataset (Zhang et al., 2021). In FL, robust generalization is essential for real-world applications where models face heterogeneous data and environmental conditions. Effective generalization prevents overfitting and guarantees the model's reliability across heterogeneous environments. Generalization is mostly pursued using first-order gradient methods (e.g., gradient descent (GD) and its variants (Andrychowicz et al., 2016; Bottou, 2010)) for minimizing training loss during the learning process. However, challenges such as the absence of flat stationary points near the trajectory of first-order gradient methods (Ahn et al., 2022), partial client participation (Li et al., 2019b), and differential privacy (DP) (Dwork, 2006) noise lead to training instability (Abadi et al., 2016). Training instability often results in non-monotonic reductions in the training loss as shown in Figure 1 (top left), affecting the model's ability to generalize (models that train stably generalize well (Chandramoorthy et al., 2022)). Poor generalization leads to models that fail to adapt to new or unseen data, significantly impacting their reliability and effectiveness in practice. Based on these limitations, our main research question is: ***how can we minimize training instability to improve generalization in heterogeneous FL environments?***

Recent developments in FL have focused on enhancing generalization through sharpness-aware optimization techniques, which aim for flatter minima within the loss landscape, a strategy proven effective in centralized learning environments (Foret et al., 2020; Kwon et al., 2021). Building on these successes, adaptations such as FedSAM (Qu et al., 2022) improve generalization by applying sharpness-aware minimization at each client, thereby promoting local generalization. Additionally, adaptive optimization techniques (Reddi et al., 2020) attempt to smooth the global loss surface to enhance generalization in FL further. However, despite these innovations, the localized nature of optimizations often does not fully address the global stochasticity of FL environments, resulting in a significant gap in achieving optimal global model performance when aggregating locally optimal updates (Sun et al., 2023). This highlights the ongoing need for novel approaches that refine local models and effectively leverage these improvements for robust global

[1] University of Massachusetts Amherst USA. Correspondence to: Khotso Selialia <kselialia@umass.edu>, Yasra Chandio <ychandio@umass.edu>.

generalization, a critical yet unmet challenge in FL settings.

To address the challenges above, we propose **Adapt**ive **Fed**erated Optimization with Learning Stability (ADAPT-FED), a framework designed to enhance both stability and generalization of FL models. ADAPT-FED dynamically adjusts learning rates based on historical relative progress (RP) metrics, which act as stability indicators within the optimization process. Specifically, ADAPT-FED increases the learning rate during stable periods and reduces it during unstable periods to ensure consistent training progress and mitigate the typical instabilities caused by erratic updates. In designing and evaluating ADAPT-FED, we make the following contributions.

- We identify and analyze the causes of training instability and poor generalization in heterogeneous FL settings, focusing on the adverse effects of GD's lack of flat stationary points, partial client participation, and DP noise.
- We propose ADAPT-FED, an FL framework that dynamically adjusts learning rates based on the relative progress (RP) metric. RP assesses the improvement or regression in model performance from one training iteration to the next. By adapting learning rates based on historical stability dynamics, ADAPT-FED enhances the stability and generalization of training.
- We theoretically validate the effectiveness of ADAPT-FED in mitigating training instability and data heterogeneity. Our analysis provides precise bounds on the improvements in stability and convergence rates, highlighting how ADAPT-FED mitigates the impact of training instability on the overall learning process.
- We conduct rigorous empirical evaluations demonstrating that ADAPT-FED significantly enhances model generalization across multiple datasets (CIFAR10, CIFAR100, and UTK), with improvements of up to $+\mathbf{5.06}\%$, $+\mathbf{14.79}\%$, and $+\mathbf{7.79}\%$ in generalization performance compared to SOTA FL algorithms.

2. Related Work

Sharpness-aware FL focuses on adapting sharpness-aware optimization techniques (Caldarola et al., 2022; Dai et al., 2023; Qu et al., 2022; Sun et al., 2023) to address the degradation of global model generalization under non-IID settings. Sharpness-aware optimization methods (Cha et al., 2021; Izmailov et al., 2018; Foret et al., 2020; Kwon et al., 2021) improve generalization in centralized learning by seeking flatter minima in the loss landscape (Foret et al., 2020; Kwon et al., 2021), which has inspired several adaptations for FL settings by prior work. For instance, FedSAM (Qu et al., 2022) and its variants (FedGAMMA (Dai et al., 2023), SWA (Izmailov et al., 2018)) apply these optimizations locally at each client, promoting convergence to flatter local minima and improving local generalization. In conclusion, by minimizing loss and sharpness with smoother loss landscapes, sharpness-aware optimizations address client drift and improve both convergence and generalization across diverse and unseen data.

Adaptive optimization techniques in FL. focus on addressing the convergence challenges posed by heterogeneous client data and communication constraints. In particular, FedAdagrad (Reddi et al., 2020) adjusts the learning rate based on the accumulated gradient squared values, making it effective for sparse-gradient tasks and ensuring that clients with less frequent updates still contribute meaningfully. FedAdam (Reddi et al., 2020) builds on this by incorporating momentum terms to smooth out the optimization trajectory, offering robustness to noisy gradients. FedYogi (Reddi et al., 2020) uses a more conservative update rule, reducing the risk of divergence in situations with large gradients. By adapting to the local landscape of each client, these optimizers ensure faster and more stable convergence, especially where simple methods like FedAvg (McMahan et al., 2017) struggle due to the high variance in client updates.

Limitations of existing techniques. Sharepness-aware optimization methods stabilize training by optimizing parameters in a local neighborhood to find flatter minima that are hypothesized to generalize better. However, FL introduces stochasticity due to non-IID data, and local perturbations might not fully account for the global variance. Secondly, while these methods enhance local sharpness minimization, they struggle to account for broader variations in client data distributions. As a result, the aggregation of locally optimal updates does not always translate to improved global generalization in FL (Sun et al., 2023), which can lead to instability or slower convergence.

Further, adaptive and sharpness-aware optimization techniques are significantly affected by intermittent client participation in FL due to common issues such as connectivity constraints, battery limitations, and privacy concerns. Since not all devices participate in every training round, model updates reflect a biased subset of the overall network. This partial participation skews gradient updates, leading to inconsistent optimization steps that can degrade generalization and slow convergence. As participation patterns vary over time, the relative influence of individual clients on model updates shifts, further complicating the challenge of learning a globally representative model in FL (Li et al., 2019b).

Research Implications. The outlined limitations highlight significant gaps in the current approaches employed in FL. There is a clear need for developing new approaches to 1) better integrate local optimizations with global convergence needs and 2) manage the inherent data heterogeneity and participation variability more effectively for training stability. This obviates the efficacy of many efficient methods (e.g., learning rate scheduling methods (Li et al., 2019b)) and highlights the need for a mechanism to scale updates based on local progress while supporting global convergence.

3. Preliminaries and Problem Setup

To establish the context for our study, we define FL with DP and introduce the problem of *training instability* in FL. Specifically, we cover the foundational elements, such as the common FL aggregation algorithm (FedAvg) with DP and data heterogeneity, the notion of training instability in centralized machine learning, and its manifestation in FL. The central question addressed through both theoretical and empirical analysis is: ***What is the effect of training instability on generalization in FL?*** The analysis demonstrates that training instability deteriorates generalization.

3.1. Common FL aggregation algorithm (FedAvg)

FedAvg trains a global model using a server and K decentralized clients. Each client $k \in K$ has local data $\mathcal{D}_k = \{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$, consisting of N_k tuples $\{(\boldsymbol{x}_k^n \in \boldsymbol{X}_k, y_k^n \in \boldsymbol{Y}_k)\}_{n=1}^{N_k}$ representing input and output spaces. Real-world FL scenarios often involve non-IID/heterogeneous decentralized clients' data due to factors such as *data distribution skew* (Hsieh et al., 2020; Liu et al., 2020), resulting in violations of identicalness. FL uses the unified local data $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ to learn an optimal global model $h^* \in H$ (with global parameters $\boldsymbol{\theta}$) from a class of models H that map inputs $\boldsymbol{x}_k^n$ to outputs y_k^n. The global model is obtained by minimizing the global *empirical risk* objective $F(\cdot)$ at each round $t+1 \in T$:

$$\boldsymbol{\theta}^* \triangleq \arg\min_{\boldsymbol{\theta}} \left\{ F(\boldsymbol{\theta}^{t+1}) = \sum_{i=1}^{K} w_k F_k({\boldsymbol{\theta_k}}^{t+1}) \right\} \tag{1}$$

where F_k is the local empirical risk for client k with local parameters initialized as $\boldsymbol{\theta}_k^{t+1} \longleftarrow \boldsymbol{\theta}^t$, and $w_k = \left(\frac{N_k}{\sum_{k=1}^{K} N_k}\right)$ denotes the importance of the k-th client, typically based on the size of the dataset at client k. At the $t+1$-th round, each client k receives the global model parameters from the server and performs local model training over E epochs. This process adjusts the local parameters by minimizing the local empirical risk using a constrained view of the first-order gradient descent (GD) algorithm:

$$\boldsymbol{\theta}^* \longleftarrow \arg\min_{\boldsymbol{\theta^*}} \left\{\boldsymbol{\theta^*} - \boldsymbol{\theta}_k^{t+1}\right\}^T \nabla F_k(\boldsymbol{\theta}_k^{t+1})\ s.t.\ ||\boldsymbol{\theta}^* - \boldsymbol{\theta}_k^{t+1}||^2 \leq \epsilon \tag{2}$$

where $\nabla F_k(\boldsymbol{\theta}_k^{t+1})$ represents the local empirical risk's gradients. Equation 2 finds the optimal $\boldsymbol{\theta}^*$ by searching for parameter values within an ϵ-ball around the current local model $\boldsymbol{\theta}_k^{t+1}$ value that maximizes the linearization of the objective $F_k(\boldsymbol{\theta}_k^{t+1})$. The locally updated models are then sent back to the server for aggregation into a global model $\boldsymbol{\theta}^* = \sum_{k=1}^{K} w_k \boldsymbol{\theta}_k^{t+1}$ without sharing clients' local data.

However, adversaries may still infer private information from the local data by analyzing the parameters that the clients have trained (Shokri & Shmatikov, 2015). To mitigate this risk, clients apply DP by adding artificial Gaussian noise $\boldsymbol{n}_k^{t+1} \sim \mathcal{N}(0, \sigma^2)$ to their gradients during local training and send the randomized parameters to the server. DP protects clients from information leakage at the expense of *instability* in learning, leading to poor generalization.

3.2. Instability in Machine Learning

Training instability in centralized learning (Ahn et al., 2022) refers to the phenomenon in which GD in Equation 2 causes the local risk $F_k(\boldsymbol{\theta}_k^{t+1})$ to decrease *non-monotonically*. This instability arises from the absence of *flat stationary points* along the GD trajectory (Ahn et al., 2022). In this work, we argue that this concept extends to FL, particularly when compounded by partial device participation, which makes the averaged sequence of global models across rounds $t+1$ $\{\boldsymbol{\theta}^{t+1}\}$ to have a large variance (Li et al., 2019b):

Proposition 1 *(Instability in FL). Assume the empirical risk of a global model parameter $\boldsymbol{\theta}$ with a weight decay is:*

$$F(\boldsymbol{\theta}^{t+1}) = \sum_{k=1}^{K} w_k F_k(\boldsymbol{\theta_k^{t+1}}) + \gamma||\boldsymbol{\theta}_k^{t+1}||_2^2 \tag{3}$$

If we partition the global model parameter $\boldsymbol{\theta}^{t+1} = [\boldsymbol{\xi}; \boldsymbol{\zeta}]$ such that a subset of the global model parameters $\boldsymbol{\zeta}$ is positive homogeneous, i.e. for any input data $\boldsymbol{x}_k$ and positive number $c > 0$, then the global empirical risk $F(\boldsymbol{\theta}^{t+1})$ has no stationary point if $\boldsymbol{\zeta} \neq 0$.

$$F(\boldsymbol{x}_k, [\boldsymbol{\xi}; \boldsymbol{\zeta}]) = F(\boldsymbol{x}_k, [\boldsymbol{\xi}; c\boldsymbol{\zeta}]) \tag{4}$$

Proof. The positive homogeneity condition implies that:

$$\langle \nabla_{\boldsymbol{\zeta}} F(\boldsymbol{x}_k, [\boldsymbol{\xi}; \boldsymbol{\zeta}]), \zeta \rangle = 0. \tag{5}$$

Therefore, if $\nabla_{\boldsymbol{\zeta}} L(\boldsymbol{\theta}^{t+1}) \neq 0$, we have:

$$\nabla_{\boldsymbol{\zeta}} F(\boldsymbol{\theta}^{t+1}) = \nabla_{\boldsymbol{\zeta}} \left(\sum_{k=1}^{K} 0 + \gamma||(\boldsymbol{\theta}_k^{t+1}||_2^2 \right) = 2\gamma\zeta. \tag{6}$$

Thus, $\nabla_{\boldsymbol{\zeta}} L(\boldsymbol{\theta}^{t+1}) \neq 0$ when $\boldsymbol{\zeta} \neq 0$ (which exists in many models that use regularization (Ahn et al., 2022)), indicating trivial stationary points with non-zero gradients. The non-zero gradients are protected from adversarial attacks through DP by perturbation through Gaussian noise:

$$\nabla_{\boldsymbol{\zeta}} F_{\text{DP}}(\boldsymbol{\theta}^{t+1}) = \nabla_{\boldsymbol{\zeta}} F(\boldsymbol{\theta}^{t+1}) + \mathcal{N}(0, \sigma^2) \neq 0. \tag{7}$$

DP noise adds additional random non-zero components to $\nabla_{\boldsymbol{\zeta}} F_{\text{DP}}(\boldsymbol{\theta}^{t+1})$, hindering stable convergence.

3.3. Characteristics of Instability in FL

Inspired by (Ahn et al., 2022) we quantify instability in terms of global empirical risk's behavior (relative progress) across rounds:

Proposition 2 *(Relative progress (RP)). Assume that the global empirical risk $F(\boldsymbol{\theta})$ is L-smooth as done in previous works that analyze GD (Ahn et al., 2022). We define RP:*

$$RP = \eta \cdot ||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)|| \cdot \left[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t)\right] \quad (8)$$

RP quantifies how much the global empirical risk improves after updating the gradients at each round relative to the size of the gradient and the step size η taken. Stability in FL is achieved when the RP consistently remains below a negative threshold, indicating steady and controlled progress in the optimization process without erratic fluctuations.

Proof. We assume that the global empirical risk $F(\boldsymbol{\theta})$ is L-smooth (we say F is L-smooth if $||\nabla F(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta}')|| \leq L||\boldsymbol{\theta} - \boldsymbol{\theta}'||$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}'$ and $L > 0$), and subsequently exploits the associated *descent lemma:*

$$F(\boldsymbol{\theta}^{t+1}) \leq F(\boldsymbol{\theta}^t) - \eta\left(1 - \frac{L\eta}{2}\right)||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)||^2 \quad (9)$$

$$F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t) \leq -\eta\left(1 - \frac{L\eta}{2}\right)||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)||^2 \quad (10)$$

$$\iff \eta \cdot ||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)|| \cdot \left[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t)\right]$$

$$\leq -\left(1 - \frac{L\eta}{2}\right) \cdot \eta^2 \cdot ||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)||^3 \quad (11)$$

Takeaway*: Analyzing implications of the descent inequality 11 reveals two scenarios:* 1) *When $L < \frac{2}{\eta}$, the right-hand side (RHS) update term remains negative, ensuring each gradient step reduces the global empirical risk, promoting stable convergence.* 2) *Conversely, when $L > \frac{2}{\eta}$, the RHS term becomes positive, potentially increasing the global empirical risk at each step, leading to divergence and destabilizing the optimization.*

The linearized GD objectives in Equation 11 and Equation 2 suggest that GD aims to find optimal model parameters $\boldsymbol{\theta}^*$ in an ϵ-ball in $\boldsymbol{\theta}_k$-space, with the step size determined by the η. This confines the objective's validity to a small region around the current parameter $\boldsymbol{\theta}_k^{t+1}$. Consequently, larger η values cannot guarantee convergence because they broaden the search beyond this designated region, resulting in substantial changes in some parameters while only slightly adjusting others.

When DP is used in FL, the gradient norms in Equation 11 become larger due to the addition of non-zero noise values. Large gradient norms lead to excessively large updates, which in turn cause the optimization process to overshoot minima. This violation of the optimal step size condition results in more unstable convergence. To avoid algorithmic instability and ensure fast and effective learning, it is important to keep the η below the threshold $\frac{2}{L}$ where L is the Lipschitz constant that bounds the rate of change of the gradient and defines the smoothness of the loss function.

3.4. Empirical Analysis of Instability in FL

As a preliminary study, we compute the instability RP and generalization (accuracy) metrics of FedAvg for the CIFAR10 benchmark across FL rounds. We use the experimental setup in §C.2.

Observation: In Figure 1, the RP variance values across FL rounds are greater than zero, indicating training instability in FL. Higher levels of DP lead to increased RP variance, suggesting higher training instability. Additionally, higher gradient norm values, which are observed with increasing DP, signal slower convergence during training and a noticeable decline in generalization.

Takeaway*: In FL with DP, there exists a sweet spot in stability where the η is optimized to maintain better generalization. Identifying the optimal η enables maximization of both the model's convergence rate and its generalization capability by allowing all parameters to reach their optimal values, necessitating using larger η for parameters that have a minimal impact on the model and smaller η for those that significantly alter it.*

4. Proposed Method: ADAPT-FED

Based on the preliminaries in §3, we now focus on the local training and aggregation steps of our method ADAPT-FED to improve the convergence and generalization capability of FL models learning under the instability regime.

4.1. Training Process of ADAPT-FED

Local training. At the start of training round $t+1$, client k receives the aggregated global model $\boldsymbol{\theta}^t$ from the previous round t, initializes its local model with the global one $\boldsymbol{\theta}_k^{t+1} \longleftarrow \boldsymbol{\theta}^t$, and runs E training epochs $\boldsymbol{\theta}_k^{t+1}$ with DP.

Gradient Descent with DP. Client k trains $\boldsymbol{\theta}_k^{t+1}$ using GD to find the best local objective $F_k(\cdot)$ such that Equation 10 is satisfied As GD progresses, the global model's training stability depends on the magnitude of the learning rate η and the gradient norms $||\nabla F(\boldsymbol{\theta}^t)||$. When η is chosen such that $L > \frac{2}{\eta}$, we have observed that the RHS term in Equation 10 becomes positive, which can increase the empirical risk at each step and lead to divergence, destabilizing the optimization process. To stabilize the optimization process, we must take into account a crucial piece of conventional wisdom originating from the quadratic Taylor approximation model of GD. According to this wisdom (LeCun et al., 1992; Schaul et al., 2013), if the sharpness at step $e \leq E$ is L, then the η should be set no larger than $\frac{2}{L}$ to prevent training instability. The $\eta = \frac{2}{L}$ rule continuously anneals the step size, ensuring that the training objective decreases at each iteration.

Challenges in Learning Rate Scheduling Scheduling the η

Algorithm 1 ADAPT-FED

```
Input: Number of clients $K$, initial global model $\boldsymbol{\theta}^0$, initial learning rate $\eta_0$, differential privacy (DP) noise variance $\sigma^2$, number of training rounds $T$, number of local epochs $E$, learning rate decay constant $\beta$.
Output: Final global model $\boldsymbol{\theta}^T$
for $t = 1$ to $T$ do
  Server broadcasts global model $\boldsymbol{\theta}^{t+1}$ to all clients
  for each client $k \in \{1, \dots, K\}$ in parallel do
    Compute relative progress ratio for client $k$: $RP_k^t$
    $= \eta \cdot ||\nabla F(\boldsymbol{\theta}^t) + \mathcal{N}(0, \sigma^2)|| \cdot \left[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t)\right]$
    Adjust learning rate $\eta$ using historical $RP$:
    $\eta = \eta_0 \cdot \left(\frac{\beta}{\overline{RP}_t}\right);$
    $\overline{RP}_t = \frac{1}{N} \sum_{i=t-N+1}^{t} \exp(RP_i); \forall i \in \{1, \dots, t\}$
    Initialize local model $\boldsymbol{\theta}_k^0 \leftarrow \boldsymbol{\theta}^{t+1}$
    for $e = 0$ to $E$ do
      Compute local gradient $\nabla F_k(\boldsymbol{\theta}_k^e)$
      Add DP noise: $\nabla F_k^{\text{DP}} = \nabla F_k(\boldsymbol{\theta}_k^e) + \mathcal{N}(0, \sigma^2)$
      Update local model: $\boldsymbol{\theta}_k^{e+1} \leftarrow \boldsymbol{\theta}_k^e - \eta \nabla F_k^{\text{DP}}$
    end for
    Client $k$ updates local model $\boldsymbol{\theta}_k^{t+1} \leftarrow \boldsymbol{\theta}_k^{e+1}$
  end for
  Server Aggregation: $\boldsymbol{\theta}^{t+1} = \sum_{k=1}^{K} w_k \boldsymbol{\theta}_k^{t+1}$
end for
```

using $\eta = \frac{2}{L}$ rule results in small η that hinder the learning process due to the progressive increase in L at each training iteration, causing slow or even stalled convergence (Cohen et al., 2021). This stalled convergence happens particularly when the model approaches areas of high sharpness (high sensitivity of the loss to perturbations in the parameter space) in the loss landscape, which are typically regions with steep gradients. Thus, the inverse relationship $\frac{2}{L}$ results in tiny η, potentially hindering convergence by making the steps too cautious and slow. It is also computationally expensive to compute L at each iteration since it involves the second-order derivative of the objective function.

4.2. ADAPT-FED Dynamic Learning Rate Adjustment

To address training instability based on sharpness information from the loss landscape and mitigate the challenges described, we present the entire process of *ADAPT-FED* in Algorithm 1. Let $F(\boldsymbol{\theta})$ be an unstable objective function: a function differentiable w.r.t. parameters $\boldsymbol{\theta}$. We want to minimize the expected value of this function, $\mathbb{E}[F(\boldsymbol{\theta})]$, relative to its parameters, $\boldsymbol{\theta}$. We use $\{RP_1, \dots, RP_T\}$ to show the objective function's training stability measures at different FL training rounds $t \in \{1, \dots, T\}$. ADAPT-FED introduces a novel method for scheduling the learning rate η. It dynamically schedules the η based on the moving averages of the historical RP, where the hyperparameter $\beta > 0$ controls the decay rate of the moving average, allowing for precise control of GD steps based on the observed training instability. Specifically, ADAPT-FED calculates the moving average of RP values across training rounds ($\overline{RP}_t$) to smooth out the measure of recent training progress over a window of N iterations. This average is vital for assessing the overall direction and stability of the learning process:

$$\overline{RP}_t = \frac{1}{N} \sum_{i=t-N+1}^{t} \exp(RP_i); \quad \forall i \in \{1, \dots, t\} \tag{12}$$

Inspired by (LeCun et al., 1992; Schaul et al., 2013), which proposes that the η should be chosen based on the inverse sharpness of the objective function $\eta_k = \frac{2}{L}$ that measures stability, ADAPT-FED schedules the η for the next iteration η_{t+1} based on the inverse of the moving average of $\overline{RP}_t$. This transformation, in which each RP_i value is exponentiated before the moving average is calculated, has several benefits: 1) The exponential function increases very rapidly, making it possible to assign more weight to higher RP values; thus, higher RP values will have a disproportionately larger learning rate (η) scheduling effect for enhanced stability. 2) If the RP includes negative values, the exponential function ensures all transformed RPs are positive to guarantee positive learning rates. This scaling is designed to stabilize the training dynamically, responding to the immediate past training stability conditions:

$$\eta = \eta_0 \cdot \left(\frac{\beta}{\overline{RP}_t}\right) \tag{13}$$

Intuition: ADAPT-FED fine-tunes η to match the actual training dynamics. When the RP is low, indicative of stable progress, η increases, which is conducive to faster convergence. Conversely, high RP signals training instability, prompting a reduction in the η to safeguard against potential divergences, mitigating training instability.

Based on the learning rate scheduling procedure in Equation 13, we perform the local model $\boldsymbol{\theta}_k^t$ update as:

$$\boldsymbol{\theta}_k^{t+1} = \boldsymbol{\theta}_k^{t+1} - \eta_0 \cdot \left(\frac{\beta}{\overline{RP}_t}\right) \cdot \nabla F_k(\boldsymbol{\theta}_k^{t+1}) \tag{14}$$

Each training round t ends with the termination of local training and the return of updated local models to the server for aggregation into a global model.

Server Aggregation: The updated local models are then aggregated at the server to newly update the global model

$\boldsymbol{\theta}^{t+1}$ for the next round. We adopt the commonly used FedAvg aggregation scheme to aggregate local models into a global model $\boldsymbol{\theta}^{t+1} = \sum_{k=1}^{K} w_k \boldsymbol{\theta}_k^{t+1}$.

4.3. Adaptive Learning Rate Components

In this section, we outline the methods for setting the learning rate decay constant β and the initial learning rate η_0.

4.3.1. Learning Rate Decay Constant β

β is selected to dynamically adapt the η across training rounds to effectively manage instability. Drawing inspiration from feedback control systems, where responses are often tailored based on the deviation from a desired state (Barto et al., 1983), β is designed to respond exponentially to the relative progress value RP_t. By setting $\beta = \exp(\overline{RP_t})$, the η adjustment becomes highly responsive to fluctuations in training stability, increasing the η when deviations are minor and decreasing it substantially when deviations are large.

This exponential transformation of RP values by β effectively scales the η adjustments in a manner that is both sensitive and proportional to the observed training dynamics. The formulation $\eta = \eta_0 \cdot \left(\frac{\exp(\overline{RP_t})}{\overline{RP_t}}\right)$ provides a solution to adaptively control the η by embedding a mechanism that intuitively mimics natural adaptive responses, enhancing the algorithm's ability to cope with the complex and variable training instability conditions in FL.

4.3.2. Initial Learning Rate η_0

We empirically determine a suitable initial learning η_0 inspired by *learning rate range test* in (Smith, 2017), which involves progressively testing a series of learning rates eta in a predefined range and observing the model performance associated with each η. The goal is to identify a *sweet spot* where empirical risk generalizes well. This practical, data-driven method makes it easy to choose an η_0 that works well with the federated network and how its data is distributed.

5. Theoretical Analysis

This section discusses the theoretical bounds of ADAPT-FED, focusing on its convergence rate. We provide theorems (with proofs established in §B) that set upper bounds on how quickly ADAPT-FED can stably converge, leading to performance generalization. These theorems are essential for understanding how the eta affects convergence. Before that, we introduce the assumptions consistent with other works in FL (Li et al., 2019b):

5.1. Assumptions:

1. **L-smoothness (Assumption 1):** Each F_k satisfies

$$F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{\theta}) + \nabla F_k(\boldsymbol{\theta})^\top(\boldsymbol{v} - \boldsymbol{\theta}) + \frac{L}{2}||\boldsymbol{v} - \boldsymbol{\theta}||^2.$$

2. **Strong convexity (Assumption 2):** Each F_k is μ-strongly convex, ensuring

$$F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{\theta}) + \nabla F_k(\boldsymbol{\theta})^\top(\boldsymbol{v} - \boldsymbol{\theta}) + \frac{\mu}{2}||\boldsymbol{v} - \boldsymbol{\theta}||^2.$$

3. **Bounded gradient variance (Assumption 3):** For each k,

$$\mathbb{E}[||\nabla F_k(\boldsymbol{\theta}_k^t, \xi_t^k) - \nabla F_k(\boldsymbol{\theta}_k^t)||^2] \leq \sigma_k^2.$$

4. **Bounded gradient norm (Assumption 4):** The expected squared norm of gradients is uniformly bounded,

$$\mathbb{E}[||\nabla F_k(\boldsymbol{\theta}_k^t, \xi_k^t)||^2] \leq G^2$$

5.2. Convergence Analysis of **ADAPT-FED**

Theorem 1: *Given the dynamic learning rates η_t and the assumptions in 5.1, the convergence behavior of the FL algorithm can be described by the following inequality bound:*

$$\mathbb{E}[||\boldsymbol{\theta}^T - \boldsymbol{\theta}^*||^2] \leq \prod_{t=0}^{T-1}(1-\eta\mu)\mathbb{E}[||\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*||^2] + \sum_{t=0}^{T-1}\eta^2\left(\sum_{k=1}^{N} w_k^2\sigma_k^2 + 6L\Gamma + 8\sum_{k=1}^{N} w_k(E-1)^2G^2\right) \tag{15}$$

where $\Gamma = F(\cdot;\boldsymbol{\theta})^* - \sum_{k=1}^{K} w_k F_k(\cdot;\boldsymbol{\theta}_k)^*$ quantifies the degree of data heterogeneity across clients. If the data is heterogeneous, then Γ is nonzero. Its magnitude reflects the heterogeneity of the data distribution, $\kappa = \frac{L}{\mu}$, $B = \sum_{k=1}^{K} w_k^2\sigma_k^2 + 6L\Gamma + 8(E-1)^2G^2$, E is the no. of local training epochs for each device k, and $\gamma = \max\{8\kappa, E\}$.

***Observation**: The convergence analysis of ADAPT-FED demonstrates the possible impact of dynamic rate adjustments on convergence efficiency and stability in FL settings.*

Remark (Decreasing Learning Rate): A decreasing η_t improves the convergence and stability by reducing the influence of the summation term in Equation 16, which includes contributions from gradient noise G and data heterogeneity Γ. G and Γ terms have a large impact on the slower convergence. When ADAPT-FED schedules η_t to be small, then the related terms in Equation 16 can be sufficiently suppressed; ADAPT-FED can accelerate the convergence (hence training stability and generalization) even under the impact of noisy gradients and heterogeneous FL environments. In the extensive experiments, results in §6 confirm that ADAPT-FED uses smaller values of η_t under training instability and heterogeneity regimes, leading to a significant generalization than other SOTA baselines.

***Takeaway**: The dynamic scheduling of η in the ADAPT-FED demonstrates the critical importance of adaptively managing η_t to realize fast convergence and stability. Decreasing η relative to historical stability levels promotes sustained and stable convergence, enhancing the algorithm's robustness to the challenges posed by data heterogeneity and gradient noise. This adaptive approach to*

Table 1. Generalization performance of ADAPT−FED versus baseline algorithms based on 20 clients across three datasets: (a) CIFAR10, (b) CIFAR100, and (c) UTK, respectively, $\eta_o = 0.1$.

	CIFAR-10								CIFAR-100								UTK							
Algorithm	Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)			
	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03
FedAvg	41.83	41.83	36.20	32.72	62.29	59.05	53.79	50.27	22.12	21.77	21.15	19.74	41.81	41.24	37.66	34.20	61.71	61.71	53.73	53.94	74.28	74.28	60.14	60.20
FedSAM	42.45	44.40	45.02	45.02	60.85	60.41	60.73	59.00	18.23	20.17	20.53	19.53	41.52	42.32	41.90	42.96	72.98	72.98	72.77	72.13	82.22	82.22	82.78	82.43
FedASAM	45.57	45.49	45.89	45.89	61.91	61.49	61.07	61.07	**22.29**	22.57	**21.91**	21.91	42.82	42.19	42.60	42.60	72.19	72.19	73.25	73.25	83.30	83.30	82.68	82.68
FedProx	**46.90**	42.64	37.10	33.85	61.50	57.51	54.37	50.03	21.93	21.98	21.15	20.12	42.10	41.36	37.62	34.15	61.70	61.70	53.64	53.26	74.01	74.01	60.93	59.08
FedAdagrad	45.24	41.83	36.20	32.74	62.30	59.10	53.79	50.28	22.13	21.62	21.15	19.74	41.83	41.35	37.77	34.50	61.89	61.89	53.73	53.95	74.30	74.30	60.41	60.43
FedAdam	45.24	41.83	36.20	32.74	62.29	59.05	53.91	50.47	22.13	21.62	21.15	19.74	41.56	41.78	37.61	34.25	61.89	61.89	53.73	53.95	74.28	74.28	60.14	60.20
FedYogi	45.24	41.83	36.20	32.74	62.47	59.44	53.27	50.14	22.13	21.62	21.15	19.74	41.82	41.56	37.69	34.20	61.89	61.89	53.73	53.95	74.91	74.48	60.27	60.20
ADAPT−FED (ours)	43.39	**49.99**	**50.78**	**49.55**	**72.01**	**73.88**	**73.88**	**74.60**	7.75	**23.08**	13.12	**24.29**	**48.41**	**52.48**	**54.88**	**56.13**	**79.20**	**79.20**	**76.22**	**76.22**	**84.95**	**84.95**	**85.13**	**84.13**

Table 2. Generalization performance of ADAPT−FED versus baseline algorithms based on 20 clients across three datasets: (a) CIFAR10, (b) CIFAR100, and (c) UTK, respectively, $\eta_o = 0.04$.

	CIFAR-10								CIFAR-100								UTK							
Algorithm	Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)			
	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03
FedAvg	50.63	48.59	44.62	40.86	67.48	66.28	62.84	59.72	33.09	32.61	30.82	29.26	50.10	49.61	47.38	45.07	70.58	70.58	53.86	53.08	80.37	80.37	59.29	58.26
FedSAM	50.47	50.73	51.03	51.11	67.59	66.66	67.25	67.42	32.97	33.32	34.09	33.24	50.75	50.71	50.61	51.03	75.59	75.59	76.39	76.54	83.75	83.75	83.75	83.59
FedASAM	51.26	51.48	51.23	51.23	67.06	67.98	67.63	67.63	34.10	33.94	33.85	33.85	51.04	51.04	50.74	50.74	76.19	76.19	77.03	77.03	83.69	83.69	83.81	83.81
FedProx	50.82	48.78	44.77	41.15	66.97	65.27	62.45	59.17	32.98	32.70	31.00	29.42	49.67	49.47	47.51	44.68	70.49	70.49	53.59	53.12	79.75	79.75	58.94	57.02
FedAdagrad	50.89	48.66	44.68	40.96	67.48	66.28	62.84	59.72	32.98	32.70	31.00	29.42	50.10	49.61	47.38	45.07	70.58	70.58	53.86	53.08	80.35	80.84	59.94	58.62
FedAdam	50.63	48.59	44.62	40.86	67.48	66.28	62.84	59.72	32.98	32.70	31.00	29.42	50.10	49.61	47.38	45.07	70.58	70.58	53.86	53.08	80.45	80.91	59.61	58.10
FedYogi	50.63	48.59	44.62	40.86	67.48	66.28	62.84	59.72	32.98	32.70	31.00	29.42	50.10	49.61	47.38	45.07	70.58	70.58	53.86	53.08	80.37	80.37	59.29	58.26
ADAPT−FED (ours)	**56.49**	**59.66**	**59.78**	**60.01**	**75.86**	**77.48**	**77.65**	**77.99**	**40.66**	**44.63**	**43.52**	**44.51**	**54.69**	**57.40**	**49.83**	**54.27**	**81.31**	**81.31**	**80.15**	**81.53**	**85.56**	**85.56**	**84.15**	**84.67**

learning rate management is beneficial for optimizing FL across real-world heterogeneous scenarios.

6. Experiments

We extensively evaluate ADAPT−FED's effectiveness in achieving generalization for FL with DP under different data heterogeneity levels while adhering to two constraints: maintaining performance stability; and faster convergence.

6.1. Experimental Setup

Models and datasets. We assess ADAPT−FED's efficacy using the setup in §C.2. We compare ADAPT−FED with SOTA baselines on the FL classification benchmarks datasets CIFAR10, CIFAR10, and UTK, examining generalization across different client partitions in FL.

Baselines: We evaluate ADAPT−FED across three key categories: 1) *FL baseline category* represented by FedAvg, serves as the standard learning scheme in FL. 2) *FL sharpness-aware category* includes FedSAM and FedASAM (Caldarola et al., 2022; Dai et al., 2023; Qu et al., 2022; Sun et al., 2023), which flattens minima in the loss landscape to improve model generalization. 3) *FL regularization category* includes FedProx(Mohri et al., 2019), which uses regularization techniques to minimize the divergence of local models for improved model generalization.

Hyperparameters For all evaluations on the benchmarks, we tuned the hyperparameters: μ for FedProx is tuned among two choices $\{0.1, 1\}$. For the parameters ρ, η of FedSAM and FedASAM, we borrow the same values that were used in the original paper by (Caldarola et al., 2022). Following the benchmark suggested in (Ahn et al., 2022; Cohen et al., 2021), we set the initial local learning rate as $\eta = \{0.1, 0.04\}$. We set the noniid-ness $\alpha = 0.3$ with DP $\sigma^2 = 0.01$ for all the evaluations in the main paper (other values of α and σ^2 are studied and presented in Appendix D). We present the empirical result across 10 and 20 clients.

6.2. Performance Evaluation

6.2.1. Generalization Analysis FL

ADAPT−FED significantly outperforms SOTA techniques in heterogeneous FL settings ($\alpha = 0.05$) on 20 clients, shown in Table 1 with $\eta_o = 0.1$ (refer to Table 4 in Appendix D.1 for results with 10 clients and detailed generalization evaluation). ADAPT−FED demonstrates generalization improvements of up to $+5.06\%$, $+14.79\%$, and $+7.79\%$ for CIFAR10, CIFAR100, and UTK respectively. These results confirm that ADAPT−FED effectively mitigates the strong training instability associated with heterogeneity, thereby enhancing generalization across clients. We believe these generalization gains are largely due to ADAPT−FED's use of stability-based adaptive learning rates, which directly address training instabilities caused by the GD learning algorithm. In contrast, existing techniques for training stability primarily focus on instabilities caused by discrepancies in local models due to data heterogeneity across clients, which does not inherently guarantee stability in GD learning. As heterogeneity is alleviated, as α increases from 0.05 to 0.3, generalization performance across all baselines improves due to the homogeneity of data distribution, which reduces discrepancies between local models across clients. Nevertheless, ADAPT−FED continues to demonstrate superior capability in enhancing generalization.

Takeaway: *ADAPT−FED improves generalization compared to SOTA techniques in non-IID FL environments.*

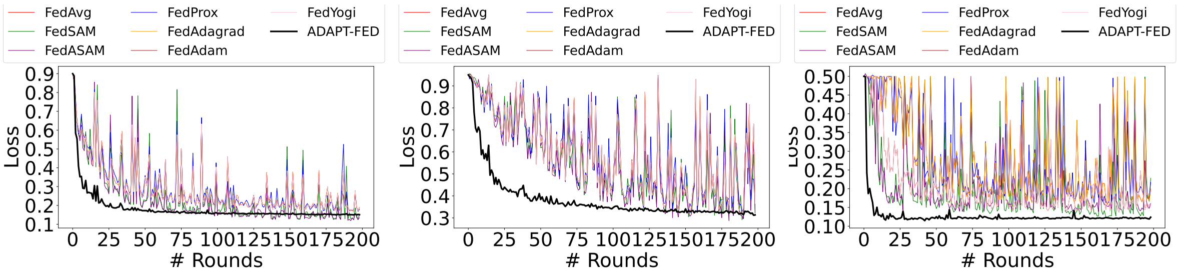

Figure 2. Convergence of the training loss of ADAPT-FED and baseline algorithms on 10 clients (CIFAR10, CIFAR100, and UTK, noniid-ness $\alpha = 0.3$) with DP $\sigma^2 = 0.01$, $\eta_o = 0.1$.

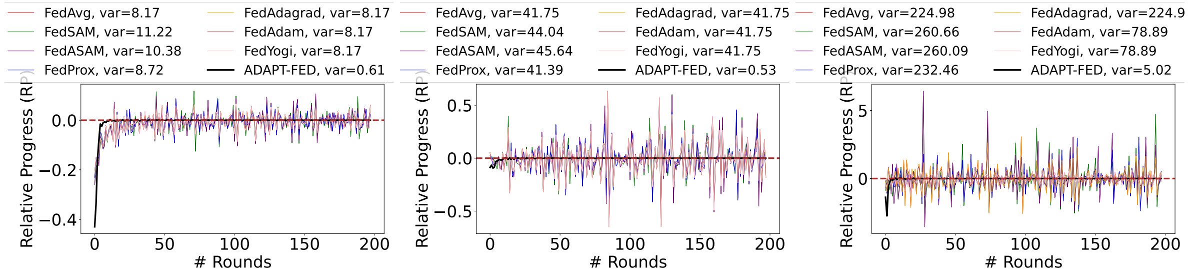

Figure 3. Stability of the training loss of ADAPT-FED and baseline algorithms on 10 clients across three datasets (CIFAR10, CIFAR100, and UTK, noniid-ness $\alpha = 0.3$, $\sigma^2 = 0.01$), $\eta_o = 0.1$.

6.2.2. Rate of Convergence Analysis

We compare ADAPT-FED with SOTA techniques to evaluate its ability to achieve faster convergence. On the CIFAR10, CIFAR100, and UTK datasets, ADAPT-FED demonstrates faster and more robust convergence than the baselines as shown in Figure 2. The improved convergence rate is a direct result of ADAPT-FED's use of adaptive learning rates, which specifically address training instabilities caused by the GD learning algorithm. In contrast, other techniques mainly focus on mitigating instability arising from discrepancies in local models due to data heterogeneity across clients, which does not inherently ensure stable learning. For detailed empirical study on rate of convergence, see Appendix D.2, similar observations can be made in Figure 20, where ADAPT-FED demonstrates robust convergence under heterogeneity in FL.

Takeaway: *ADAPT-FED leads to faster and more robust convergence compared to SOTA techniques in FL with DP.*

6.2.3. Training Stability Analysis

We compare ADAPT-FED with baselines to assess its effectiveness in achieving training stability. ADAPT-FED exhibits more stable training than baseline methods as shown in Figure 3. This enhanced stability is a direct result of ADAPT-FED 's use of adaptive learning rates, which address the issue of high gradient norms in GD. For detailed empirical stability analysis, see Appendix D.3.

Takeaway: *ADAPT-FED enhances training stability compared to SOTA baseline techniques in FL with DP.*

6.2.4. Analysis of Initial Learning Rate η_o

Table 2 shows the generalization performance of ADAPT-FED and SOTA baselines relative to small initial learning rate η_o. When the initial learning rate is low, ADAPT-FED still outperforms SOTA techniques in FL settings. However, we observe a degradation in overall generalization due to the slow learning effects of small η. For detailed ablation study on η see Appendix E.1.

Takeaway: *ADAPT-FED enhances generalization compared to SOTA techniques even in settings with low η_o.*

7. Conclusion

We introduce ADAPT-FED, an FL method that tackles the challenges of training instability and suboptimal generalization in FL. ADAPT-FED dynamically adjusts learning rates based on historical relative progress metrics, enhancing stability and improving the generalization across clients with heterogeneous data. We establish a detailed theoretical framework analyzing how ADAPT-FED mitigates the impacts of data heterogeneity and gradient noise on the learning process. Our theoretical findings are supported by empirical evaluations across various datasets, where ADAPT-FED consistently outperforms SOTA optimization methods in improving stability, accelerating convergence, and generalization. These improvements make ADAPT-FED a robust solution for practical, real-world FL applications.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Ahn, K., Zhang, J., and Sra, S. Understanding the unstable convergence of gradient descent. In *International Conference on Machine Learning*, pp. 247–257. PMLR, 2022.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Barto, A. G., Sutton, R. S., and Anderson, C. W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Caldarola, D., Caputo, B., and Ciccone, M. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pp. 654–672. Springer, 2022.

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Chandramoorthy, N., Loukas, A., Gatmiry, K., and Jegelka, S. On the generalization of learning algorithms that do not converge. *Advances in Neural Information Processing Systems*, 35:34241–34257, 2022.

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Dai, R., Yang, X., Sun, Y., Shen, L., Tian, X., Wang, M., and Zhang, Y. Fedgamma: Federated learning with global sharpness-aware minimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13733–13742, 2021.

Dwork, C. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pp. 1–12. Springer, 2006.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pp. 5905–5914. PMLR, 2021.

LeCun, Y., Simard, P., and Pearlmutter, B. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. *Advances in neural information processing systems*, 5, 1992.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019a.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019b.

Liu, Z., Lan, G., Stojkovic, J., Zhang, Y., Joe-Wong, C., and Gorlatova, M. Collabar: Edge-assisted collaborative image recognition for mobile augmented reality. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 301–312. IEEE, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Qu, Z., Li, X., Duan, R., Liu, Y., Tang, B., and Lu, Z. Generalized federated learning via sharpness aware minimization. In *International conference on machine learning*, pp. 18250–18280. PMLR, 2022.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Savchenko, A. V. Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*, pp. 119–124. IEEE, 2021.

Schaul, T., Zhang, S., and LeCun, Y. No more pesky learning rates. In *International conference on machine learning*, pp. 343–351. PMLR, 2013.

Sharma, N., Jain, V., and Mishra, A. An analysis of convolutional neural networks for image classification. *Procedia computer science*, 132:377–384, 2018.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.

Smith, L. N. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 464–472. IEEE, 2017.

Sun, Y., Shen, L., Chen, S., Ding, L., and Tao, D. Dynamic regularized sharpness aware minimization in federated learning: Approaching global consistency and smooth landscape. In *International Conference on Machine Learning*, pp. 32991–33013. PMLR, 2023.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020a.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pp. 7252–7261. PMLR, 2019.

Zeng, D., Liang, S., Hu, X., Wang, H., and Xu, Z. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023. URL http://jmlr.org/papers/v24/22-0440.html.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Appendix

We provide additional information for our paper, *ADAPT-FED: Adaptive Federated Optimization with Learning Stability*, in the following order:

- More Related Work (Appendix A)
- Additional Theoretical Analysis (Appendix B)
- Detailed Preliminaries and Problem Setup (Appendix C)
- Additional Experimental Results (Appendix D)
- Additional Ablation Study (Appendix E)

A. Related Work

Sharpness-aware FL focuses on adapting sharpness-aware optimization techniques (Caldarola et al., 2022; Dai et al., 2023; Qu et al., 2022; Sun et al., 2023) to FL to address the degradation of global model generalization under non-IID settings. Sharpness-aware optimization methods (Cha et al., 2021; Izmailov et al., 2018; Foret et al., 2020; Kwon et al., 2021) have successfully improved generalization in centralized learning by seeking flatter minima in the loss landscape, as shown by (Foret et al., 2020) and (Kwon et al., 2021). Given these improvements in centralized settings, the question arises: can these techniques enhance generalization in FL, especially with heterogeneous data across clients?

To answer this question, several adaptations have been proposed. FedSAM (Qu et al., 2022) was one of the first to apply SAM locally at each client, promoting convergence to flatter local minima and improving local generalization. However, SAM alone did not fully address client drift caused by data heterogeneity. To overcome this challenge, FedGAMMA (Dai et al., 2023) extended sharpness-aware optimization globally by aligning local updates with a flatter global minimum, incorporating client variance reduction to ensure local models aligned with the global objective and overcoming the limitations of simple aggregation methods like FedAvg.

Additionally, Stochastic Weight Averaging (SWA) and its dense variant, SWAD, were introduced to further enhance global model generalization. SWA, as demonstrated by (Izmailov et al., 2018), averages weights from multiple iterations to smooth the global loss surface. SWAD, which builds on SWA, reduces overfitting by densely sampling weights, resulting in flatter minima and better generalization, particularly in domain generalization tasks (Caldarola et al., 2022; Cha et al., 2021).

In conclusion, sharpness-aware optimization has proven highly effective in FL, addressing client drift and improving both convergence and generalization of the global model. By minimizing both loss and sharpness, these methods create smoother loss landscapes, enabling FL models to generalize more robustly across diverse and unseen data.

B. Theoretical Analysis

This section explores how ADAPT-FED behaves theoretically, focusing on its convergence rate. We provide theorems that set upper bounds on how quickly ADAPT-FED can converge. These theorems are essential for understanding how the learning rate affects convergence. Before that, we introduce the following assumptions: **Assumptions:**

1. **L-smoothness (Assumption 1):** Each F_k satisfies

$$F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{\theta}) + \nabla F_k(\boldsymbol{\theta})^\top (\boldsymbol{v} - \boldsymbol{\theta}) + \frac{L}{2}||\boldsymbol{v} - \boldsymbol{\theta}||^2.$$

2. **Strong convexity (Assumption 2):** Each F_k is μ-strongly convex, ensuring

$$F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{\theta}) + \nabla F_k(\boldsymbol{\theta})^\top (\boldsymbol{v} - \boldsymbol{\theta}) + \frac{\mu}{2}||\boldsymbol{v} - \boldsymbol{\theta}||^2.$$

3. **Bounded gradient variance (Assumption 3):** For each k,

$$\mathbb{E}[||\nabla F_k(\boldsymbol{\theta}_k^t, \xi_t^k) - \nabla F_k(\boldsymbol{\theta}_k^t)||^2] \leq \sigma_k^2.$$

B.1. Convergence Analysis of ADAPT-FED

Theorem 1: Given the dynamic learning rates η_t and the assumptions stated, the convergence behavior of the federated learning algorithm can be described by:

$$\mathbb{E}[||\boldsymbol{\theta}_T - \boldsymbol{\theta}^*||^2] \leq \prod_{t=0}^{T-1}(1-\eta_t\mu)\mathbb{E}[||\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*||^2] + \sum_{t=0}^{T-1}\eta_t^2\left(\sum_{k=1}^{N} w_k^2\sigma_k^2 + 6L\Gamma + 8\sum_{k=1}^{N} w_k(E-1)^2G^2\right) \tag{16}$$

where $\Gamma = F(\cdot;\boldsymbol{\theta})^* - \sum_{k=1}^{K} w_k F_k(\cdot;\boldsymbol{\theta}_k)^*$ (F^* and F_k^* are the minimum values of F^* and F_k^*, respectively) quantifies the degree of data heterogeneity; If the data are non-iid, then Γ is nonzero, and its magnitude reflects the heterogeneity of the data distribution, $\kappa = \frac{L}{\mu}$, $B = \sum_{k=1}^{K} w_k^2\sigma_k^2 + 6L\Gamma + 8(E-1)^2G^2$, E is the number of local training rounds/epochs for each device k, and $\gamma = \max\{8\kappa, E\}$.

Observation: The convergence analysis of ADAPT-FED under varying learning rate schedules highlights the nuanced impact of dynamic rate adjustments on the stability and efficiency of convergence in federated learning settings.

Case 1: Increasing Learning Rate As the learning rate (η_t) increases, the decay factor $(1-\eta_t\mu)$ in the convergence bound diminishes, which could inadvertently raise the upper bound of the expected error. This effect intensifies if $\eta_t\mu > 1$, potentially resulting in the product term becoming positive and causing the model to diverge.

Practical Impacts:

- **Potential for Faster Convergence:** An elevated η_t can accelerate the convergence process when the model parameters are significantly misaligned from the optimum and when the gradient noise is relatively low. This can be particularly beneficial in the early stages of training, where rapid progress towards the optimum is desired.
- **Risk of Instability:** Continuous increases in η_t without careful modulation may lead to overshooting and divergence, especially pronounced in non-IID data environments where the effects of data heterogeneity (Γ) are substantial.

Case 2: Decreasing Learning Rate: A decreasing η_t improves the stability by preserving the decay factor closer to zero, systematically mitigating the initial error's impact. This controlled reduction in the learning rate also curtails the influence of the summation term, which includes contributions from gradient noise and data heterogeneity.

Practical Impacts:

- **Enhanced Stability and Convergence:** Reducing η_t gradually ensures a more stable convergence trajectory, crucial in federated environments characterized by significant data heterogeneity (Γ) and high gradient variance (σ_k^2). This approach ensures that the learning process is not only stable but also progressively moves toward the optimal set of parameters.
- **Robustness to Non-IID Data:** The systematic decrease in η_t proves especially effective in non-IID settings, mitigating the adverse impacts of uneven data distributions and ensuring that the federated learning model remains robust across diverse and variable client data.

Takeaway: The dynamic scheduling of learning rates within the ADAPT-FED framework, as demonstrated by the theoretical analysis, underscores the critical importance of adaptively managing η_t to balance the trade-offs between fast convergence and stability. In increasing learning rate scenarios, while there is a potential for quick initial progress, there exists a significant risk of instability and divergence, particularly under non-IID conditions. Conversely, decreasing learning rates promote sustained and stable convergence, enhancing the algorithm's robustness to the challenges posed by data heterogeneity and gradient noise. This adaptive approach to learning rate management is crucial for optimizing federated learning processes, ensuring effective convergence across a spectrum of real-world scenarios where data distributions are inherently diverse and complex.

B.1.1. Additional Notation

Let $\boldsymbol{\theta}_k^t$ be the model parameter maintained in the k-th device at the t-th step. Let T_E be the set of global synchronization steps, i.e., $T_E = \{nE|n = 1, 2, \dots\}$. If $t+1 \notin T_E$, i.e., the time step to communication, FedAvg activates all devices.

Then, the update of FedAvg with partial active devices can be described as:

$$\boldsymbol{v_k^{t+1}} = \boldsymbol{\theta_k^t} - \eta_t \nabla F_k(\boldsymbol{\theta_k^t}, \xi_k^t), \tag{17}$$

$$\boldsymbol{\theta_k^{t+1}} = \begin{cases} \boldsymbol{v_k^{t+1}} & \text{if } t+1 \notin T_E, \\ \frac{\sum_{k=1}^N p_k \boldsymbol{v_k^{t+1}}}{\sum_{k=1}^N p_k} & \text{if } t+1 \in T_E. \end{cases} \tag{18}$$

Here, an additional variable $\boldsymbol{v_k^{t+1}}$ is introduced to represent the immediate result of one step GD update from $\boldsymbol{\theta_k^t}$. We interpret $\boldsymbol{\theta_k^{t+1}}$ as the parameter obtained after communication steps.

B.1.2. Key Lemmas

To convey the proof clearly, it would be necessary to prove certain useful lemmas. We refer the reader to (Li et al., 2019b) for detailed proofs.

Lemma 1 (Results of one step SGD): Assume Assumption 1 and 2. If $\eta_t \leq \frac{1}{4L}$, we have

$$\mathbb{E}[||\boldsymbol{v_k^{t+1}} - \boldsymbol{\theta}^*||^2] \leq (1-\eta_t\mu)\mathbb{E}[||\boldsymbol{\theta}_t - \boldsymbol{\theta}^*||^2] + \eta_t^2\mathbb{E}[||g_t - \bar{g}_t||^2] + 6L\eta_t^2\Gamma + 2\eta_t^2\sum_{k=1}^N p_k\mathbb{E}[||\boldsymbol{\theta}_t - \boldsymbol{\theta}_k^t||^2],$$

where $\Gamma = F^* - \sum_{k=1}^N p_k F_k^* \geq 0$.

Lemma 2 (Bounding the variance): Assume Assumption 3 holds. It follows that

$$\mathbb{E}[||g_t - \bar{g}_t||^2] \leq \sum_{k=1}^N p_k^2\sigma_k^2.$$

Lemma 3 (Bounding the divergence of $\{w_k^t\}$): Assume Assumption 4, that η_t is non-increasing and $\eta_t \leq 2\eta_{t+E}$ for all $t \geq 0$. It follows that

$$\mathbb{E}\left[\sum_{k=1}^N p_k||\boldsymbol{\theta}_t - \boldsymbol{\theta}_k^t||^2\right] \leq 4\eta_t^2(E-1)^2G^2.$$

1. **Starting Point and Gradient Update:**

$$\mathbb{E}[||\mathbf{v}_{t+1} - \mathbf{w}^*||^2] \leq (1-\eta_t\mu)\mathbb{E}[||\mathbf{w}_t - \mathbf{w}^*||^2] + \eta_t^2\mathbb{E}[||\mathbf{g}_t - \mathbf{g}_{t,k}||^2] + 6L\eta_t^2\Gamma + 2\eta_t^2\sum_{k=1}^N p_k\mathbb{E}[||\mathbf{w}_t - \mathbf{w}_{t,k}||^2]$$

2. **Using Lemma 2 and Lemma 3:**

$$\mathbb{E}[||\mathbf{g}_t - \mathbf{g}_{t,k}||^2] = \sum_{k=1}^N p_k^2\sigma_k^2$$

$$\mathbb{E}[||\mathbf{w}_t - \mathbf{w}_{t,k}||^2] = 4\eta_t^2(E-1)^2G^2$$

3. **Combining Errors and Simplifying:**

$$\mathbb{E}[||\mathbf{v}_{t+1} - \mathbf{w}^*||^2] \leq (1-\eta_t\mu)\mathbb{E}[||\mathbf{w}_t - \mathbf{w}^*||^2] + \eta_t^2(\sum_{k=1}^N p_k^2\sigma_k^2 + 6L\Gamma + 8\sum_{k=1}^N p_k(E-1)^2G^2)$$

4. **Recurrence Relation and Final Convergence Bound:**

$$\mathbb{E}[||\mathbf{w}_T - \mathbf{w}^*||^2] \leq \prod_{t=0}^{T-1}(1-\eta_t\mu)\mathbb{E}[||\mathbf{w}_0 - \mathbf{w}^*||^2] + \sum_{t=0}^{T-1}\eta_t^2(\sum_{k=1}^N p_k^2\sigma_k^2 + 6L\Gamma + 8\sum_{k=1}^N p_k(E-1)^2G^2)$$

C. Preliminaries and Problem Setup

The purpose of this section is to provide additional *Preliminaries and Problem Setup* details that are dropped due to the limited space of the main paper. It includes the plots of training instability and the Experimental Setup for the CIFAR10, CIFAR100, and UTK datasets.

C.1. Training Instability Study: Results Overview

We measure the training instability (relative progresss RP) and its impacts on generalization (utility/accuracy) in FL across CIFAR10 (Figure 4, Figure 5, Figure 6, and Figure 7), UTK (Figure 8, Figure 9, Figure 10, and Figure 11) and CIFAR100 (Figure 12, Figure 13, Figure 14, and Figure 15) datasets. As depicted in most of the graphs on the left, the RP, which measures the stability of training, shows significant variance across training rounds. This variance is persistent and positive, indicating that the learning process is not stable. This instability is further compounded as the differential privacy level increases. The increasing variance in RP with higher levels of DP suggests that the noise added for privacy protection is disrupting the learning process, making it harder for the model to converge consistently. This behavior demonstrates the challenge of balancing model privacy with learning efficacy in FL environments.

The middle graphs show RP variance against different levels of differential privacy and confirm that as privacy constraints tighten (σ^2 increases), the overall variability in model performance also increases. The trend line indicates a clear positive correlation between RP variance and the privacy level, highlighting a direct impact of enhanced privacy measures on learning stability. Higher differential privacy levels introduce more noise into the training process, which can lead to larger updates that are less about the true gradient direction and more about compensating for the noise. This can cause the training process to become unstable, as shown by the rising RP variance.

The right most graphs illustrates that with increasing DP, not only do gradient norms increase, but also accuracy decreases significantly. This suggests that the model is struggling to generalize effectively under higher training instability. Larger gradient norms indicate more substantial updates during training, which can overshoot optimal points due to the high noise levels introduced by DP. This is likely contributing to the observed decrease in model accuracy as DP levels increase, illustrating the difficulty in navigating the trade-off between privacy and generalization.

These detailed analyses and observations demonstrate the complex interplay between privacy, stability, and generalization in FL. By fine-tuning the learning rates and understanding the impact of differential privacy on learning dynamics, it is possible to improve both the stability and generalization of models trained under privacy constraints.

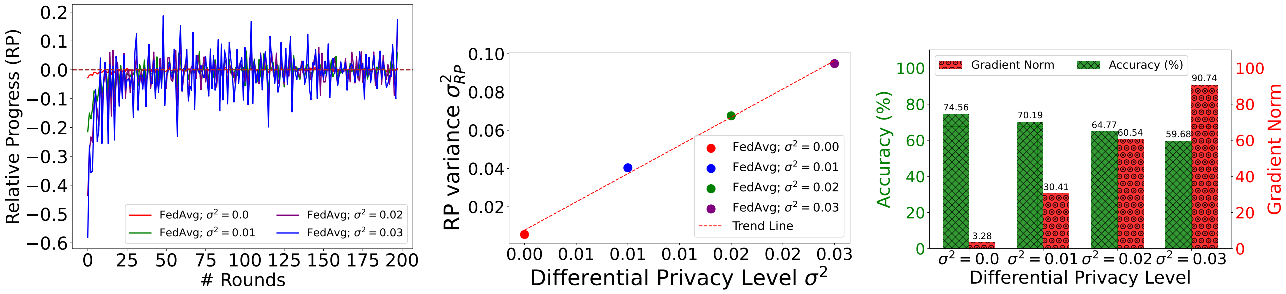

Figure 4. Correlation of DP noise with training instability in a 10-client CIFAR10 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

C.2. Experimental Setup

C.2.1. Datasets and Model Architectures

Dataset: We analyze two widely utilized image classification datasets for federated learning: CIFAR10 (Krizhevsky et al., 2009) and CIFAR100 (Sharma et al., 2018), along with the UTK (Savchenko, 2021) image classification dataset. The benchmarks for these datasets in a federated learning context are adopted from established benchmarks based on CIFAR-10/100, as proposed by (Foret et al., 2020). Each dataset is allocated among $K \in \{10, 20\}$ clients, employing a Dirichlet distribution-based approach for data distribution as done in (Zeng et al., 2023). The resultant data partitions

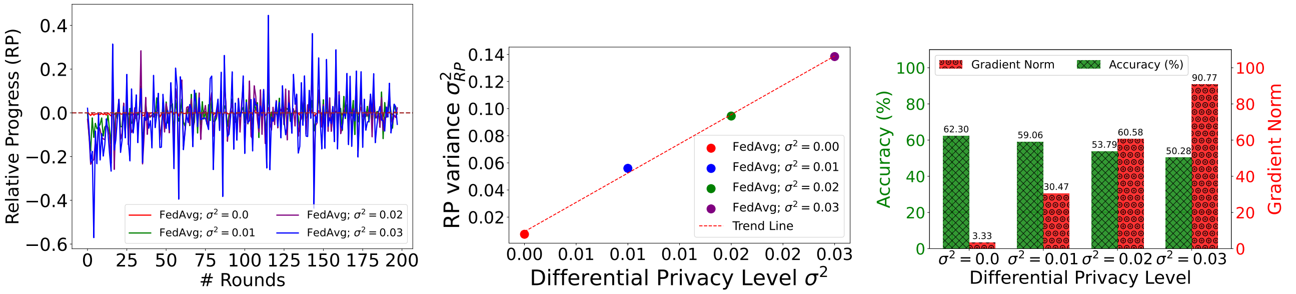

Figure 5. Correlation of DP noise with training instability in a 20-client CIFAR10 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

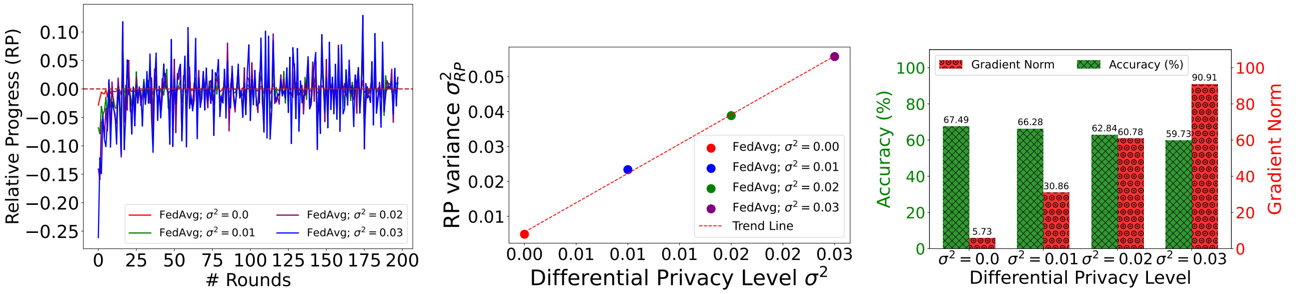

Figure 6. Correlation of DP noise with training instability in a 20-client CIFAR10 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.04$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

Table 3. Datasets and Clients

Dataset	Task	Total Clients	Total Samples	Training Samples	Test Samples
CIFAR10	Image classification	10,20	60,000	50,000	10,000
CIFAR100	Image classification	10,20	60,000	50,000	10,000
UTK	Image classification	10,20	23,708	19,208	4,500

are shown in Figure 16, Figure 17, Figure 18, and Figure 19. Here, each client's prior distribution follows a multinomial distribution derived from a symmetric Dirichlet distribution with parameter α. As α approaches infinity, the data distribution among clients approximates an IID scenario. Conversely, a reduction in α, moving towards zero, shifts the distribution towards a non-IID scenario. We explore different scenarios with $\alpha \in \{0.05, 0.3\}$ across the CIFAR10, CIFAR100, and UTK datasets.

Model architecture: By following the backbone architecture of the unstable convergence of gradient descent work (Ahn et al., 2022; Cohen et al., 2021). Specifically, we use GD to train a VGG (with batch normalization) neural network (Ding et al., 2021). For a fair comparison, we use the same backbone architecture for all different types of methods for all evaluations. Also, the same architecture is identically used for the two CIFAR-10/100 benchmarks. Noteworthy, we added ResNet backbone for UTK dataset because of poor performance relative to VGG on this dataset.

C.2.2. Data Pre-processing (CIFAR-10 and CIFAR-100).

All training and test input images of size 32×32 pixels are first padded by 4 pixels on each side, then randomly cropped back to 32×32 pixels. This technique helps the model become invariant to small translations of the input image. Each image is flipped horizontally with a probability of 0.5. This step increases the diversity of the training data and helps prevent overfitting by simulating different viewing angles. After converting the image to a tensor, pixel values are normalized using the dataset-specific mean $(0.4914, 0.4822, 0.4465)$ and standard deviation $(0.2023, 0.1994, 0.2010)$. This normalization facilitates faster convergence by scaling the input features to have zero mean and unit variance.

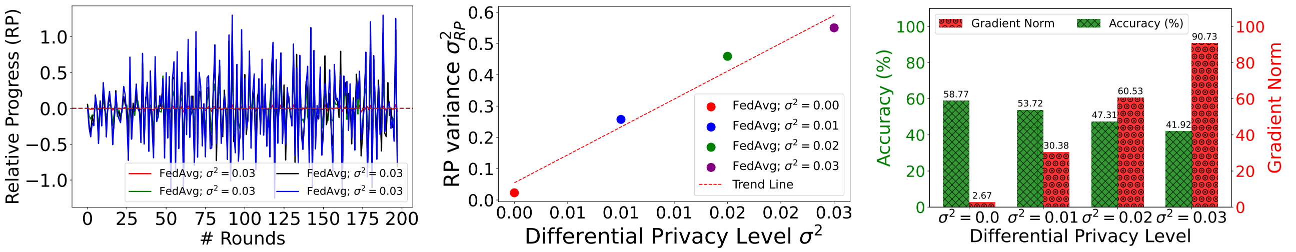

Figure 7. Correlation of DP noise with training instability in a 10-client CIFAR10 setup ($\alpha = 0.05$ non-iid). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

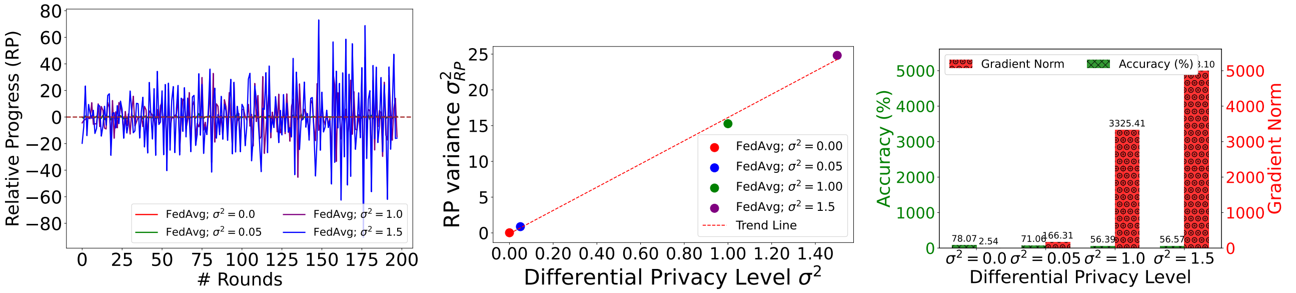

Figure 8. Correlation of DP noise with training instability in a 10-client UTK setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

C.2.3. Data Pre-processing (UTK).

All training and test input images are resized to 32×32 pixels, standardizing the input size across all images and making it suitable for processing by the model designed for CIFAR datasets. Pixel values are normalized using the mean (0.49) and standard deviation (0.23). This dataset appears to have grayscale images (indicated by a single channel mean and standard deviation), and normalization adjusts the pixel intensity distribution similarly to CIFAR datasets. Images undergo the same resizing to 32×32 pixels and are normalized using the same values as the training images. Consistent image size and normalization between the training and testing phases help in evaluating the model's performance accurately.

D. Additional Experimental Results

Here, we provide the additional experimental results that are dropped due to the limited space of the main paper. It includes the the plots for *generalization analysis*, *rate of convergence analysis*, and *training stability analysis* using for the CIFAR10, CIFAR100, and UTK datasets.

D.1. Generalization Analysis FL

We conduct a thorough analysis of ADAPT−FED's generalization performance against various baseline FL algorithms. Our primary goal is to assess the efficacy of ADAPT−FED in generalizing under diverse privacy settings and heterogeneous data distributions. Generalization analyses are performed on three widely recognized datasets: CIFAR10, CIFAR100, and UTK, comparing ADAPT−FED with several SOTA FL algorithms, including FedAvg, FedProx, FedAdagrad, FedYogi, FedSAM, and FedASAM.

D.2. Rate of Convergence Analysis

We conduct a thorough analysis of ADAPT−FED's convergence performance against various baseline FL algorithms. Our primary goal is to assess the efficacy of ADAPT−FED in achieving faster and more stable convergence rates, particularly under diverse privacy settings and heterogeneous data distributions. Convergence analyses are performed on three widely recognized datasets: CIFAR-10, CIFAR-100, and UTK, comparing ADAPT−FED with SOTA FL algorithms, including

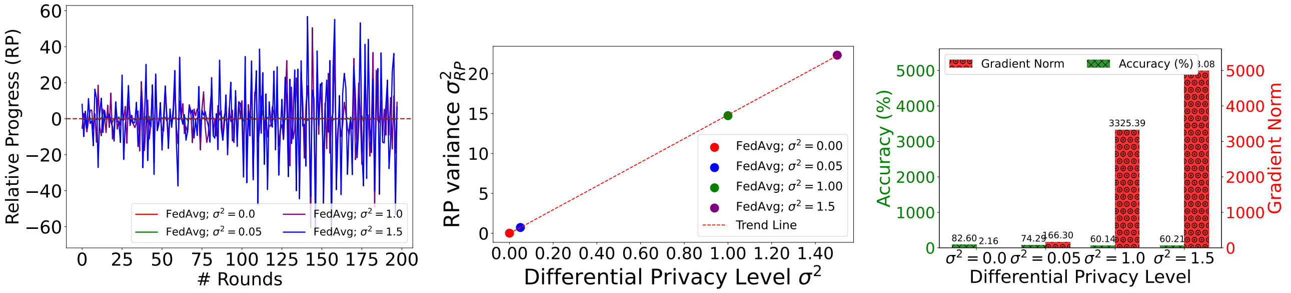

Figure 9. Correlation of DP noise with training instability in a 20-client UTK setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

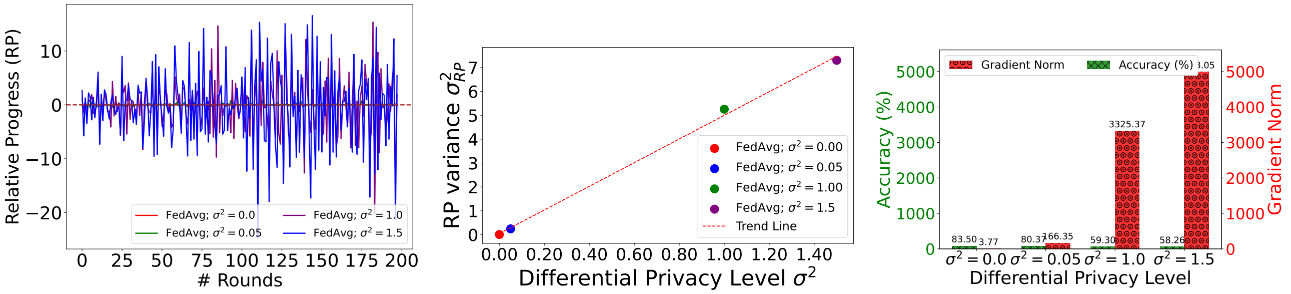

Figure 10. Correlation of DP noise with training instability in a 20-client UTK setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.04$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

FedAvg, FedProx, FedAdagrad, FedYogi, FedSAM, and FedASAM.

As illustrated in Figure 21, Figure 22, Figure 23, Figure 24, Figure 25, Figure 30, Figure 31, Figure 32, Figure 33, Figure 26, Figure 27, Figure 28, and Figure 29, ADAPT−FED demonstrates robust convergence in settings with data heterogeneity ($\alpha = 0.3$). This performance is indicative of the adaptive learning rate mechanism within ADAPT−FED, which fine-tunes the updates based on the observed instability and heterogeneity levels, thereby enhancing the convergence rate.

ADAPT−FED utilizes an innovative adaptive learning rate strategy that dynamically adjusts based on the model's performance from one iteration to the next. This approach addresses not only the variability introduced by differential privacy but also the challenges posed by non-IID data across clients. Unlike traditional methods that apply uniform updates, ADAPT−FED tailors the learning rates to mitigate the impact of high gradient variances and ensures consistent learning progress.

Despite the advancements in FL algorithms to handle heterogeneous data and privacy constraints, our results reveal a persistent challenge in Figure 23, Figure 24, Figure 32, Figure 33, Figure 28, Figure 29: stability under high data heterogeneity and strict privacy conditions. ADAPT−FED's adaptive learning rate mechanism, designed to stabilize the training process, struggles against the compounded noise introduced by increased differential privacy levels. The higher the σ^2, the more pronounced are the oscillations in the loss, suggesting that differential privacy noise can significantly

Table 4. Generalization performance of ADAPT−FED versus baseline algorithms based on 10 clients across three datasets: (a) CIFAR10, (b) CIFAR100, and (c) UTK, respectively, $\eta_o = 0.1$. ADAPT−FED outperforms the baseline algorithms in terms of generalization performance across datasets.

	CIFAR-10								CIFAR-100								UTK							
Algorithm	Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)				Dir. ($\alpha = 0.05$, non-IID)				Dir. ($\alpha = 0.3$)			
	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03	0.0	0.01	0.02	0.03
FedAvg	58.76	53.72	47.31	41.91	74.55	70.18	64.76	59.68	37.73	37.03	33.94	30.89	41.21	40.91	37.48	34.84	64.11	64.11	55.18	54.98	78.89	78.89	64.87	64.83
FedSAM	57.93	58.08	58.67	58.74	74.80	74.18	74.61	74.90	37.43	38.49	38.57	38.17	42.18	42.64	42.81	43.89	73.28	73.28	73.58	73.35	79.06	79.06	79.10	78.83
FedASAM	59.09	58.78	58.74	58.74	75.40	74.91	74.51	74.51	38.56	37.83	37.99	37.99	43.39	43.12	43.28	43.28	73.31	73.31	74.07	74.07	78.63	78.63	79.48	79.48
FedProx	59.86	55.27	49.14	42.82	73.54	69.48	64.36	59.91	37.90	36.37	34.30	30.97	42.36	40.66	36.87	34.84	65.42	65.42	55.05	54.68	71.40	71.40	57.03	56.65
FedAdagrad	58.76	53.72	47.31	41.91	74.55	70.18	64.76	59.68	37.73	37.03	33.94	30.89	41.21	40.91	37.48	34.84	64.11	64.11	55.18	54.98	71.06	71.06	56.38	56.56
FedAdam	58.76	53.72	47.31	41.91	74.55	70.18	64.76	59.68	37.73	37.03	33.94	30.89	41.21	40.91	37.48	34.84	64.11	64.11	55.18	54.98	78.89	78.89	64.87	64.83
FedYogi	58.76	53.72	47.31	41.91	74.55	70.18	64.76	59.68	37.73	37.03	33.94	30.89	41.21	40.91	37.48	34.84	64.11	64.11	55.18	54.98	78.89	78.89	64.87	64.83
ADAPT−FED (ours)	**63.02**	**65.18**	**65.39**	**65.83**	**80.46**	**81.33**	**81.24**	**81.75**	**51.44**	**53.59**	**54.26**	**54.34**	**58.18**	**61.25**	**61.38**	**60.39**	**75.05**	**75.05**	**75.05**	**74.31**	**86.85**	**86.85**	**86.49**	**86.86**

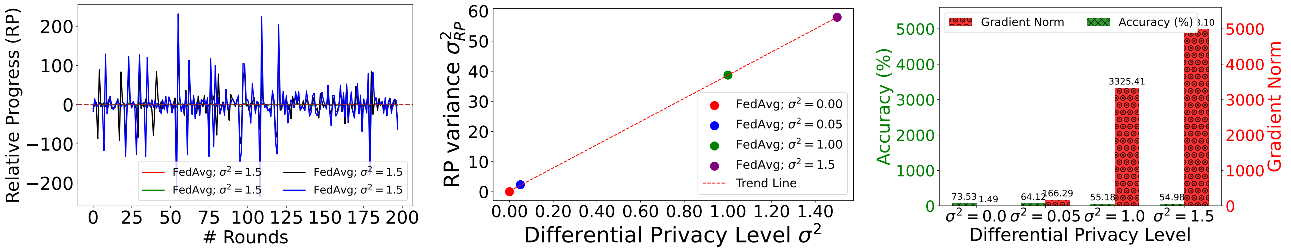

Figure 11. Correlation of DP noise with training instability in a 10-client UTK setup ($\alpha = 0.05$ non-iid). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

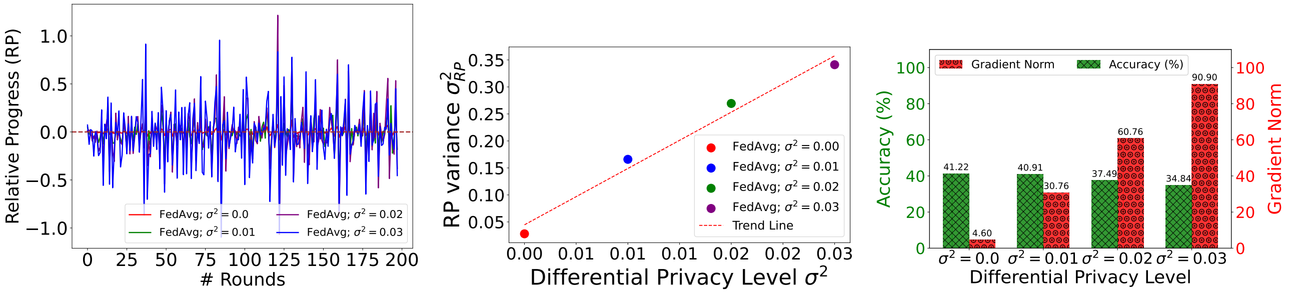

Figure 12. Correlation of DP noise with training instability in a 10-client CIFAR100 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

overshadow the underlying data distribution, misleading the adaptive learning rate adjustments. The high data heterogeneity exacerbates the difficulty of achieving stable convergence. The variability in local datasets leads to highly divergent local model updates, which, when aggregated, can destabilize the global model learning process. This issue is highlighted in our experiments, where even ADAPT−FED shows fluctuations in convergence under extreme conditions.

D.3. Training Stability Analysis

We evaluate the training stability of ADAPT−FED in comparison to various baseline FL algorithms. These experiments are conducted across the CIFAR10, CIFAR100, and UTK datasets, with emphasis on differential privacy settings and data heterogeneity.

Figure 34, Figure 35, Figure 36, Figure 37, Figure 38, Figure 42, Figure 43, Figure 44, Figure 39, Figure 40 and Figure 41, illustrate the relative progress (RP) across 200 training round under varying conditions. These figures capture the effectiveness of ADAPT−FED's adaptive learning rate mechanism in enhancing training stability compared to traditional FL approaches. This strategy significantly reduces the oscillations in RP, particularly evident in scenarios with high differential privacy levels and heterogeneous data distributions. In Figure Figure 3, ADAPT−FED maintains a lower variance in RP compared to baselines like FedAvg and FedProx, indicating more consistent progress and reduced training disruptions despite the introduction of noise through differential privacy. Figure 34, Figure 35, Figure 36, Figure 37, Figure 38, Figure 42, Figure 43, Figure 44, Figure 39, Figure 40 and Figure 41, highlight ADAPT−FED's ability to sustain lower variability in RP even under severe data heterogeneity, reflecting its capacity to adapt to heterogeneous data distributions effectively. ADAPT−FED employs an adaptive learning rate that dynamically adjusts based on the observed gradient norms.

While baseline algorithms exhibit increased RP fluctuations, indicating struggles with gradient noise and data heterogeneity, ADAPT−FED demonstrates a markedly smoother convergence curve. This distinction demonstrates the limitations of SOTA methods that do not account dynamically for changing gradient scales, often leading to inefficient learning rates that either overstep or underutilize the learning potential of the model.

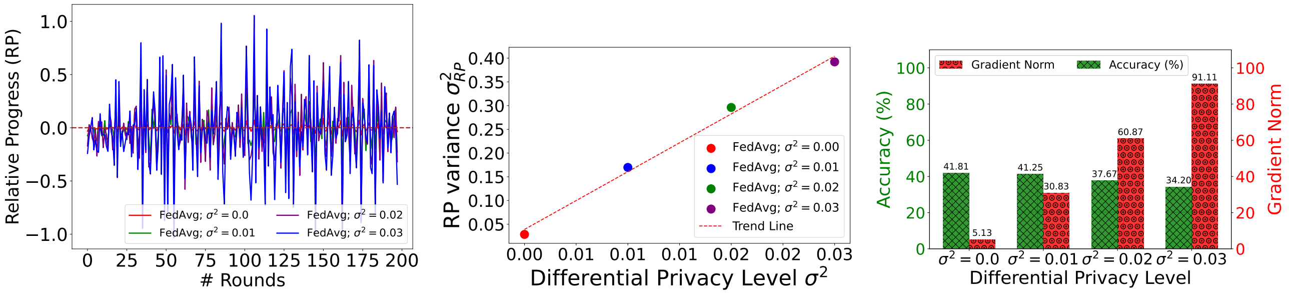

Figure 13. Correlation of DP noise with training instability in a 20-client CIFAR100 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.1$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

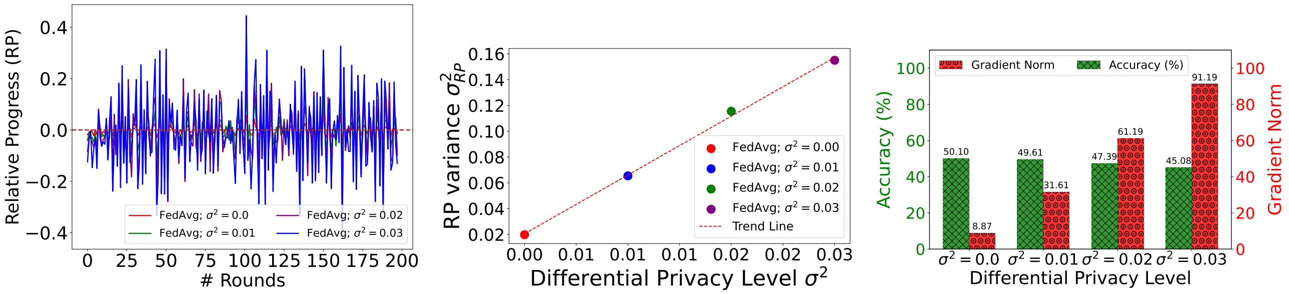

Figure 14. Correlation of DP noise with training instability in a 20-client CIFAR100 setup ($\alpha = 0.3$ non-iid, $\eta_0 = 0.04$). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

E. Additional Ablation Study

Here, we provide the additional experimental analyses that are dropped due to the limited space of the main paper. These analyses include the plots for ADAPT−FED'S *learning rate scheduling dynamics*.

E.1. Learning Rate Scheduling Dynamics Analysis

Figure 46 illustrates the behavior of the learning rate η for ADAPT−FED across different levels of differential privacy (σ^2) during the training process. These dynamics are important in understanding the adaptive nature of ADAPT−FED's learning rate adjustment mechanism.

As shown in Figure 46, the learning rate sharply decreases within the initial rounds, regardless of the differential privacy setting. This rapid decay is part of ADAPT−FED's strategy to quickly converge to a stable state before refining its model parameters under decreasing learning rates, thereby mitigating the risk of overshooting minima due to high initial rates.

Table 2 reveals that despite the aggressive reduction in learning rate, ADAPT−FED consistently outperforms SOTA baselines in terms of generalization across all datasets when starting from a high initial learning rate $\eta_{k,o}$. This demonstrates the effectiveness of ADAPT−FED's adaptive learning rate mechanism, which, although it starts high and decays rapidly, manages to maintain superior learning quality by optimally balancing exploration and exploitation phases during training.

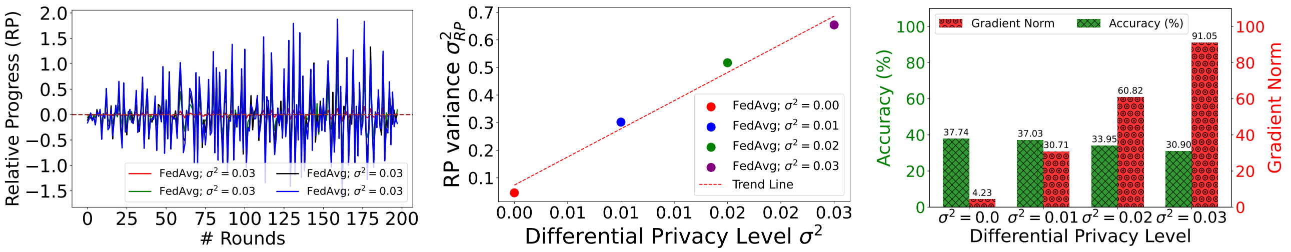

Figure 15. Correlation of DP noise with training instability in a 10-client CIFAR100 setup ($\alpha = 0.05$ non-iid). Increased DP noise elevates instability, as shown by RP value variance, causing larger gradient norms and lower accuracy.

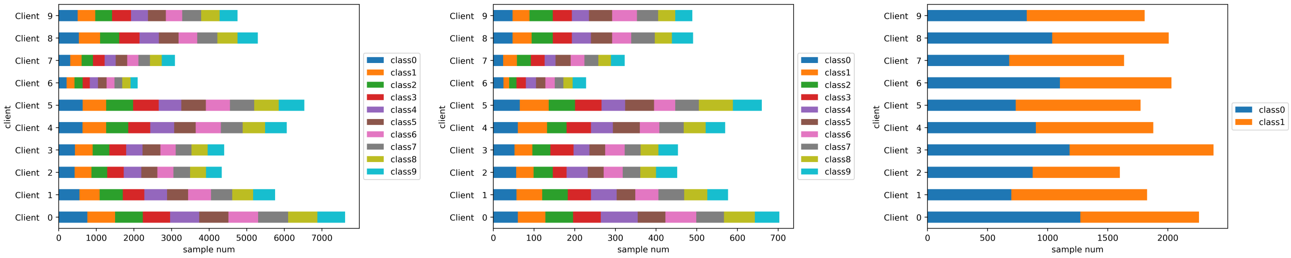

Figure 16. Noniid partition used in (Yurochkin et al., 2019) and (Wang et al., 2020a). The number of CIFAR10, CIFAR1OO, and UTK data points and class proportions are unbalanced. Samples will be partitioned into 10 clients by sampling $\alpha = 0.3$.

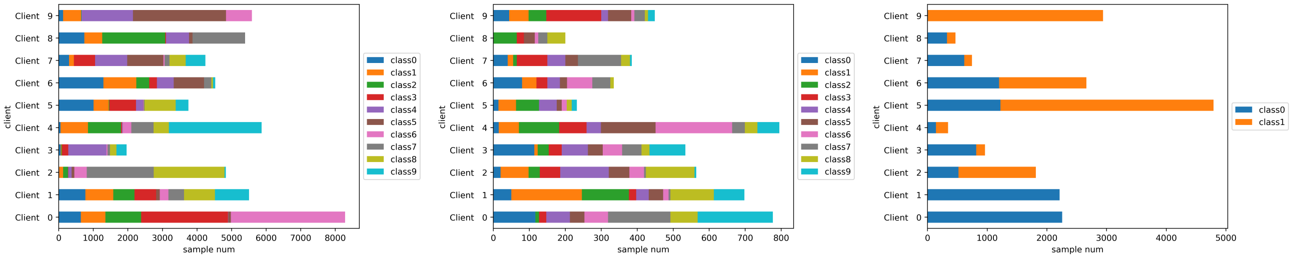

Figure 17. Noniid partition used in (Yurochkin et al., 2019) and (Wang et al., 2020a). Number of CIFAR10, CIFAR1OO, and UTK data points and class proportions are unbalanced. Samples will be partitioned into 10 clients by sampling $\alpha = 0.05$.

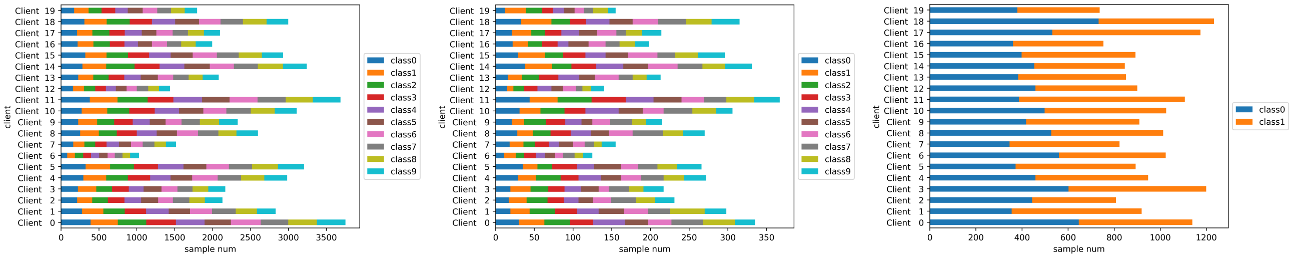

Figure 18. Noniid partition used in (Yurochkin et al., 2019) and (Wang et al., 2020a). The number of CIFAR10, CIFAR1OO, and UTK data points and class proportions are unbalanced. Samples will be partitioned into 20 clients by sampling $\alpha = 0.3$.

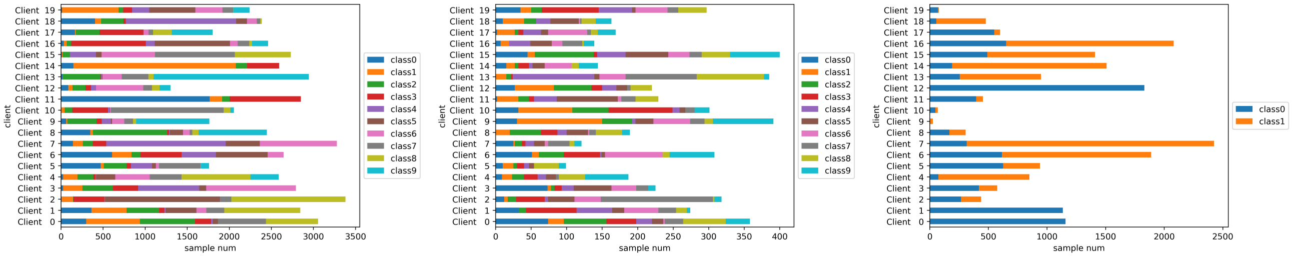

Figure 19. Noniid partition used in (Yurochkin et al., 2019) and (Wang et al., 2020a). The number of CIFAR10, CIFAR1OO, and UTK data points and class proportions are unbalanced. Samples will be partitioned into 20 clients by sampling $\alpha = 0.05$.

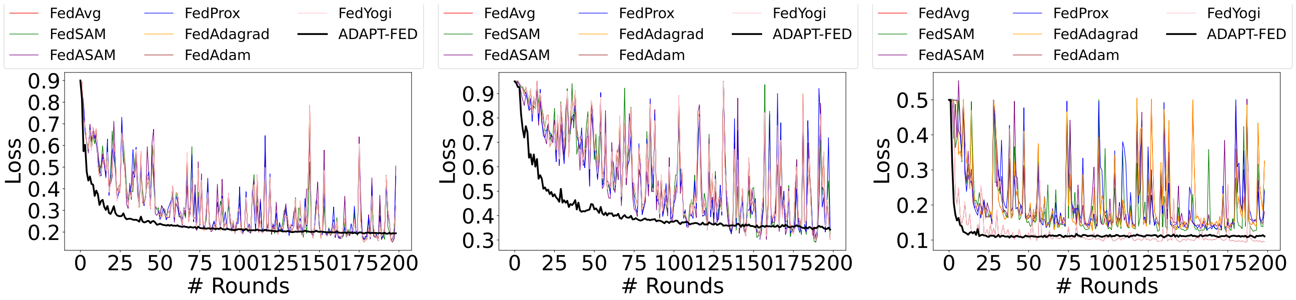

Figure 20. Convergence of the training loss of ADAPT−FED and SOTA algorithms on 10 clients (CIFAR10, CIFAR100, and UTK nonoiid-ness, $\alpha = 0.3$) with DP $\sigma^2 = 0.0$, $\eta_o = 0.04$.

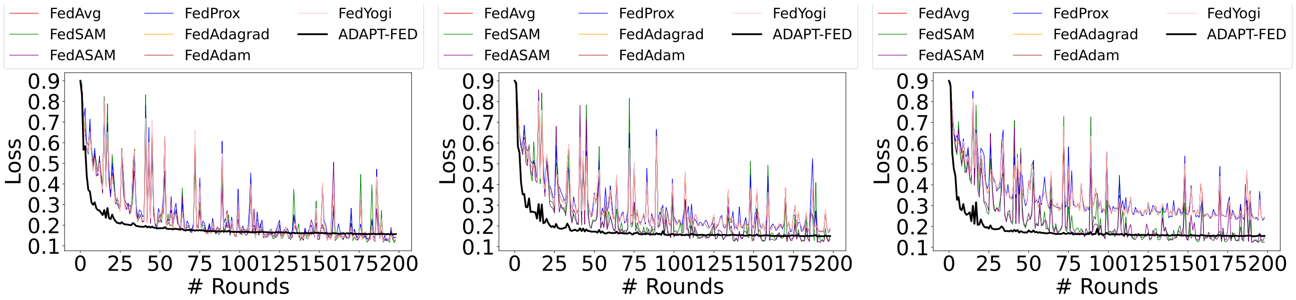

Figure 21. Convergence of the training loss of ADAPT−FED and baseline algorithms on 10 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

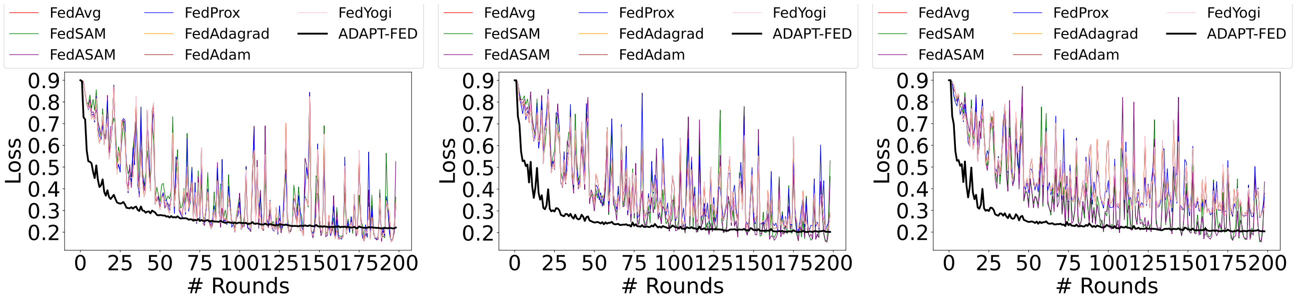

Figure 22. Convergence of the training loss of ADAPT−FED and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

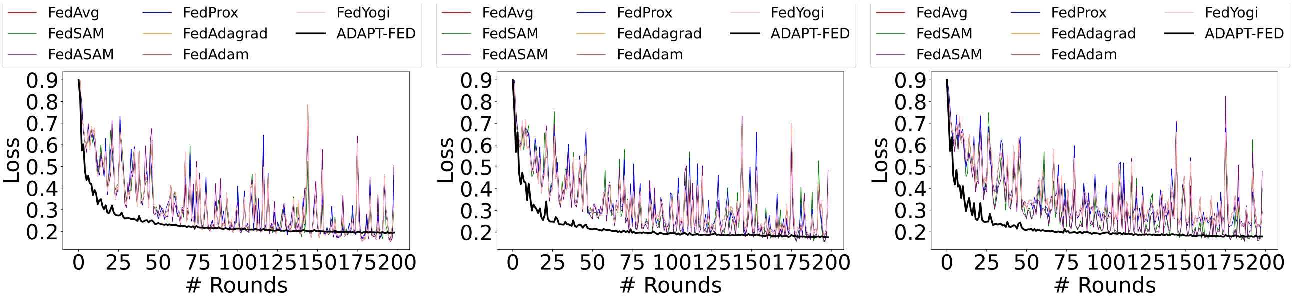

Figure 23. Convergence of the training loss of `ADAPT-FED` and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta = 0.04$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.`ADAPT-FED` exhibits faster and more robust convergence.

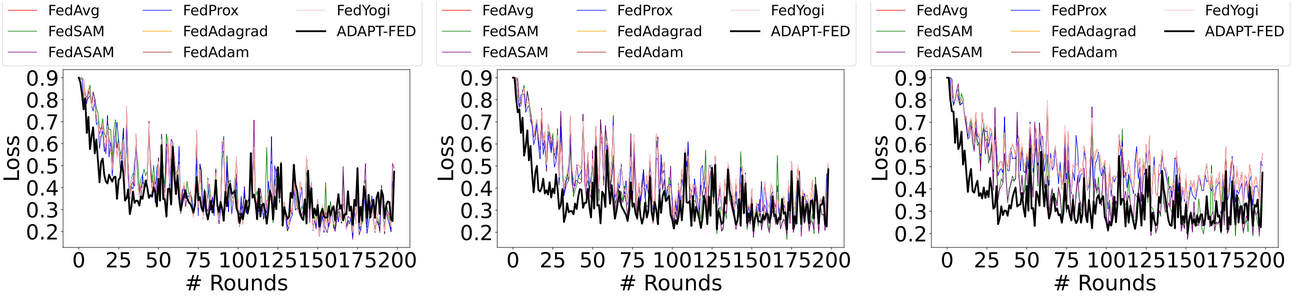

Figure 24. Convergence of the training loss of `ADAPT-FED` and baseline algorithms on 10 clients (CIFAR10 noniid-ness $\alpha = 0.05$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.`ADAPT-FED` exhibits faster and more robust convergence.

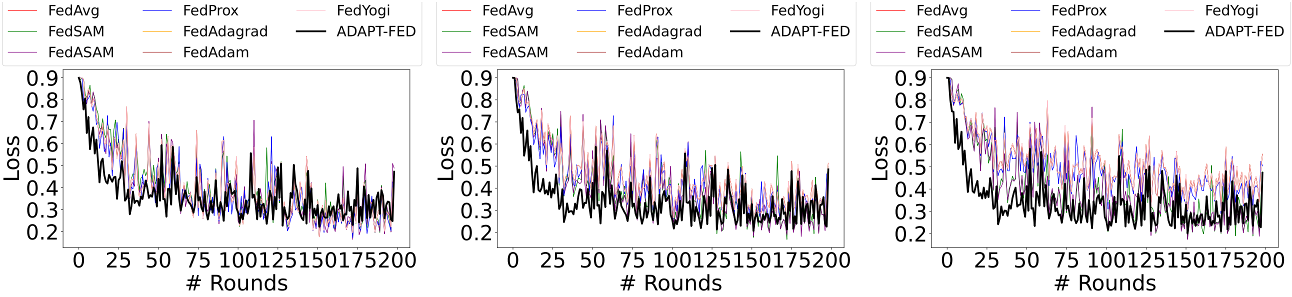

Figure 25. Convergence of the training loss of `ADAPT-FED` and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.05$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.`ADAPT-FED` exhibits faster and more robust convergence.

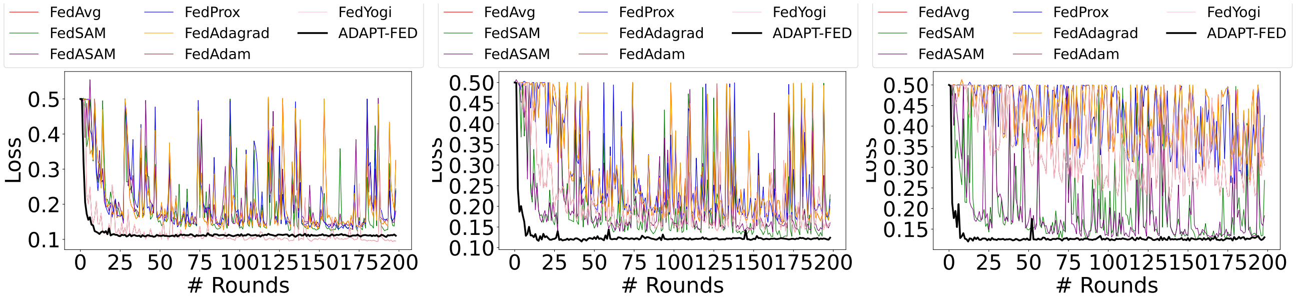

Figure 26. Convergence of the training loss of ADAPT−FED and baseline algorithms on 10 clients (UTK noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

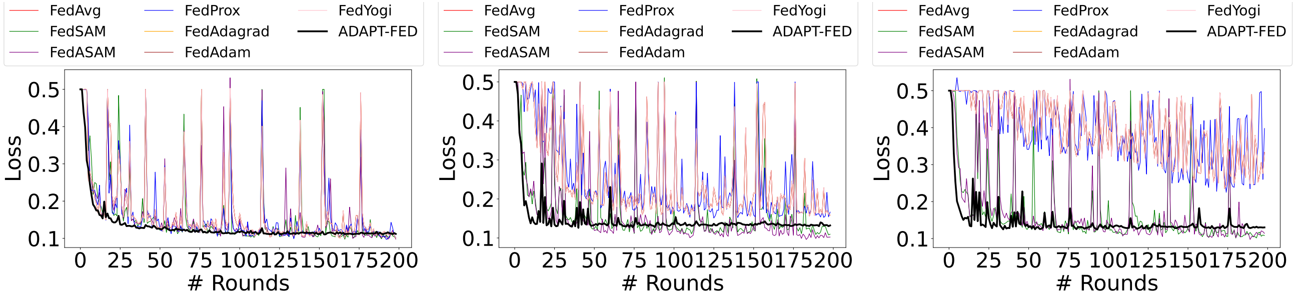

Figure 27. Convergence of the training loss of ADAPT−FED and baseline algorithms on 20 clients (UTK noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

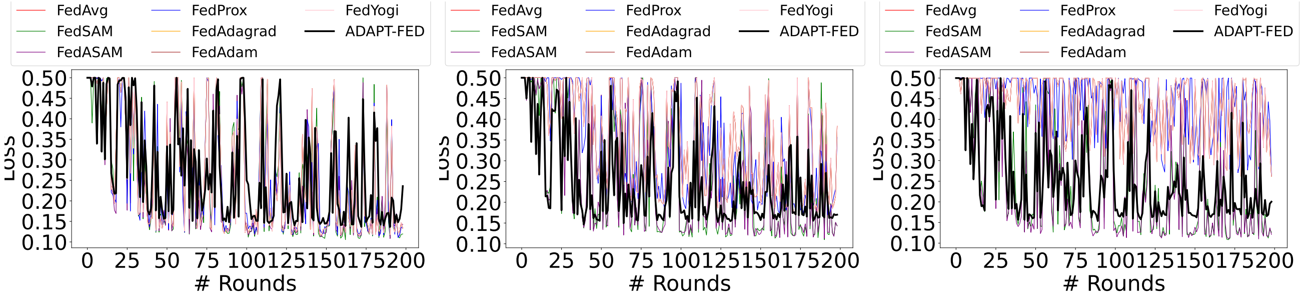

Figure 28. Convergence of the training loss of ADAPT−FED and baseline algorithms on 10 clients (UTK noniid-ness $\alpha = 0.05$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

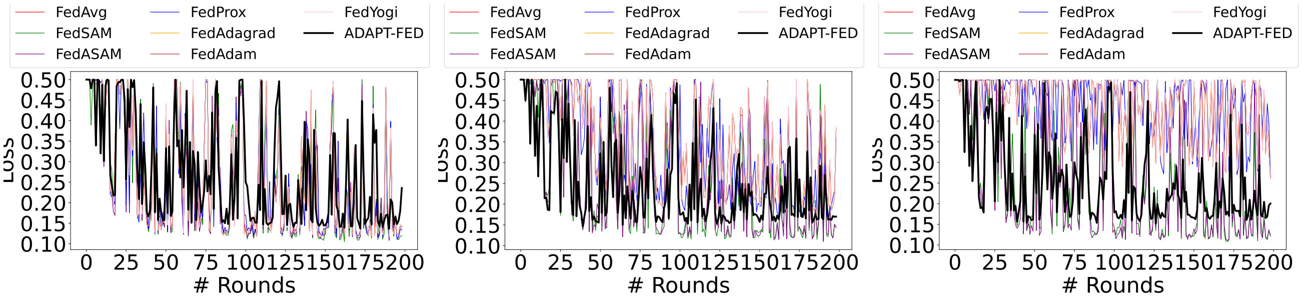

Figure 29. Convergence of the training loss of ADAPT-FED and baseline algorithms on 20 clients (UTK noniid-ness $\alpha = 0.05$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT-FED exhibits faster and more robust convergence.

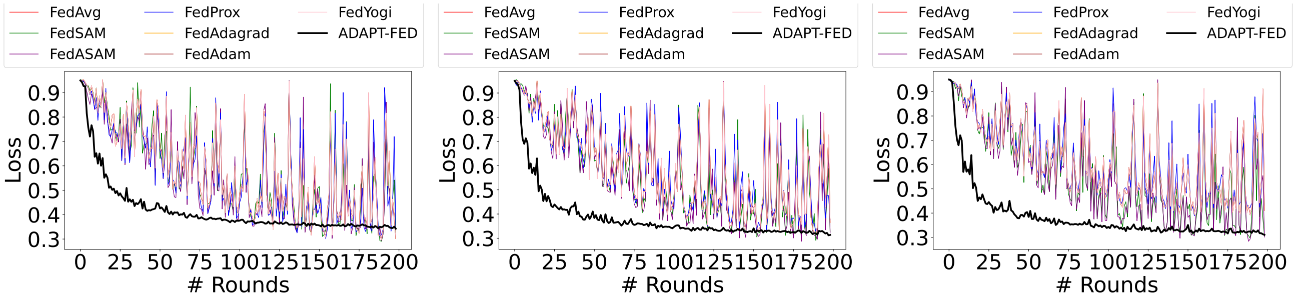

Figure 30. Convergence of the training loss of ADAPT-FED and baseline algorithms on 10 clients (CIFAR100 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT-FED exhibits faster and more robust convergence.

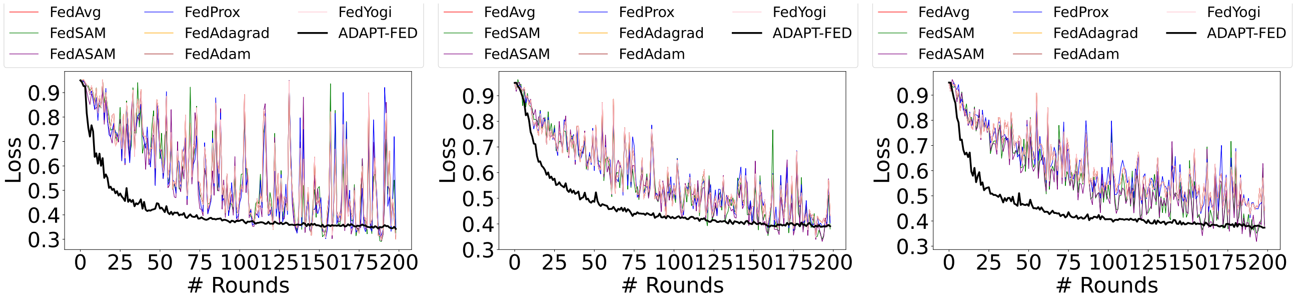

Figure 31. Convergence of the training loss of ADAPT-FED and baseline algorithms on 20 clients (CIFAR100 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT-FED exhibits faster and more robust convergence.

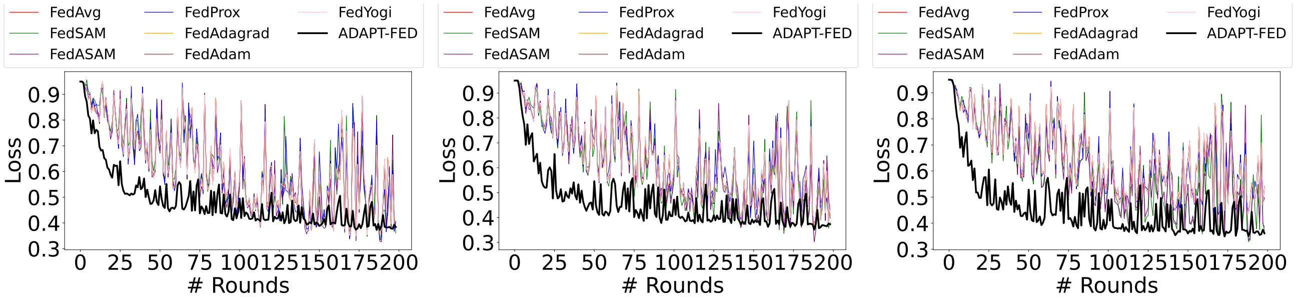

Figure 32. Convergence of the training loss of ADAPT−FED and baseline algorithms on 10 clients (CIFAR100 noniid-ness $\alpha = 0.05$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

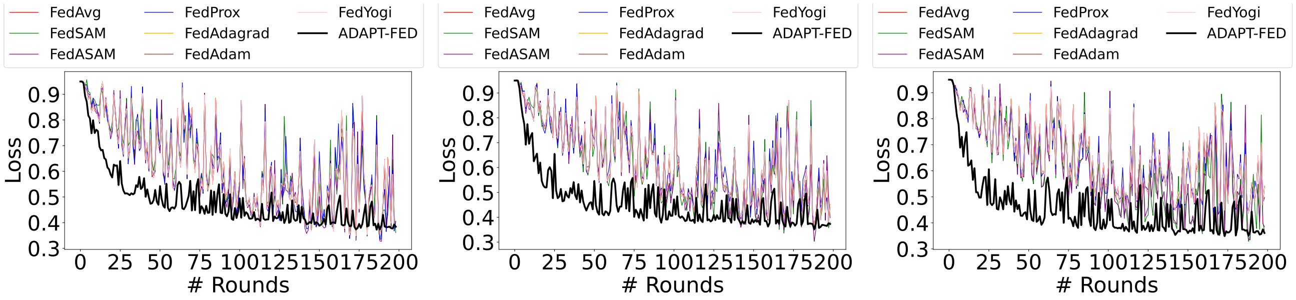

Figure 33. Convergence of the training loss of ADAPT−FED and baseline algorithms on 20 clients (CIFAR100 noniid-ness $\alpha = 0.05$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively.ADAPT−FED exhibits faster and more robust convergence.

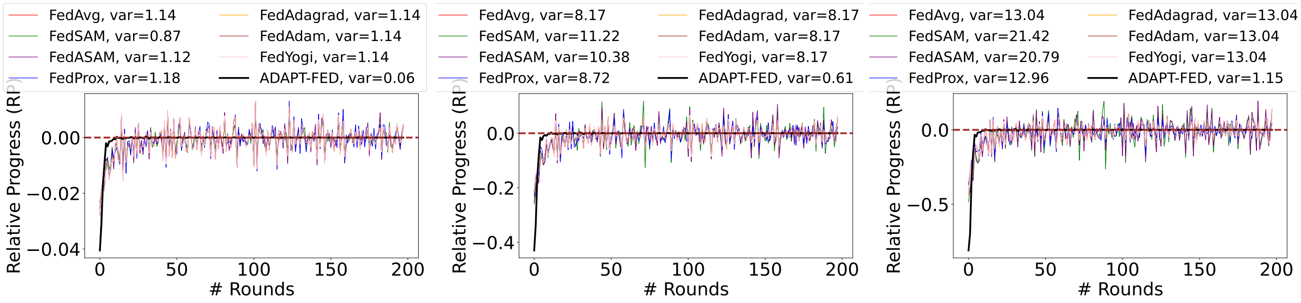

Figure 34. Stability of the training loss of ADAPT−FED and baseline algorithms on 10 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT−FED exhibits more stable convergence compared to baselines.

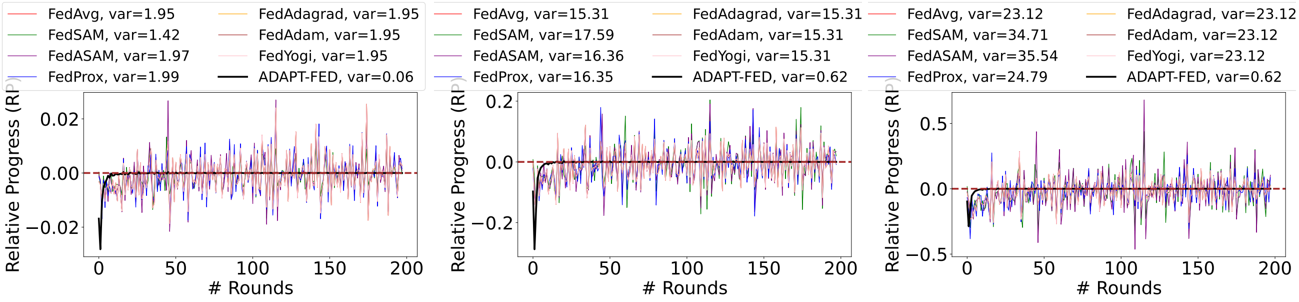

Figure 35. Stability of the training loss of ADAPT-FED and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

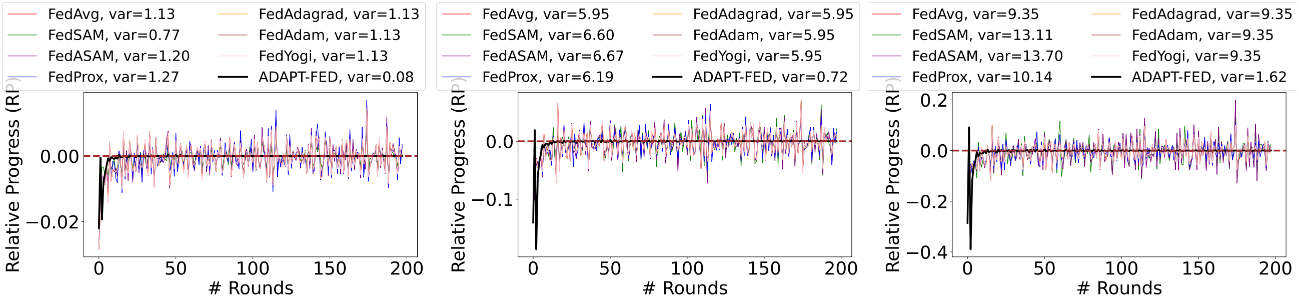

Figure 36. Stability of the training loss of ADAPT-FED and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.04$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

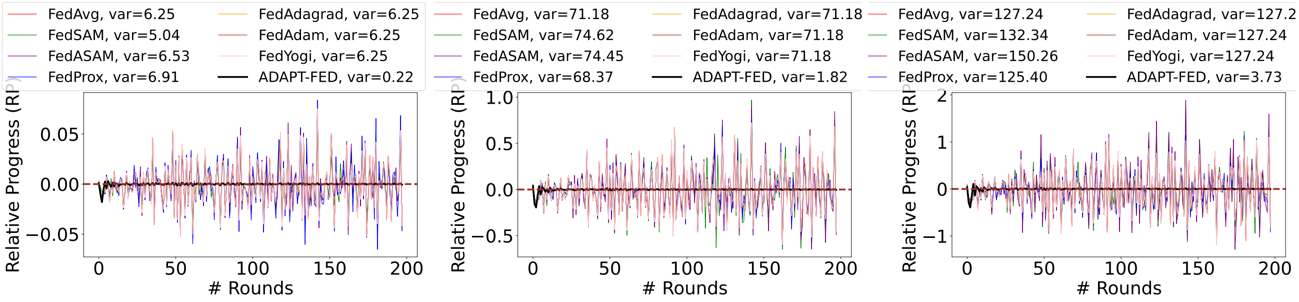

Figure 37. Stability of the training loss of ADAPT-FED and baseline algorithms on 10 clients (CIFAR10 noniid-ness $\alpha = 0.05$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

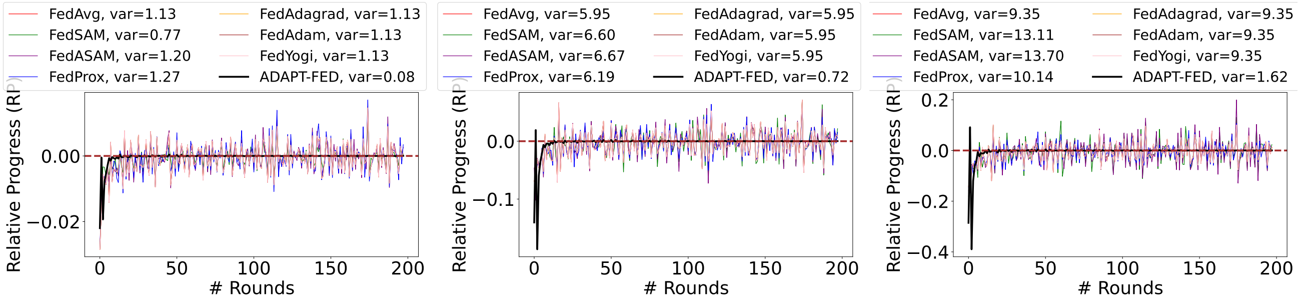

Figure 38. Stability of the training loss of ADAPT-FED and baseline algorithms on 20 clients (CIFAR10 noniid-ness $\alpha = 0.05$, $\eta_0 = 0.04$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

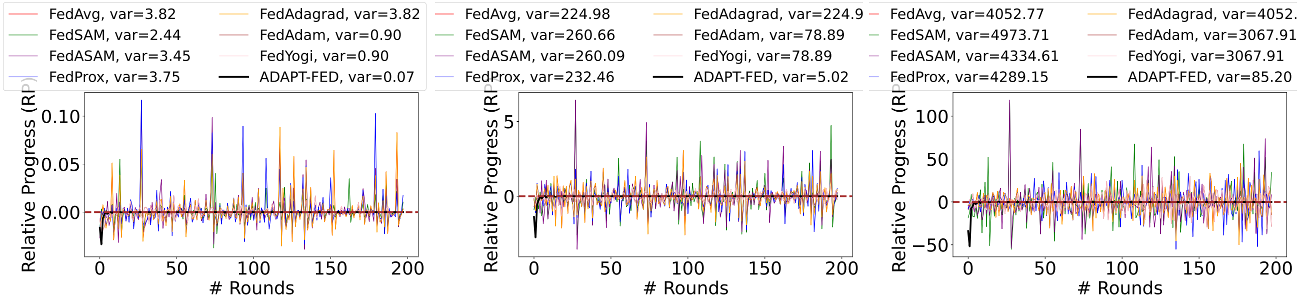

Figure 39. Stability of the training loss of ADAPT-FED and baseline algorithms on 10 clients (UTK noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

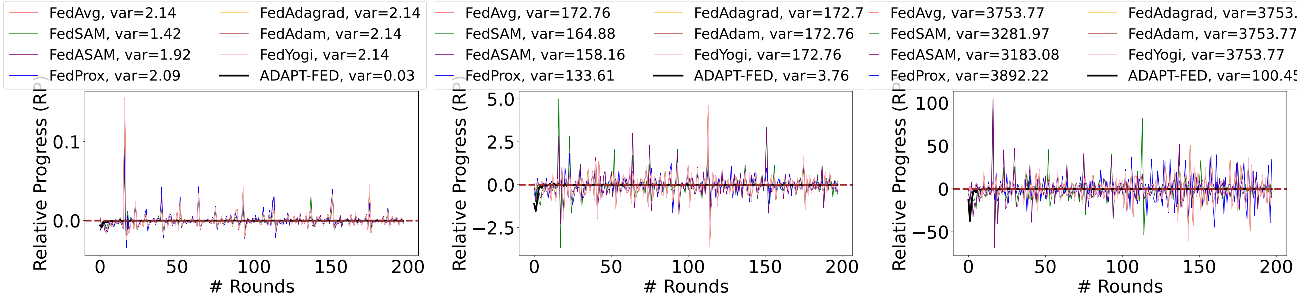

Figure 40. Stability of the training loss of ADAPT-FED and baseline algorithms on 20 clients (UTK noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT-FED exhibits more stable convergence compared to baselines.

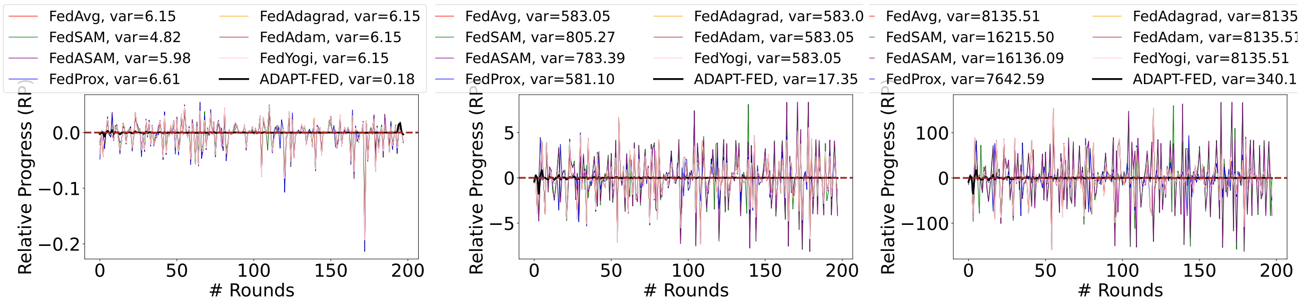

Figure 41. Stability of the training loss of ADAPT−FED and baseline algorithms on 20 clients (UTK noniid-ness $\alpha = 0.05$, $\eta_0 = 0.1$) across three DP levels: (a) $\sigma^2 = 0.0$, (b) $\sigma^2 = 0.01$, and (c) $\sigma^2 = 0.02$, respectively. ADAPT−FED exhibits more stable convergence compared to baselines.

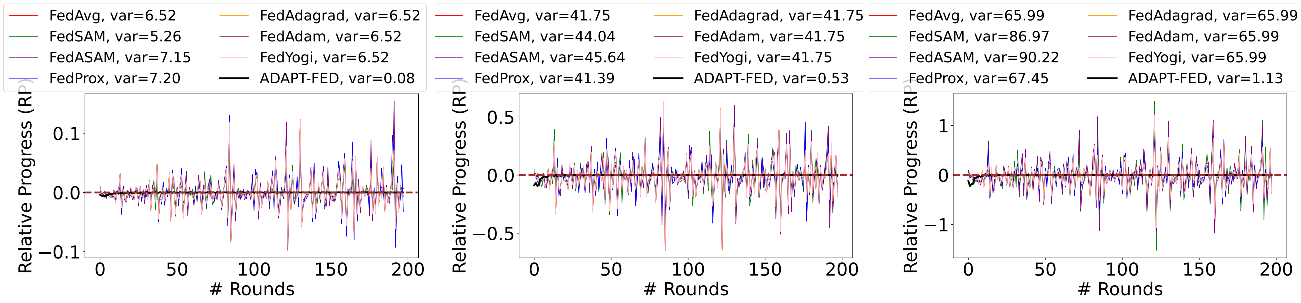

Figure 42. Stability of the training loss of ADAPT−FED and baseline algorithms on 10 clients (CIFAR100 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$). ADAPT−FED exhibits more stable convergence compared to baselines.

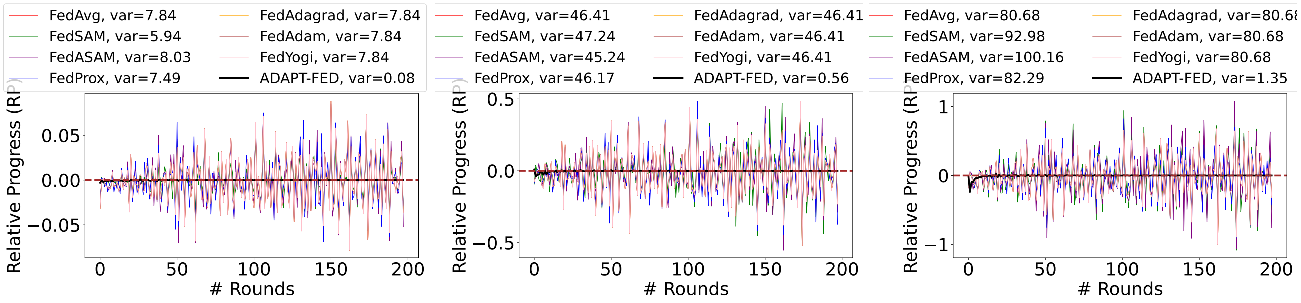

Figure 43. Stability of the training loss of ADAPT−FED and baseline algorithms on 20 clients (CIFAR100 noniid-ness $\alpha = 0.3$, $\eta_0 = 0.1$). ADAPT−FED exhibits more stable convergence compared to baselines.

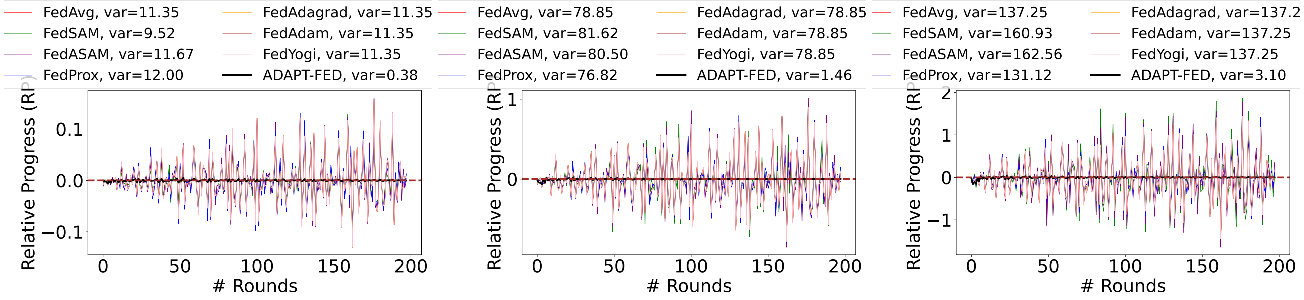

Figure 44. Stability of the training loss of ADAPT-FED and baseline algorithms on 10 clients (CIFAR100 noniid-ness $\alpha = 0.05$, $\eta_0 = 0.1$). ADAPT-FED exhibits more stable convergence compared to baselines.

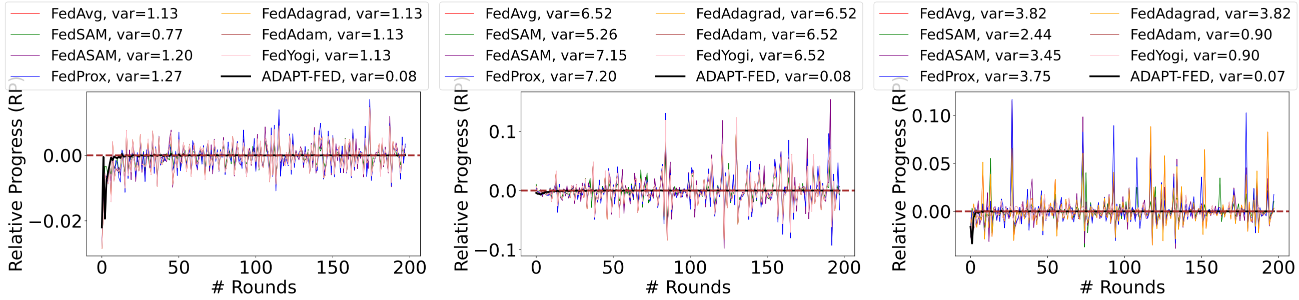

Figure 45. Stability of the training loss of ADAPT-FED and baseline algorithms on 10 clients across three datasets (CIFAR10, CIFAR100, and UTK, noniid-ness $\alpha = 0.3$, $\sigma^2 = 0.0$), $\eta_o = 0.04$.

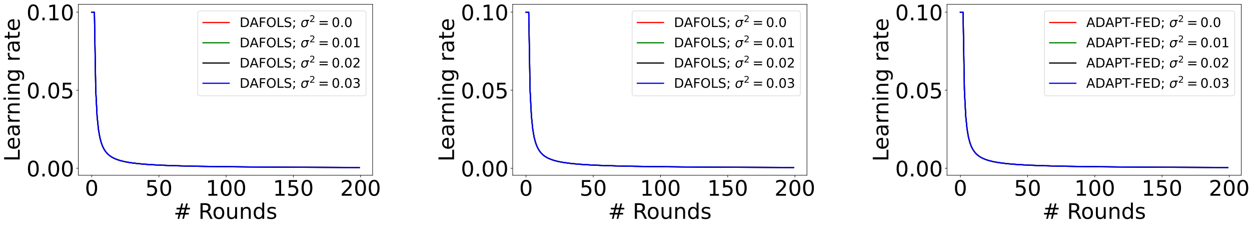

Figure 46. Behavior of η for CIFAR10, CIFAR100, and UTK