FedAdamW: A Communication-Efficient Optimizer with Convergence and Generalization Guarantees for Federated Large Models

Junkang Liu[1], Fanhua Shang[1 *], Hongying Liu[2*], Yuxuan Tian[1], Yuanyuan Liu[3*], Jin Liu[3], Kewen Zhu[1], Zhouchen Lin[4,5]

[1]School of Computer Science and Technology, Tianjin University, Tianjin, China
[2]Medical College, Tianjin University, Tianjin, China
[3]School of Artificial Intelligence, Xidian University, China
[4]State Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University
[5]Pazhou Laboratory (Huangpu), Guangzhou, Guangdong, China
fhshang@tju.edu.cn, hyliu2009@tju.edu.cn, yyliu@xidian.edu.cn

Abstract

AdamW has become one of the most effective optimizers for training large-scale models. We have also observed its effectiveness in the context of federated learning (FL). However, directly applying AdamW in federated learning settings poses significant challenges: (1) due to data heterogeneity, AdamW often yields high variance in the second-moment estimate $\boldsymbol{v}$; (2) the local overfitting of AdamW may cause client drift; and (3) reinitializing moment estimates ($\boldsymbol{v}, \boldsymbol{m}$) at each round slows down convergence. To address these challenges, we propose the first Federated AdamW algorithm, called `FedAdamW`, for training and fine-tuning various large models. `FedAdamW` aligns local updates with the global update using both a **local correction mechanism** and decoupled weight decay to mitigate local overfitting. `FedAdamW` efficiently aggregates the `mean` of the second-moment estimates to reduce their variance and reinitialize them. Theoretically, we prove that `FedAdamW` achieves a linear speedup convergence rate of $\mathcal{O}(\sqrt{(L\Delta\sigma_l^2)/(SKR\epsilon^2)} + (L\Delta)/R)$ without **heterogeneity assumption**, where S is the number of participating clients per round, K is the number of local iterations, and R is the total number of communication rounds. We also employ PAC-Bayesian generalization analysis to explain the effectiveness of decoupled weight decay in local training. Empirically, we validate the effectiveness of `FedAdamW` on language and vision Transformer models. Compared to several baselines, `FedAdamW` significantly reduces communication rounds and improves test accuracy.

Code — https://github.com/junkangLiu0/FedAdamW
Extended version — https://arxiv.org/pdf/2510.27486

Introduction

With the rapid growth of data and rising concerns over user privacy, traditional centralized training paradigms have become inadequate. **Federated Learning (FL)** (McMahan et al. 2017) offers a scalable and privacy-preserving framework that enables collaborative model training across decentralized clients without sharing raw data (Bian et al. 2025a; Wang et al. 2024; Bian et al. 2025b, 2024). As data becomes increasingly siloed, FL is a practical solution for large-scale distributed deep learning (Li et al. 2025a, 2023; An et al. 2022, 2024b,a; Liu et al. 2023, 2025d,e,c).

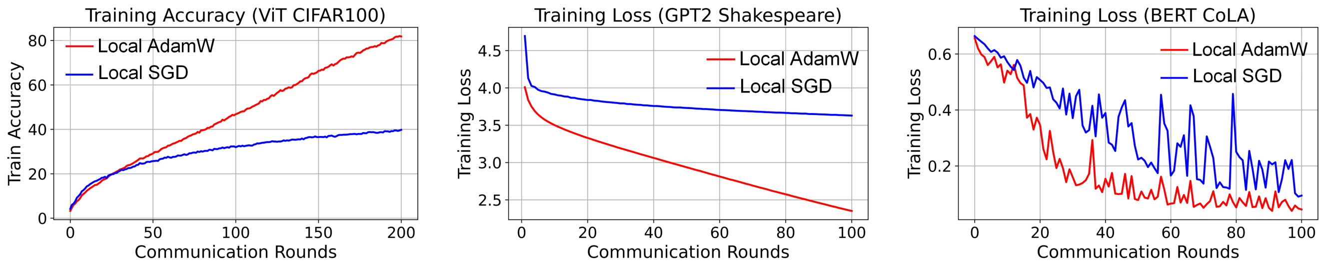
However, recent trends in model design-particularly the rise of large-scale architectures such as GPT (Radford et al. 2018), RoBERTa (Liu et al. 2019), and Vision Transformers (ViT) (Dosovitskiy et al. 2020)—pose new challenges for existing FL algorithms. Specifically, the widely-used **FedAvg** algorithm, which relies on stochastic gradient descent (**SGD**) (Bottou 2010) in local, struggles to efficiently train Transformer models. This is due to the slow convergence and poor adaptivity of SGD in Transformer models (Zhang et al. 2024c; Liu et al. 2025a), which have more complex architectures compared to CNNs. For example, components such as query, key, and value often require different learning rates to be trained effectively (Zhang et al. 2024c). In contrast, **AdamW** (Loshchilov, Hutter et al. 2017), an adaptive optimizer with decoupled weight decay, has demonstrated superior performance in centralized training of large models based on Transformer (Vaswani et al. 2017; Liu et al. 2019), offering faster convergence and improved generalization, compared to **Adam** (Kingma and Ba 2014) and SGD.

Empirically, we also observe this advantage in FL: as shown in **Figure 1**, local training with AdamW (`Local AdamW`) converges significantly faster than `Local SGD` (McMahan et al. 2017) for training various Transformer models . *However, naively applying AdamW in FL leads to the following new challenges:*

- **Challenge 1: High variance in second-moment estimate ($\boldsymbol{v}$).** Due to non-i.i.d. data across clients, gradient noise leads to high variance in second-moment estimate.
- **Challenge 2: Local overfitting and client drift.** While AdamW accelerates local training, it intensifies local overfitting. Under non-i.i.d. data, this manifests as client drift, severely hindering the global model's performance.
- **Challenge 3: Moment estimate reinitialization.** reinitializing the first- and second-moment estimates from scratch in every round hinders the convergence rate.

These challenges motivate us to develop **Federated AdamW** (`FedAdamW`), a novel optimizer tailored for fed-

*Corresponding authors.



(a) ViT on CIFAR100

(b) GPT2 on Shakespeare

(c) BERT on CoLA

Figure 1: Performance of Local SGD and Local AdamW. For training ViT-Base, GPT2, and BERT (Liu et al. 2019), we carefully tune the learning rate. For training all these Transformer models, Local SGD is still significantly worse than Local AdamW.

erated learning. `FedAdamW` addresses the above issues through two key designs: (1) a **local correction mechanism** that integrates global gradient estimates into the local update, effectively aligning local and global updates to reduce client drift; and (2) a **moment aggregation strategy** that aggregates the `mean` of second-moment estimates is theoretically grounded in the Hessian block structure, to reduce variance of $\boldsymbol{v}$ and avoid repeated initialization.

Our contributions are summarized as follows:

- **Empirical importance of AdamW and challenges in FL.** We empirically demonstrate the effectiveness of the AdamW optimizer in federated settings, particularly for training Transformer models. Our analysis reveals three key challenges when applying AdamW in FL.
- **We propose `FedAdamW`, a principled FL algorithm tailored for adaptive optimizers.** To address the above challenges, `FedAdamW` integrates global update estimate into local updates to mitigate overfitting and improve consistency. Inspired by the Hessian structure, we design a communication-efficient aggregation strategy that communicates the `mean` of second-moment across clients.
- **Theoretical guarantees with improved convergence and generalization.** `FedAdamW` achieves a linear speedup convergence rate of $\mathcal{O}(\sqrt{(L\Delta\sigma_l^2)/(SKR\epsilon^2)} + (L\Delta)/R)$. To the best of our knowledge, this is the first federated adaptive optimization algorithm without requiring **gradient heterogeneity assumption**. Furthermore, we utilize the **PAC-Bayesian theory** to provide insights into the generalization benefits of decoupled weight decay and global-local alignment.

Related Work

• **Heterogeneity Issues in Federated Learning.** Data heterogeneity across clients is a fundamental challenge in FL. A range of algorithms have been proposed to mitigate the adverse effects of non-i.i.d. data distributions. For example, FedProx (Li et al. 2020) introduces a proximal term to restrict local updates; SCAFFOLD (Karimireddy et al. 2020) applies control variates to correct client drift; and FedCM (Xu et al. 2021) leverages client momentum to stabilize updates. FedNSAM (Liu et al. 2025b) analyzed the consistency between local and global flatness, FedBCGD (Liu et al. 2024) proposed a communication-efficient accelerated block coordinate gradient method. FedSWA (Liu et al. 2025a) further improved generalization under highly heterogeneous data via stochastic weight averaging.

• **Adaptive Optimization in Centralized Settings.** Adaptive gradient methods have demonstrated superior empirical performance over SGD in centralized settings, particularly for deep neural networks. Pioneering works include Adagrad (Duchi, Hazan, and Singer 2011), Adam (Kingma and Ba 2014), AMSGrad (Reddi, Kale, and Kumar 2019), and AdamW (Loshchilov, Hutter et al. 2017). AdamW, in particular, decouples weight decay from gradient updates, offering improved generalization and training stability—attributes especially critical for Transformer models (Liu et al. 2019; Zhang et al. 2024c; Ouyang et al. 2025; Qian et al. 2024; Li et al. 2025b; Yang et al. 2025; Zhang et al. 2025, 2024b,a; Wei et al. 2025; Zhou et al. 2023, 2024)

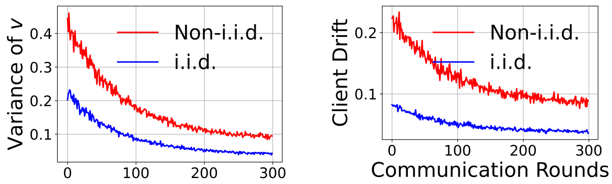
• **Adaptive Optimization in Federated Learning.** Recent efforts have explored integrating adaptive methods into FL. FedOpt (Reddi et al. 2020) incorporates server-side adaptivity using Adam and Yogi. FAFED (Wu et al. 2023) aggregates both the first- and second-moment estimates of Adam across clients to stabilize training. FedAMS (Chen, Li, and Li 2020) shows that averaging the second-moment estimate of Adam is crucial to prevent divergence. More recently, Sun et al. (2023) proposed to only aggregate the second-moment estimate to reduce communication overhead. However, these works only conducted experiments on CNN models. These studies are all based on **Adam**, which performs poorly with large weight decay.

FL Problem Setup

FL aims to optimize model parameters with local clients, i.e., minimizing the following population risk:

$$f(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\left(f_i(\boldsymbol{x}) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left[F_i\left(\boldsymbol{x};\xi_i\right)\right]\right). \quad (1)$$

The function f_i represents the loss function on client i. $\mathbb{E}_{\xi_i\sim\mathcal{D}_i}[\cdot]$ denotes the conditional expectation with respect to the sample ξ_i. ξ_i is drawn from distribution $\mathcal{D}_i$ in client i. N is the number of clients.



(a) High Variance in $\boldsymbol{v}$ (b) Local AdamW client drift

Figure 2: Training on CIFAR-100 using ViT-Tiny. (a) Data heterogeneity causes high variance in second-moment estimates across clients of Local AdamW. (b) Local AdamW suffers from more severe client drift than Local SGD under non-i.i.d. data.

Challenges of AdamW in FL

Despite the widespread use of AdamW (Loshchilov, Hutter et al. 2017; Vaswani et al. 2017) in centralized deep learning, its adaptation to federated settings remains largely unexplored. In this section, we analyze three fundamental challenges that hinder its effectiveness in FL settings.

Challenge 1: High Variance in Second-Moment Estimates ($\boldsymbol{v}$). AdamW maintains a second-moment estimate ($\boldsymbol{v}$) to scale gradients adaptively, updated as:

$$\boldsymbol{v}_i^{r,k} = \beta_2 \boldsymbol{v}_i^{r,k-1} + (1-\beta_2)\boldsymbol{g}_i^{r,k} \odot \boldsymbol{g}_i^{r,k}, \tag{2}$$

where $\boldsymbol{v}_i^{r,k}$ denotes the second-moment estimate maintained by client i at local step k of round r, $\boldsymbol{g}_i^{r,k}$ is the stochastic gradient, $\beta_2 = 0.999$ is the exponential decay rate for the second moment, and $\odot$ represents the element-wise (Hadamard) product in **Algorithm 1**. In FL, data heterogeneity leads to gradient heterogeneity. The squared stochastic gradients $\boldsymbol{g}_i^{r,k} \odot \boldsymbol{g}_i^{r,k}$ in Eq. (2) amplify the variance of $\boldsymbol{v}$ across clients in **Figure 2 (a)**. This can cause instability and inefficient aggregation, especially when using non-i.i.d. data (Chen, Li, and Li 2020).

Challenge 2: Local Overfitting and Client Drift. While AdamW accelerates convergence through its adaptivity, it may exacerbate local overfitting. In FL, where each client minimizes its own local objective $f_i(\cdot)$, creating a natural gap between the local and global optima. Adaptive optimizers such as AdamW, with stronger update magnitudes, can drive clients further toward their local optima—diverging from the global direction. This leads to client drift as illustrated in **Figure 2 (b)**, which manifests as inconsistencies in local models that degrade the global performance.

Challenge 3: Reinitialization Overhead. In FL, AdamW optimizer states are reinitialized from zero each round:

$$\boldsymbol{m}_i^{r,0} \leftarrow \mathbf{0}, \quad \boldsymbol{v}_i^{r,0} \leftarrow \mathbf{0}. \tag{3}$$

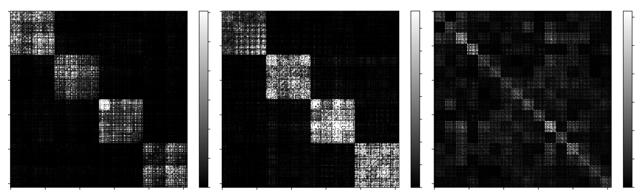
Reinitializing moment estimates across rounds erases temporal memory, hindering the accumulation of adaptive statistics and slowing convergence, particularly in deep or large-scale models.

Our Algorithm: FedAdamW

Based on theoretical motivation, we propose an efficient improvement to AdamW called Federated AdamW (`FedAdamW`). To address **Challenge 1**, it was experimentally discovered that aggregating AdamW second-moment estimate can stabilize the training process (see **Table** 7 below). However, aggregating second-moment estimate leads to a double communication.

Algorithm 1: Local AdamW Algorithm

Initial model $\boldsymbol{x}^0$, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, time step $t \leftarrow 0$, the number of all clients N, each round selected clients S, weight decay λ.
for $r = 1, \ldots, R$ **do**
  for each selected client $i \in \{1, \ldots, S\}$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r$, $\boldsymbol{m}_i^{r,0} \leftarrow \mathbf{0}$, $\boldsymbol{v}_i^{r,0} \leftarrow \mathbf{0}$;
    for $k = 1, \ldots, K$ **do**
      $\boldsymbol{g}_i^{r,\tau} \leftarrow \nabla f_i(\boldsymbol{x}_i^{r,k}; \xi_i)$;
      $\boldsymbol{m}_i^{r,k} = \beta_1 \boldsymbol{m}_i^{r,k-1} + (1-\beta_1)\, \boldsymbol{g}_i^{r,k}$;
      $\boldsymbol{v}_i^{r,k} = \beta_2 \boldsymbol{v}_i^{r,k-1} + (1-\beta_2)\, \boldsymbol{g}_i^{r,k} \odot \boldsymbol{g}_i^{r,k}$;
      $\hat{\boldsymbol{m}}_i^{r,k} = \boldsymbol{m}_i^{r,k} / \left(1-\beta_1^k\right)$;
      $\hat{\boldsymbol{v}}_i^{r,k} = \boldsymbol{v}_i^{r,k} / \left(1-\beta_2^k\right)$;
      $\boldsymbol{x}_i^{r,k+1} = \boldsymbol{x}_i^{r,k} - \eta(\hat{\boldsymbol{m}}_i^{r,k} / (\sqrt{\hat{\boldsymbol{v}}_i^{r,k}} + \epsilon) - \lambda \boldsymbol{x}_i^{r,k})$;
    end for
    Communicate $(\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0})$ to Server;
  end for
  $\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \frac{1}{S}\sum_{i=1}^{S}(\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0})$;
  Communicate $(\boldsymbol{x}^{r+1})$ to Clients.
end for



(a) `query` (4 heads) (b) `key` (4 heads) (c) `value` (4 heads)

Figure 3: (a–c):Block-wise Hessian structure of Transformer parameters under FL. Visualizing the Hessian submatrices of query, key, and value heads. The near block-diagonal structure supports block-wise second-moment aggregation in FedAdamW.

(Q1) How to efficiently aggregate v?

We observe that the Hessian matrix in neural networks exhibits an approximate block-diagonal structure with several dense sub-blocks (Collobert 2004; Zhang et al. 2024d) as shown in **Figure 3**. In such a structure, a single learning rate can effectively capture the curvature within each block. Leveraging this, we propose a communication-efficient strategy that partitions the second-moment estimate $\boldsymbol{v}$ into B blocks and transmits only the mean of each in **Figure 4**:

$$\bar{\boldsymbol{v}}_b = \mathtt{mean}(\boldsymbol{v}_b), \quad b = 1, \ldots, B. \tag{4}$$

Block-wise Partitioning Strategy (ViT Example). We group the parameters into semantically aligned classes that exhibit similar curvature patterns as shown in Figure 3:

Algorithm 2: `FedAdamW` Algorithm

Initial model $\boldsymbol{x}^0$, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, time step $t \leftarrow 0$, the number of all clients N, each round selected clients S, weight decay λ.
for $r = 1, \ldots, R$ **do**
  for each selected client $i \in \{1, \ldots, S\}$ in parallel **do**
    $\boldsymbol{x}_i^{r,0} \leftarrow \boldsymbol{x}^r, \boldsymbol{m}_i^{r,0} \leftarrow \boldsymbol{0}, \boldsymbol{v}_i^{r,0} \leftarrow \bar{\boldsymbol{v}}^r$;
    for $k = 1, \ldots, K$ **do**
      $t \leftarrow t + 1$;
      $\boldsymbol{g}_i^{r,k} \leftarrow \nabla f_i(\boldsymbol{x}_i^{r,k}; \xi_i)$;
      $\boldsymbol{m}_i^{r,k} = \beta_1 \boldsymbol{m}_i^{r,k-1} + (1 - \beta_1)\, \boldsymbol{g}_i^{r,k}$;
      $\boldsymbol{v}_i^{r,k} = \beta_2 \boldsymbol{v}_i^{r,k-1} + (1 - \beta_2)\, \boldsymbol{g}_i^{r,k} \odot \boldsymbol{g}_i^{r,k}$;
      Bias correction
      $\hat{\boldsymbol{m}}_i^{r,k} = \boldsymbol{m}_i^{r,k} / \left(1 - \beta_1^k\right)$;
      $\hat{\boldsymbol{v}}_i^{r,k} = \boldsymbol{v}_i^{r,k} / \left(1 - \beta_2^t\right)$;
      $\vartheta_i^{r,k} = 1/(\sqrt{\hat{\boldsymbol{v}}_i^{r,k}} + \epsilon)$;
      Update model parameters
      $\boldsymbol{x}_i^{r,k+1} = \boldsymbol{x}_i^{r,k} - \eta(\hat{\boldsymbol{m}}_i^{r,k} \odot \vartheta_i^{r,k} + \alpha \boldsymbol{\Delta}_G^r - \lambda \boldsymbol{x}_i^{r,k})$;
    end for
    Communicate $(\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0}, \bar{\boldsymbol{v}}_i = \mathtt{mean}\left(\boldsymbol{v}_i^{r,K}\right))$ to Server;
  end for
  $\boldsymbol{\Delta}_G^r = \frac{-1}{SK\eta} \sum_{i=1}^{S} (\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0})$;
  $\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \frac{1}{S} \sum_{i=1}^{S} (\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0})$;
  $\bar{\boldsymbol{v}}^{r+1} = \frac{1}{S} \sum_{i=1}^{S} \bar{\boldsymbol{v}}_i$;
  Communicate $(\boldsymbol{x}^{r+1}, \bar{\boldsymbol{v}}^{r+1}, \boldsymbol{\Delta}_G^r)$ to Clients.
end for

- **Class 1: `query` and `key`.** Query and Key parameters. Each block corresponds to one attention head.
- **Class 2: `attn.proj` and MLPs.** Blocks align with output neurons in projection and feedforward layers.
- **Class 3: `value`.** Structure is less regular but still shows diagonal blocks; curvature magnitude is notably higher (up to $10^6\times$), possibly due to its position after softmax.
- **Class 4: `Embedding` and `output` layers.** Sub-blocks align with input tokens, forming near-diagonal Hessians.

CNNs (e.g., ResNet): Blocked by convolutional layers or residual blocks. This reduces the communication cost from billions of scalars to B values while preserving adaptive behavior. Empirically, we find this approach also improves generalization in local optimization as shown in **Table** 7 below. See **Appendix D** for block partitioning details.

(Q2) How to overcome overfitting in Local AdamW?

To address local overfitting (i.e., **Challenge 2**), we adopt a stronger weight decay. Unlike Adam, AdamW employs decoupled weight decay, which improves generalization, particularly in federated settings (see **Table 6** below). To further mitigate client drift under non-i.i.d. data, we incorporate a global update estimate into the local update rule:

$$\boldsymbol{x}_i^{r,k+1} = \boldsymbol{x}_i^{r,k} - \eta\left(\hat{\boldsymbol{m}}_i^{r,k} \odot \vartheta_i^{r,k} - \lambda \boldsymbol{x}_i^{r,k} + \alpha \boldsymbol{\Delta}_G^r\right), \quad (5)$$

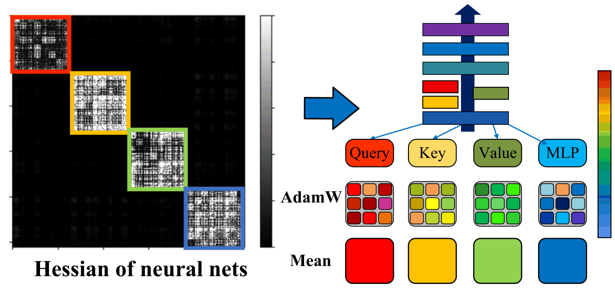


Figure 4: Illustration of FedAdamW's block-wise aggregation strategy Clients estimate local second-moment statistics and send block-wise means to the server, reducing communication cost.

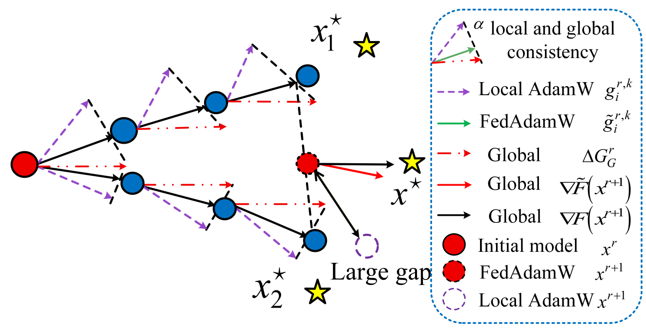


Figure 5: An illustration of local update in `FedAdamW`, which corrects client drift caused through global update guidance.

where $\boldsymbol{\Delta}_G^r = \frac{-1}{SK\eta} \sum_{i=1}^{S} (\boldsymbol{x}_i^{r,K} - \boldsymbol{x}_i^{r,0})$ is the estimated global update. As shown in **Figure 5**, this alignment reduces the divergence of local models and improves global consistency.

(Q3) How to initialize second-moment estimates?

We find that initializing the second-moment estimate $\boldsymbol{v}$ with its aggregated `mean` $\boldsymbol{v}$ significantly accelerates convergence (see **Table 7** below). In contrast, we reinitialize the first-moment estimate $\boldsymbol{m}$ to zero at each round. This is because $\boldsymbol{m}$ adapts quickly to recent gradients and does not require long-term accumulation to remain effective.

Theoretical Analysis

Convergence Analysis

In this part, we give the convergence theoretical analysis of our proposed `FedAdamW` algorithm. Firstly we state some standard assumptions for the non-convex function f.

Assumption 1 (Smoothness)**.** *(Smoothness) The non-convex f_i is a L-smooth function for all $i \in [m]$, i.e., $\|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$.*

Assumption 2 (Bounded Stochastic Gradient)**.** *$\boldsymbol{g}_i^r = \nabla f_i(\boldsymbol{x}_i^r, \xi_i^r)$ computed by using a sampled mini-batch data ξ_i^r in the local client i is an unbiased estimator of ∇f_i with bounded variance, i.e., $\mathbb{E}_{\xi_i^r}[\boldsymbol{g}_i^r] = \nabla f_i(\boldsymbol{x}_i^r)$ and $\mathbb{E}_{\xi_i^r}\|g_i^r - \nabla f_i(\boldsymbol{x}_i^r)\|^2 \leq \sigma_l^2$, for all $\boldsymbol{x}_i^r \in \mathbb{R}^d$.*

Assumption 3 (Bounded Stochastic Gradient II). *Each element of stochastic gradient $\boldsymbol{g}_i^r$ is bounded, i.e., $\|\boldsymbol{g}_i^r\|_\infty = \|f_i(\boldsymbol{x}_i^r, \xi_i^r)\|_\infty \leq G_g$, for all $\boldsymbol{x}_i^r \in \mathbb{R}^d$ and any sampled mini-batch data ξ_i^r.*

Assumption 4 (Bounded Heterogeneity). *The dissimilarity between local clients is bounded on the gradients, i.e., $\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \sigma_g^2$, for all $\boldsymbol{x} \in \mathbb{R}^d$.*

These assumptions are standard in federated adaptive optimization literature (Fan et al. 2024; Sun et al. 2023).

Theorem 1 (Convergence for non-convex functions). *Under Assumptions 1, 2, and 3, if we take $g^0 = 0, \beta_1 = 0, \lambda = 0$ then `FedAdamW` converges as follows*

$$\frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^r)\|^2\right] \lesssim \mathcal{O}\left(\sqrt{\frac{L\Delta\sigma_l^2}{SKR\epsilon^2}} + \frac{L\Delta}{R}\right). \quad (6)$$

Here $G_0 := \frac{1}{N}\sum_{i=1}^N \left\|\nabla f_i\left(\boldsymbol{x}^0\right)\right\|^2, \Delta = f\left(\boldsymbol{x}^0\right) - f^\star$, S is the number of participating clients per round, K is the number of local iterations, and R is the total number of communication rounds.

The proof is provided in **Appendix A**. The convergence rate of `FedAdamW` is faster than that of Local AdamW and FedLADA's $\mathcal{O}\left(\sqrt{\frac{L\Delta(\sigma_l^2+\sigma_g^2)}{SKR\epsilon^2}} + \frac{L\Delta}{R}\right)$, and we do not need **Assumption 4**. This is due to the suppression of local drift by the global update estimation $\boldsymbol{\Delta}_G^r$. We have verified this in **Table 5** below.

Generalization Analysis

Theorem 2. *Assume the prior hypothesis $\boldsymbol{x}_0$ satisfies $\mathcal{P}_{pre} \sim \mathcal{N}(\mathbf{0}, \rho\boldsymbol{I})$. Then the expected risk for the posterior hypothesis $\boldsymbol{x} \sim \mathcal{P}$ of `FedAdamW` learned on training dataset $\mathcal{D}_{tr} \sim \mathcal{D}$ with n samples holds*

$$\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{D},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] - \mathbb{E}_{\boldsymbol{\xi}\in\mathcal{D}_{tr},\boldsymbol{x}\sim\mathcal{P}}[f(\boldsymbol{x},\boldsymbol{\xi})] \leq$$

$$\frac{\sqrt{8}}{\sqrt{n}}\left(\sum_{i=1}^d \log\frac{2\rho b\left(\sigma_i^{\frac{1}{2}}+\lambda\right)}{\eta} + \frac{\eta}{2\rho b}\sum_{i=1}^d \frac{1}{\sigma_i^{\frac{1}{2}}+\lambda} + c_0\right)^{\frac{1}{2}}$$

, with at least probability $1-\tau$, where $\tau \in (0,1)$ and $c_0 = \frac{1}{2\rho}\|\boldsymbol{x}_*\|^2 - \frac{d}{2} + 2\ln\left(\frac{2n}{\tau}\right)$. Here, σ_i represents the local curvature (e.g., Hessian eigenvalue). b is the batch size, η is the learning rate, λ is a weight decay parameter, n is the training set size, and d is the parameter dimension.

The proof is provided in **Appendix B**. **Theorem 2** shows that the generalization error of `FedAdamW` can be upper bounded by $\mathcal{O}(1/\sqrt{n})$, where n is the number of tatal data, consistent with classical results from PAC theory, stability, and uniform convergence (Shalev-Shwartz and Ben-David 2014). We further analyze the impact of the decoupled weight decay parameter λ on this bound. As λ increases, the first term $\sum_{i=1}^d \log 2\rho b(\sigma_i^{1/2}+\lambda)\eta^{-1}$ increases, while the second term $\frac{\eta}{2\rho b}\sum_{i=1}^d(\sigma_i^{1/2}+\lambda)^{-1}$ decreases. Although choosing the optimal λ is challenging in practice, this trade-off suggests that tuning λ appropriately can lead to a smaller generalization error, as shown in Table **Table 6** below. This explains why `FedAdamW` often outperforms Local Adam (which corresponds to $\lambda = 0$).

Experiments

Datasets. We evaluate `FedAdamW` on both vision and language tasks. (*i*) For image classification, we use CIFAR-100 (Krizhevsky, Hinton et al. 2009), and Tiny ImageNet (Le and Yang 2015). (*ii*) For NLP tasks, we adopt benchmark datasets from the GLUE benchmark, including SST-2, QQP. To simulate data heterogeneity across clients, we follow the Dirichlet partitioning scheme (Hsu, Qi, and Brown 2019), where a Dir-0.6 corresponds to a low heterogeneity and Dir-0.1 implies high heterogeneity.

Model Architectures. We explore a variety of model types: (*i*) ResNet-18 (He et al. 2016) as a representative convolutional neural network (CNN), (*ii*) Swin Transformer (Liu et al. 2021) and ViT-Tiny (Dosovitskiy et al. 2020) for Vision Transformers, and (*iii*) RoBERTa-Base (Liu et al. 2019) for large-scale language model.

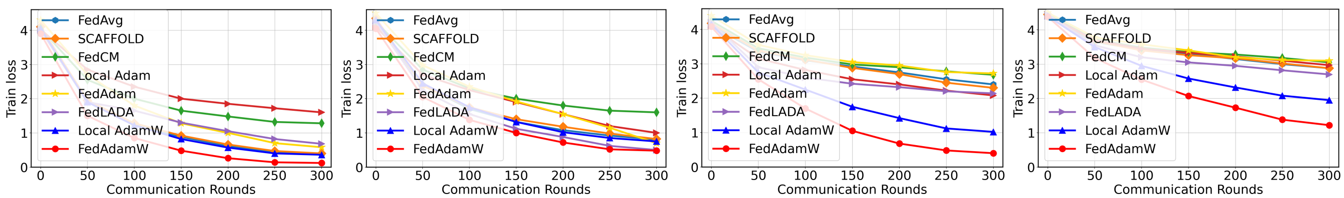
Baselines. We compare our method against state-of-the-art FL algorithms: `FedAvg` (McMahan et al. 2017), `SCAFFOLD` (Karimireddy et al. 2020), `FedCM` (Xu et al. 2021), `FedAdam` (Reddi et al. 2020), `FedLADA` (Sun et al. 2023), `Local Adam` and `Local AdamW`.

Hyperparameter Settings. For `FedAvg`, `SCAFFOLD`, `FedCM`, `FedAdam`, the lr is selected from $\{10^{-2},\ 3\times10^{-2},\ 5\times10^{-2},\ 10^{-1},\ 3\times10^{-1}\}$, with a weight decay of 0.001. For `FedAdamW`, `FedLADA`, `Local Adam` and `Local AdamW`, the lr is selected from $\{10^{-4},\ 3\times10^{-4},\ 5\times10^{-4},\ 8\times10^{-4},\ 10^{-3}\}$, with weight decay 0.01 or 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We apply cosine learning rate decay, and set `FedAdamW` to $\boldsymbol{\alpha} = 0.5$, weight decay $\boldsymbol{\lambda} = 0.01$. Additional hyperparameter configurations are detailed in **Appendix C**. We release all code, configuration files to ensure full reproducibility. All results are averaged over 5 runs with std reported.

Questions. Our experiments are designed to answer the following: **Q1.** *Does `Local AdamW` outperform `Local SGD` when training Transformer models?* **Q2.** *Can `FedAdamW` effectively address the three challenges identified for AdamW in FL?* **Q3.** *Is `FedAdamW` generally effective across both CNNs and Transformers?* **Q4.** *Are individual components of `FedAdamW`—such as global update correction, decoupled weight decay, and block-wise $\boldsymbol{v}$ averaging—empirically beneficial?* **Q5.** *Do our theoretical findings (Theorems 1 and 2) align with empirical results?*

Results on Convolutional Neural Networks

Training on CIFAR-100 with ResNet-18. **Table 1** and **Figure 6** present the test accuracy and training loss on CIFAR-100 using ResNet-18. `FedAdamW` achieves the best performance under both Dir-0.6 and Dir-0.1 settings, reaching a top accuracy of **66.12%** and **63.01%**, respectively. It also attains the lowest training loss (**0.122** and **0.480**), demonstrating faster and more stable convergence. Compared to other adaptive baselines such as `FedAdam`, `FedAdamW` shows superior generalization under data heterogeneity, confirming its effectiveness in CNNs **(Q3)**.



(a) ResNet18, Dir-0.6 (b) ResNet18, Dir-0.1 (c) ViT-Tiny, Dir-0.6 (d) ViT-Tiny, Dir-0.1

Figure 6: Training loss curves on CIFAR-100 using ResNet-18 and ViT-Tiny under Dir-0.1, Dir-0.6.

Method	**ResNet-18 (Dir-0.6)**		**ResNet-18 (Dir-0.1)**		**ViT-Tiny (Dir-0.6)**		**ViT-Tiny (Dir-0.1)**		**Comm**
	Test Acc	Train Loss	Test Acc	Train Loss	Test Acc	Train Loss	Test Acc	Train Loss	
FedAvg	$64.08_{\pm 0.18}$	0.376	$60.25_{\pm 0.20}$	0.767	$32.36_{\pm 0.08}$	2.350	$27.14_{\pm 0.12}$	2.867	1×
SCAFFOLD	$65.01_{\pm 0.15}$	0.365	$59.37_{\pm 0.16}$	0.814	$32.17_{\pm 0.12}$	2.295	$27.31_{\pm 0.11}$	2.855	2×
FedCM	$48.69_{\pm 0.10}$	1.305	$44.43_{\pm 0.08}$	1.645	$26.33_{\pm 0.06}$	2.681	$23.18_{\pm 0.15}$	3.038	1×
Local Adam	$60.98_{\pm 0.28}$	1.598	$58.88_{\pm 0.20}$	0.975	$38.69_{\pm 0.16}$	2.082	$29.88_{\pm 0.08}$	2.961	1×
FedAdam	$63.77_{\pm 0.13}$	0.562	$61.62_{\pm 0.16}$	0.707	$28.77_{\pm 0.12}$	2.709	$23.49_{\pm 0.15}$	3.084	1×
FedLADA	$65.07_{\pm 0.18}$	0.671	$59.93_{\pm 0.21}$	0.556	$37.31_{\pm 0.16}$	2.127	$35.33_{\pm 0.16}$	2.678	2×
Local AdamW	$62.84_{\pm 0.08}$	0.363	$58.97_{\pm 0.10}$	0.794	$40.47_{\pm 0.09}$	1.026	$36.86_{\pm 0.11}$	1.954	1×
`FedAdamW`	**66.12**$_{\pm 0.10}$	**0.122**	**63.01**$_{\pm 0.12}$	**0.480**	**42.56**$_{\pm 0.10}$	**0.401**	**39.86**$_{\pm 0.16}$	**1.251**	1×

Table 1: Test accuracy, training loss, and communication cost of each method on CIFAR-100 using **ResNet-18** and **ViT-Tiny** over 300 communication rounds under Dir-0.6 and Dir-0.1 settings (100 clients, 10% participation, batch size 50, $K = 50$).

Method	**CIFAR-100**		**Tiny ImageNet**	
	Test Acc	Train Loss	Test Acc	Train Loss
FedAvg	$80.02_{\pm 0.28}$	0.588	$80.38_{\pm 0.22}$	0.826
SCAFFOLD	$81.30_{\pm 0.18}$	0.514	$82.41_{\pm 0.18}$	0.650
FedCM	$82.38_{\pm 0.19}$	0.565	$83.18_{\pm 0.19}$	0.522
Local Adam	$79.75_{\pm 0.26}$	0.534	$73.63_{\pm 0.28}$	1.045
FedAdam	$77.48_{\pm 0.19}$	0.651	$78.20_{\pm 0.22}$	0.834
FedLADA	$74.64_{\pm 0.18}$	0.598	$70.95_{\pm 0.19}$	0.944
Local AdamW	$83.35_{\pm 0.10}$	0.381	$80.26_{\pm 0.12}$	0.686
`FedAdamW`	**85.85**$_{\pm 0.08}$	**0.285**	**85.23**$_{\pm 0.10}$	**0.446**

Table 2: Comparison of test accuracy and training loss for **Swin Transformer** under Dir-0.1 with 100 communication rounds(100 clients, 5% participation, batch size 16, $K = 50$).

Results on Transformer Models

Training on CIFAR-100 with ViT-Tiny. Table 1 and **Figure 6** show `FedAdamW` achieves the best performance across both heterogeneity levels, with test accuracies of **42.56%** (Dir-0.6) and **38.25%** (Dir-0.1), and the lowest training loss (**0.401** and **1.251**), confirming its efficient convergence **(Q5)**. Compared to `Local AdamW`, it provides consistent improvements in both accuracy and stability **(Q1,Q2, Q3)**. Moreover, other adaptive baselines such as `FedAdam` and `FedLADA` perform significantly worse under high heterogeneity, highlighting the effectiveness of global update correction and decoupled weight decay **(Q4)**. These results validate that `FedAdamW` is particularly effective for federated vision Transformers under non-i.i.d. conditions.

The small dataset CIFAR100 is difficult to support the performance of ViT, resulting in lower accuracy. Therefore, we continued to test on the pretrained model.

Fine-tuning Results on Swin Transformer. Table 2 reports results on Swin Transformer under Dir-0.1. `FedAdamW` achieves the highest test accuracy on both CIFAR-100 (**85.85%**) and Tiny ImageNet (**85.23%**), while also attaining the lowest training loss, reflecting faster convergence and improved generalization. Compared to other adaptive baselines such as `FedAdam` and `FedLADA`, `FedAdamW` consistently outperforms across both datasets, demonstrating its effectiveness in fine-tuning large Transformer models under non-i.i.d. conditions.

Fine-tuning Results on LLMs. Table 3 summarizes results on the GLUE benchmark using RoBERTa-Base with LoRA, 20 clients, 20% participation, batch size 32, $K = 50$, rank=16. `FedAdamW` achieves the highest average accuracy of **81.79%**, outperforming strong baselines such as `FedAvg` (77.68%) and `Local AdamW` (78.91%). It is particularly strong on challenging tasks like **RTE** and **QQP**, exceeding the next best methods by **+1.50%** and **+1.74%**, respectively.

Ablation Study

Impact of A1, A2, A3. Table 4 summarizes the effect of removing key components in `FedAdamW`. We draw the following observations: • Removing second-moment aggregation (**A1**) significantly degrades performance, indicating that `mean`$(\boldsymbol{v})$ aggregation stabilizes adaptive updates across clients. • Without global gradient alignment (**A2**), the local models drift apart, resulting in higher train loss and lower generalization. • The use of standard (non-

Method (Dir-0.8)	CoLA	RTE	SST-2	QQP	MRPC	QNLI	MNLI	Avg Acc.
FedAvg	$56.12_{\pm 0.18}$	$48.72_{\pm 0.25}$	$93.66_{\pm 0.10}$	$85.87_{\pm 0.14}$	$86.00_{\pm 0.12}$	$90.21_{\pm 0.09}$	$83.22_{\pm 0.17}$	$77.68_{\pm 0.17}$
SCAFFOLD	$57.79_{\pm 0.21}$	$51.62_{\pm 0.28}$	$93.15_{\pm 0.11}$	$84.25_{\pm 0.15}$	$86.11_{\pm 0.13}$	$90.32_{\pm 0.10}$	$83.49_{\pm 0.18}$	$77.82_{\pm 0.17}$
FedCM	$56.29_{\pm 0.16}$	$64.98_{\pm 0.22}$	$93.25_{\pm 0.12}$	$83.19_{\pm 0.17}$	$85.56_{\pm 0.13}$	$88.13_{\pm 0.15}$	$78.90_{\pm 0.19}$	$78.33_{\pm 0.18}$
Local Adam	$56.08_{\pm 0.19}$	$62.81_{\pm 0.23}$	$93.80_{\pm 0.09}$	$85.07_{\pm 0.13}$	$84.55_{\pm 0.14}$	$88.57_{\pm 0.12}$	$82.62_{\pm 0.16}$	$79.07_{\pm 0.15}$
FedAdam	$55.26_{\pm 0.20}$	$58.12_{\pm 0.26}$	$93.26_{\pm 0.10}$	$85.12_{\pm 0.13}$	$86.11_{\pm 0.11}$	$89.21_{\pm 0.09}$	$83.16_{\pm 0.17}$	$78.32_{\pm 0.16}$
FedLADA	$50.00_{\pm 0.24}$	$57.40_{\pm 0.25}$	$93.57_{\pm 0.11}$	$85.88_{\pm 0.14}$	$82.59_{\pm 0.15}$	$89.76_{\pm 0.10}$	$82.99_{\pm 0.16}$	$77.17_{\pm 0.17}$
Local AdamW	$56.45_{\pm 0.17}$	$54.15_{\pm 0.27}$	$93.57_{\pm 0.09}$	$86.93_{\pm 0.12}$	$86.27_{\pm 0.11}$	$90.73_{\pm 0.08}$	$84.26_{\pm 0.14}$	$78.91_{\pm 0.15}$
`FedAdamW (ours)`	$\mathbf{58.21}_{\pm 0.15}$	$\mathbf{66.48}_{\pm 0.20}$	$\mathbf{94.03}_{\pm 0.08}$	$\mathbf{87.62}_{\pm 0.11}$	$\mathbf{86.76}_{\pm 0.10}$	$\mathbf{90.88}_{\pm 0.07}$	$\mathbf{84.55}_{\pm 0.13}$	$\mathbf{81.79}_{\pm 0.14}$

Table 3: Test accuracy (%) using RoBERTa-Base with LoRA across seven GLUE tasks over 100 communication rounds.

Variant	Test Acc (%)	Train Loss
A1: w/o $\bar{\boldsymbol{v}}$ (no moment agg.)	$37.51_{\pm 0.12}$	1.504
A2: w/o $\boldsymbol{\Delta}_G$ (no global align.)	$37.42_{\pm 0.14}$	1.621
A3: w/o decoupled weight decay	$38.25_{\pm 0.16}$	1.356
A4: `FedAdamW` (Full)	$\mathbf{39.86}_{\pm 0.16}$	**1.251**

Table 4: Ablation study of on CIFAR-100 using ViT-Tiny (Dir-0.1, 300 rounds).

α (Dir-0.1)	0.00	0.25	**0.50**	0.75	1.00
Test Acc (%)	36.86	37.93	**39.86**	37.47	36.25
Train Loss	1.954	1.586	**1.251**	1.362	1.491

Table 5: Impact of α using ViT-Tiny on CIFAR-100.

decoupled) weight decay (**A3**) leads to suboptimal regularization, confirming the necessity of decoupled weight decay for Transformer-based federated training. • The complete `FedAdamW` (**A4**) consistently outperforms all ablated versions, validating the effectiveness of our joint design.

Impact of α. Table 5 evaluates the effect of the global update alignment parameter α in `FedAdamW`. As predicted by our convergence analysis (**Theorem 1**), incorporating global update direction helps suppress client drift and accelerates convergence. We observe that $\alpha = 0.5$ yields the best performance, striking a balance between local adaptivity and global consistency, in line with our theoretical insight (**Q5**).

Impact of weight decay λ. Table 6 shows that decoupled weight decay, as used in AdamW and `FedAdamW`, consistently improves test accuracy over standard Adam. `FedAdamW` generalizes well across all λ values, with $\lambda = 0.01$ performing best. This aligns with our PAC-Bayesian analysis (**Theorem 2**), where an appropriate λ balances regularization and curvature for better generalization (**Q5**).

Impact of Aggregation Strategy. Table 7 shows that our strategy, `Agg-mean-`$\boldsymbol{v}$, achieves the best balance between accuracy and communication cost. While `Agg-`$\boldsymbol{v}$ improves performance by reducing variance, full aggregation (`Agg-`$\boldsymbol{vm}$) introduces excessive communication with marginal gains. In contrast, `Agg-mean-`$\boldsymbol{v}$ attains similar benefits with only $\mathcal{O}(B)$ communication, where B is the number of blocks, demonstrating its scalability and effectiveness in stabilizing updates.

λ	**0.0005**	**0.001**	**0.005**	**0.010**	**0.020**
Local Adam	28.86	29.88	18.65	8.56	4.05
Local AdamW	35.82	36.12	36.54	36.86	36.28
`FedAdamW`	**38.26**	**39.24**	**39.55**	**39.86**	**38.56**

Table 6: **Ablation on weight decay** λ using ViT-Tiny on CIFAR-100 (Dir-0.1). FedAdamW consistently outperforms local baselines across a range of λ values.

Aggregation Strategy	Acc	Train Loss	Comm(↑)
NoAgg	$36.86_{\pm 0.11}$	1.954	5.7M
Agg-$\boldsymbol{m}$	$37.12_{\pm 0.13}$	1.854	11.4M
Agg-$\boldsymbol{v}$	$38.01_{\pm 0.12}$	1.652	11.4M
Agg-$\boldsymbol{vm}$ (FullAgg)	$38.12_{\pm 0.12}$	1.645	17.1M
Agg-`mean`-$\boldsymbol{v}$	$\mathbf{38.15}_{\pm 0.10}$	**1.601**	5.7M

Table 7: Ablation study of moment aggregation strategies of Local AdamW on CIFAR-100 with **ViT-Tiny** under Dir-0.1.

Conclusion

In this work, we proposed a novel federated optimization algorithm (`FedAdamW`) for training large-scale Transformer models. `FedAdamW` tackles the key challenges of applying AdamW in federated settings, including high variance in second-moment estimates, local overfitting under non-i.i.d. data, and inefficiencies from frequent reinitialization. It integrates second-moment aggregation, global update correction, and decoupled weight decay. We provided convergence analysis under non-convex and use the PAC Bayesian theory to support its generalization benefits. Extensive experiments on vision and language tasks verified that `FedAdamW` consistently outperforms strong FL baselines, especially on Transformer architectures, demonstrating its practical and theoretical strengths. We believe `FedAdamW` opens a new direction for adapting modern optimizers to FL such as LAMB (Chen et al. 2023) or Lion (Chen et al. 2023).

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62276182, 62276004), Peng Cheng Lab Program (No. PCL2023A08), Tianjin Natural Science Foundation (Nos. 24JCYBJC01230, 24JCYBJC01460), Tianjin Municipal Education Commission Research Plan (No. 2024ZX008).

References

An, W.; Liu, Y.; Shang, F.; and Liu, H. 2024a. Robust and faster zeroth-order minimax optimization: complexity and applications. *Advances in Neural Information Processing Systems*, 37: 37050–37069.

An, W.; Liu, Y.; Shang, F.; Liu, H.; and Jiao, L. 2024b. DEs-Inspired Accelerated Unfolded Linearized ADMM Networks for Inverse Problems. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3): 5319–5333.

An, W.; Yue, Y.; Liu, Y.; Shang, F.; and Liu, H. 2022. A numerical DEs perspective on unfolded linearized admm networks for inverse problems. In *Proceedings of the 30th ACM International Conference on Multimedia*, 5065–5073.

Bian, J.; Wang, L.; Yang, K.; Shen, C.; and Xu, J. 2024. Accelerating Hybrid Federated Learning Convergence Under Partial Participation. *IEEE Transactions on Signal Processing*, 72: 3258–3271.

Bian, J.; Wang, L.; Zhang, L.; and Xu, J. 2025a. FedALT: Federated Fine-Tuning through Adaptive Local Training with Rest-of-World LoRA. *arXiv preprint arXiv:2503.11880*.

Bian, J.; Wang, L.; Zhang, L.; and Xu, J. 2025b. LoRA-FAIR: Federated LoRA Fine-Tuning with Aggregation and Initialization Refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3737–3746.

Bottou, L. 2010. 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*, 177–186. Springer.

Chen, X.; Li, X.; and Li, P. 2020. Toward communication efficient adaptive gradient method. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, 119–128.

Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.-J.; Lu, Y.; et al. 2023. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36: 49205–49233.

Collobert, R. 2004. Large scale machine learning. *Idiap Res. Inst., Martigny, Switzerland, RR-04-42*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Fan, Z.; Hu, S.; Yao, J.; Niu, G.; Zhang, Y.; Sugiyama, M.; and Wang, Y. 2024. Locally Estimated Global Perturbations are Better than Local Perturbations for Federated Sharpness-aware Minimization. *arXiv preprint arXiv:2405.18890*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*.

Li, Y.; Chen, C.; Zhang, Y.; Liu, W.; Lyu, L.; Zheng, X.; Meng, D.; and Wang, J. 2023. Ultrare: Enhancing receraser for recommendation unlearning via error decomposition. *Advances in Neural Information Processing Systems*, 36: 12611–12625.

Li, Y.; Zhang, Y.; Liu, W.; Feng, X.; Han, Z.; Chen, C.; and Yan, C. 2025a. Multi-Objective Unlearning in Recommender Systems via Preference Guided Pareto Exploration. *IEEE Transactions on Services Computing*.

Li, Z.; Bai, T.; Yao, M.; Qin, Z.; Li, G.; et al. 2025b. MVA: Linear Attention with High-order Query-Keys Integration and Multi-level Vocabulary Decomposition. In *Forty-second International Conference on Machine Learning*.

Liu, J.; Liu, Y.; Shang, F.; Liu, H.; Liu, J.; and Feng, W. 2025a. Improving Generalization in Federated Learning with Highly Heterogeneous Data via Momentum-Based Stochastic Controlled Weight Averaging. In *Forty-second International Conference on Machine Learning*.

Liu, J.; Shang, F.; Liu, Y.; Liu, H.; Li, Y.; and Gong, Y. 2024. Fedbcgd: Communication-efficient accelerated block coordinate gradient descent for federated learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2955–2963.

Liu, J.; Shang, F.; Tian, Y.; Liu, H.; and Liu, Y. 2025b. Consistency of local and global flatness for federated learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 3875–3883.

Liu, J.; Shang, F.; Zhou, J.; Liu, H.; Liu, Y.; and Liu, J. 2025c. FedMuon: Accelerating Federated Learning with Matrix Orthogonalization. *arXiv preprint arXiv:2510.27403*.

Liu, J.; Shang, F.; Zhu, K.; Liu, H.; Liu, Y.; and Liu, J. 2025d. FedAdamW: A Communication-Efficient Optimizer with Convergence and Generalization Guarantees for Federated Large Models. *arXiv preprint arXiv:2510.27486*.

Liu, J.; Tian, Y.; Shang, F.; Liu, Y.; Liu, H.; Zhou, J.; and Ding, D. 2025e. DP-FedPGN: Finding Global Flat Minima for Differentially Private Federated Learning via Penalizing Gradient Norm. *arXiv preprint arXiv:2510.27504*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y.; Shang, F.; An, W.; Liu, J.; Liu, H.; and Lin, Z. 2023. A single-loop accelerated extra-gradient difference algorithm with improved complexity bounds for constrained minimax optimization. *Advances in Neural Information Processing Systems*, 36: 61699–61711.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Loshchilov, I.; Hutter, F.; et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5): 5.

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

Ouyang, K.; Zhang, S.-Y.; Liu, S.-L.; Tian, J.; Li, Y.; Tong, H.; Bai, H.; Wang, W. H.; and Hu, Y. 2025. Graph Learning Metallic Glass Discovery from Wikipedia. *AI for Science*.

Qian, Z.; Han, R.; Feng, W.; and Wang, S. 2024. From a Bird's Eye View to See: Joint Camera and Subject Registration without the Camera Calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 863–873.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *OpenAI Technical Report*.

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

Reddi, S. J.; Kale, S.; and Kumar, S. 2019. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Sun, Y.; Shen, L.; Sun, H.; Ding, L.; and Tao, D. 2023. Efficient federated learning via local adaptive amended optimizer with linear speedup. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 14453–14464.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L.; Bian, J.; Zhang, L.; Chen, C.; and Xu, J. 2024. Taming Cross-Domain Representation Variance in Federated Prototype Learning with Heterogeneous Data Domains. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Processing Systems*, volume 37, 88348–88372. Curran Associates, Inc.

Wei, D.; Wang, Z.; Kang, H.; Sha, X.; Xie, Y.; Dai, A.; and Ouyang, K. 2025. A comprehensive analysis of digital inclusive finance's influence on high quality enterprise development through fixed effects and deep learning frameworks. *Scientific Reports*, 15(1): 30095.

Wu, X.; Huang, F.; Hu, Z.; and Huang, H. 2023. Faster adaptive federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 10379–10387.

Xu, J.; Wang, S.; Wang, L.; and Yao, A. C.-C. 2021. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*.

Yang, L.; Liu, H.; Liu, Y.; Shang, F.; Wan, L.; and Jiao, L. 2025. Distillation guided deep unfolding network with frequency hierarchical regularization for low-dose CT image denoising. *Neurocomputing*, 130535.

Zhang, F.; Chen, G.; Wang, H.; and Zhang, C. 2024a. CF-DAN: Facial-expression recognition based on cross-fusion dual-attention network. *Computational Visual Media*, 10(3): 593–608.

Zhang, R.; Huang, Y.; Lou, Y.; Xin, Y.; Chen, H.; Cao, Y.; and Wang, H. 2025. Exploit Your Latents: Coarse-Grained Protein Backmapping with Latent Diffusion Models. 1111–1119. AAAI Press.

Zhang, R.; Lou, Y.; Xu, D.; Cao, Y.; Wang, H.; and Huang, Y. 2024b. A Learnable Discrete-Prior Fusion Autoencoder with Contrastive Learning for Tabular Data Synthesis. 16803–16811. AAAI Press.

Zhang, Y.; Chen, C.; Ding, T.; Li, Z.; Sun, R.; and Luo, Z. 2024c. Why transformers need adam: A hessian perspective. *Advances in neural information processing systems*, 37: 131786–131823.

Zhang, Y.; Chen, C.; Li, Z.; Ding, T.; Wu, C.; Kingma, D. P.; Ye, Y.; Luo, Z.-Q.; and Sun, R. 2024d. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*.

Zhou, S.; Tian, Z.; Chu, X.; Zhang, X.; Zhang, B.; Lu, X.; Feng, C.; Jie, Z.; Chiang, P. Y.; and Ma, L. 2023. FastPillars: A Deployment-friendly Pillar-based 3D Detector. *arXiv preprint arXiv:2302.02367*.

Zhou, S.; Yuan, Z.; Yang, D.; Zhao, Z.; Hu, X.; Shi, Y.; Lu, X.; and Wu, Q. 2024. Information Entropy Guided Height-aware Histogram for Quantization-friendly Pillar Feature Encoder. *arXiv preprint arXiv:2405.18734*.