FedDUAL: A Dual-Strategy with Adaptive Loss and Dynamic Aggregation for Mitigating Data Heterogeneity in Federated Learning

Pranab Sahoo *pranab_2021cs25@iitp.ac.in*
Department of Computer Science
Indian Institute of Technology Patna, India

Ashutosh Tripathi *ashutoshtripathi191@gmail.com*
Electrical and Electronics Engg. Department
Rajiv Gandhi Institute of Petroleum Technology, India

Sriparna Saha *sriparna@iitp.ac.in*
Department of Computer Science
Indian Institute of Technology Patna, India

Samrat Mondal *samrat@iitp.ac.in*
Department of Computer Science
Indian Institute of Technology Patna, India

Reviewed on OpenReview: https://openreview.net/forum?id=8M3XfmNhTZ

Abstract

Federated Learning (FL) marks a transformative approach to distributed model training by combining locally optimized models from various clients into a unified global model. While FL preserves data privacy by eliminating centralized storage, it encounters significant challenges such as performance degradation, slower convergence, and reduced robustness of the global model due to the heterogeneity in client data distributions. Among the various forms of data heterogeneity, label skew emerges as a particularly formidable and prevalent issue, especially in domains such as image classification. To address these challenges, we begin with comprehensive experiments to pinpoint the underlying issues in the FL training process, such as gradient instability and the emergence of sharp minima in the global model, both of which contribute to performance inconsistencies. Based on our findings, we introduce an innovative dual-strategy approach designed to effectively resolve these issues. First, we introduce an adaptive loss function for client-side training, meticulously crafted to preserve previously acquired knowledge while maintaining an optimal equilibrium between local optimization and global model coherence. Secondly, we develop a dynamic aggregation strategy for aggregating client models at the server. This approach adapts to each client's unique learning patterns, effectively addressing the challenges of diverse data across the network. Our comprehensive evaluation, conducted across three diverse real-world datasets, coupled with theoretical convergence guarantees, demonstrates the superior efficacy of our method compared to several established state-of-the-art approaches. The code can be found at https://github.com/Pranabiitp/FedDUAL.

1 Introduction

Federated learning (FL) has revolutionized collaborative model training by enabling multiple clients to contribute to a global model without compromising the privacy of their local data (McMahan et al., 2017).

This decentralized strategy avoids the need for sending data to a central server, thus maintaining data privacy. As the digital landscape evolves, with an increasing number of distributed data sources emerging from mobile devices, healthcare institutions, and Internet of Things (IoT) networks, FL has emerged as a pivotal solution for training sophisticated deep networks across geographically dispersed and heterogeneous environments (Bonawitz et al., 2016), Sahoo et al. (2024b), (Hu et al., 2024). However, a significant practical obstacle encountered during federated training is data heterogeneity in the form of skewness in labels and quantity of the data across various clients (Kairouz et al., 2021), (Li et al., 2020). Diverse user behaviors can lead to significant heterogeneity in the local data of different clients, leading to non-independent and identically distributed (non-IID) data. This variability can introduce biases in model training, leading to unstable convergence and potentially degrading the model's performance or making it counterproductive (Li et al., 2022), (Zhao et al., 2018). While FedAvg (McMahan et al., 2017) is effective and widely used, it often falls short in accuracy and convergence with static aggregation methods. These methods combine model updates from different clients in a fixed manner, failing to adapt to heterogeneous data distributions and client drift, as discussed in (Karimireddy et al., 2020).

Previous studies have addressed the issue of client drift by implementing penalties for deviations between client and server models (Li et al., 2020), (Li et al., 2021a), employing variance reduction techniques during client updates (Karimireddy et al., 2020), (Acar et al., 2021), or utilizing novel aggregation methods on the server side (Chen et al., 2023), (Chowdhury & Halder, 2024).

1.1 Motivation

Prior studies by Yashwanth et al. (2024), Hu et al. (2024) have demonstrated that in non-IID scenarios, federated models tend to converge to 'sharp minima', resulting in significant performance degradation and compromised generalizability. In this study, we investigate the root causes of this phenomenon and propose a novel solution to mitigate its effects. Our study begins with a detailed analysis of loss landscapes for FedAvg-trained models across IID and non-IID data distributions. Figure 1 visually depicts the loss landscapes of two models on the FMNIST dataset with systematic parameter perturbations. The model trained on IID data exhibits a notably smoother and wider valley in its loss landscape, suggesting greater robustness and better generalization. In contrast, the model trained on non-IID manifests sharper peaks and narrower valleys, indicating higher sensitivity to parameter variations and potential overfitting. These visualizations offer strong evidence that in the presence of non-IID data, the FedAvg algorithm achieves suboptimal generalization. Motivated by this observation, we investigate the underlying mechanisms by analyzing gradient norms to identify which parts of the neural network are most affected by data heterogeneity. Our findings, presented in Fig. 2 reveal a notable pattern: in non-IID scenarios, the gradient norms of the final layers, including the classification layer, exhibit significant amplification compared to their IID counterparts. Such amplification leads to model instability, impedes convergence, and ultimately compromises the generalizability of the federated model. Our investigation suggests that effective federated training in non-IID environments necessitates targeted adjustments during server-side aggregation, particularly for these highly affected layers, to achieve performance comparable to IID settings.

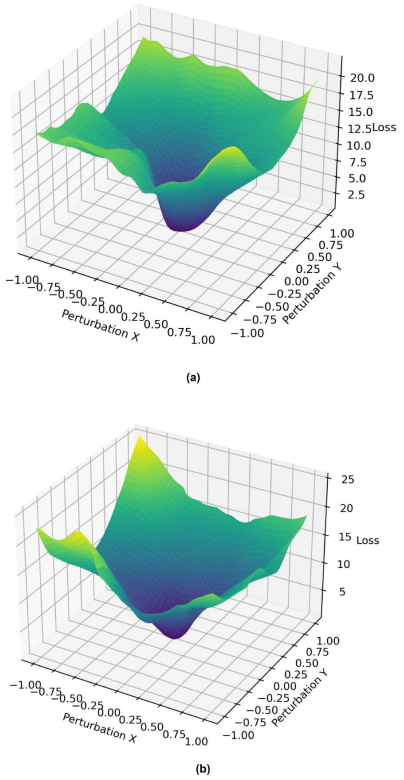
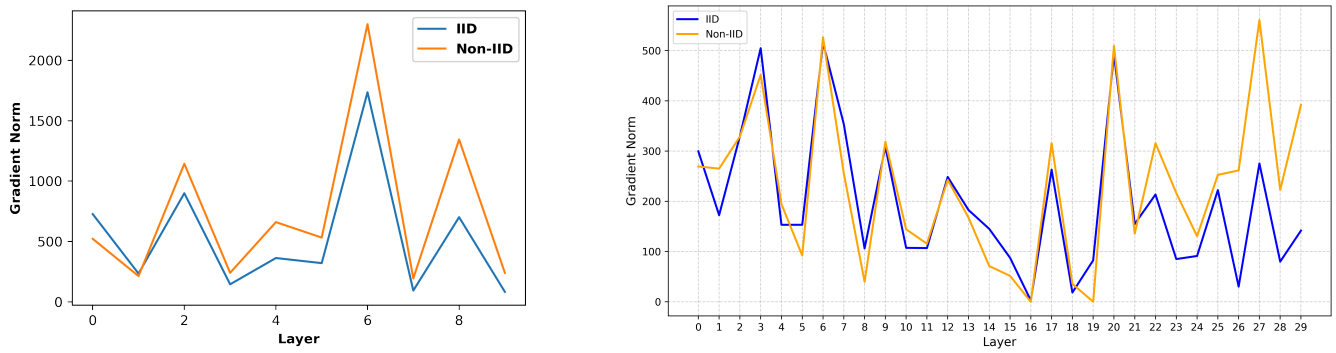


Figure 1: Visualization of the loss surface for the global model trained on the FMNIST dataset using the FedAvg algorithm: (a) depicts the loss landscape when trained on IID data, while (b) illustrates the landscape for non-IID data distribution.

This prompts one critical question: Can static aggregation methods effectively address severe non-IID data distributions across clients while maintaining higher convergence, performance, and generalizability in federated models? The answer is decidedly negative. Static aggregation methods inherently struggle with the dynamic heterogeneity present in federated networks, where adjusting parameters based on client distributions and performance in each communication round is crucial. Although incorporating predetermined parameters into the aggregation process may provide some partial mitigation, these methods fail to address the complex challenges posed by non-IID data distributions. A more dynamic and nuanced approach is necessary to effectively manage these multifaceted issues. To address this challenge, we apply dynamic aggregation to the model's final layers, where gradient norms fluctuate significantly in non-IID scenarios, while using traditional aggregation (FedAvg) for the lower layers. For dynamic aggregation, we leverage the concept of Wasserstein Barycenter (Agueh & Carlier, 2011), derived from optimal transport theory, to integrate client-specific learning behaviors in these affected layers. By minimizing discrepancies from non-IID data, the Wasserstein Barycenter helps to align gradients from diverse clients, offering precise model updates. This approach ensures fair aggregation, adapts to data heterogeneity, reduces bias, and enhances robustness, ultimately leading to more stable model convergence and improved generalization.



(a) (b)

Figure 2: Comparison of gradient norms between models trained on IID and non-IID datasets using the FedAvg algorithm. (a) FMNIST dataset using LeNet model. (b) CIFAR10 dataset using VGG16 model.

In addition to the server-side dynamic aggregation, we introduce an adaptive loss function for local training on the client side. This function allows clients to effectively explore the minima on their local datasets while preventing overfitting, thereby enhancing local optimization. Simultaneously, it preserves the global knowledge of the federated model, ensuring that the benefits from all participating clients are integrated. By incorporating a regularization parameter, β, the local loss function dynamically balances the trade-offs between local and global objectives. The contributions of this paper are as follows:

- We introduce FedDUAL, an innovative dual-strategy approach designed to effectively develop a robust and generalized federated model in highly heterogeneous data environments.
- We introduce an adaptive loss function for client-side training to balance the trade-offs between local and global objectives.
- Instead of straightforward server-side averaging, we propose a dynamic aggregation technique that uses Wasserstein Barycenter to reduce the effects of non-IID data by integrating the learning behaviors of participating clients.
- We conducted extensive experiments on three real-world datasets, demonstrating significant performance improvements over state-of-the-art methods and offering theoretical convergence guarantees for both convex and non-convex scenarios.

2 Related Work

The landscape of FL research has been significantly shaped by efforts to address data heterogeneity challenges, yielding a diverse array of innovative solutions. These approaches can be divided into three primary categories: (1) client drift mitigation strategies, which refine local client objectives to foster better alignment with the global model (Li et al., 2021a), (Karimireddy et al., 2020), (Acar et al., 2021), (Luo et al., 2021), (Li et al., 2023) (2) aggregation scheme optimization, aimed at enhancing server-side fusion of model updates (Hsu et al., 2019), (Lin et al., 2020), (Wang et al., 2020b), (Wang et al., 2020a) and (3) personalized FL, which tailors models to individual clients (Fallah et al., 2020), (Sattler et al., 2020), (Bui et al., 2019). Our research primarily focuses on two interconnected aspects of FL: mitigating client drift and optimizing server-side aggregation, and we will discuss the same in the literature review.

McMahan et al. (2017) introduced FL as an extension of local Stochastic Gradient Descent (SGD) (Stich, 2019), enabling increased local gradient updates on client devices before server synchronization and significantly reducing communication costs in identically distributed data settings. However, the method faces considerable obstacles when dealing with non-IID scenarios. Since then, various methods have emerged to address the challenge of data heterogeneity in FL (Li et al., 2019), (Yang et al., 2021), (Lin et al., 2018), (Hsu et al., 2019). FedProx (Li et al., 2020) incorporates a proximal regularization term to the optimization function to reduce model drift and addresses client stragglers. However, this term can also lead to local updates being biased towards the previous global model, which may result in misalignment between local and global optima. Building on previous work, Acar et al. (2021) introduced a dynamic regularization term to align local updates more closely with global model parameters, effectively reducing client drift caused by local model overfitting. Sun et al. (2023) further advanced the field with a momentum-based algorithm that accelerates convergence by combining global gradient descent with a locally adaptive optimizer. Similarly, several studies use variance reduction techniques, such as SCAFFOLD (Karimireddy et al., 2020). However, this approach often results in higher communication costs due to the transmission of additional control variates (Halgamuge et al., 2009). FedPVR (Li et al., 2023) addresses these limitations by reassessing FedAvg's performance on deep neural networks, uncovering substantial diversity in the final classification layers. By proposing a targeted variance reduction strategy focused solely on these final layers, FedPVR outperforms several benchmarks. MOON (Li et al., 2021a) introduces an innovative model-contrastive framework leveraging a contrastive loss to align local client representations with the global model, effectively mitigating client drift, and enhancing convergence, particularly in challenging non-IID environments. Luo et al. (2021) introduced CCVR (Classifier Calibration and Variance Reduction), which employs a classifier regularization and calibration method to enhance federated learning performance. CCVR's approach involves fine-tuning the classifier using virtual representations sampled from an approximated Gaussian mixture model. Shi et al. (2023) introduced a novel differentially private federated learning (DPFL) algorithm that integrates the Sharpness-Aware Minimization (SAM) optimizer to enhance stability and robustness against weight perturbations. By generating flatter loss landscapes and reducing the impact of differential privacy (DP) noise, it mitigates performance degradation and achieves state-of-the-art results, supported by theoretical analysis and rigorous privacy guarantees. Fanì et al. (2024) proposed FED3R, leveraging Ridge Regression on pre-trained features to tackle non-IID data challenges, effectively mitigating client drift, enhancing convergence, and optimizing efficiency in cross-device settings.

Another line of research targets optimizing server-side aggregation in FL. For instance, Hsu et al. (2019) investigated the impact of non-IID data on visual classification by creating datasets with diverse distributions and found that increased data heterogeneity negatively affected performance, leading them to propose server momentum as a potential solution. FedNova (Wang et al., 2020b) addressed the problem of objective inconsistency due to client heterogeneity in federated optimization by introducing a normalized averaging technique, which resolves this inconsistency and ensures rapid error convergence. Addressing the limitations of traditional parameter averaging methods, Lin et al. (2020) introduced ensemble distillation for model fusion. This approach allows for the flexible aggregation of heterogeneous client models by training a central classifier on unlabeled data, using the outputs from the client models as guidance. FedMRL (Sahoo et al., 2024a) introduced a novel framework by using a loss function that promotes fairness among clients and employed a multi-agent reinforcement learning for personalized proximal terms , and a self-organizing map to dynamically adjust server-side weights during aggregation.

3 Definitions

In this subsection, we summarize the key mathematical definitions used in the paper to ensure clarity.

Kullback–Leibler (KL) Divergence. The KL divergence measures how one probability distribution diverges from a second reference distribution. For two distributions P and Q over the same probability space, it is defined as:

$$D_{\mathrm{KL}}(P \,\|\, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \tag{1}$$

Wasserstein Distance. The Wasserstein distance (also known as the Earth Mover's Distance) measures the optimal cost of transporting mass to transform one probability distribution into another. For two distributions μ and ν, the p-Wasserstein distance is defined as:

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x,y)^p \, d\gamma(x,y) \right)^{1/p}, \tag{2}$$

where $\Gamma(\mu, \nu)$ denotes the set of all couplings with marginals μ and ν, and $d(\cdot, \cdot)$ is a ground metric. We leverage the Wasserstein distance and its barycenter to aggregate client models in a manner that accounts for data heterogeneity.

Wasserstein Barycenter. Given distributions $\{\mu_k\}_{k=1}^K$ and weights $\{\lambda_k\}_{k=1}^K$, the Wasserstein barycenter $\hat{\mu}$ is defined as the distribution minimizing the weighted sum of Wasserstein distances:

$$\hat{\mu} = \arg\min_{\nu} \sum_{k=1}^{K} \lambda_k W_p(\mu_k, \nu). \tag{3}$$

This barycenter allows us to aggregate the last-layer representations of client models more effectively than simple averaging, improving robustness to non-IID data.

4 Methods and Materials

We consider a practical FL scenario with non-IID data distribution among K independent clients, each with local training data $D_k(x, y)$, where (x, y) denoting the data points. We initialize the global model weights θ_r^g and share it to the participating clients. The clients download the weighs from the server and train it using their local dataset $D_k(x, y)$. The updated model parameters θ_k^r from each client k for r^{th} communication round are uploaded to the server to aggregate into a global model θ_r^g. Our objective is to develop a robust global model by collaboratively training local models across clients, even under varying heterogeneous conditions. To formalize, we define the optimal global model θ^* as follows:

$$\theta^* := \arg\min_{\theta} F(\theta), \; F(\theta) := \frac{1}{K} \sum_k f_k(\theta), \tag{4}$$

where $f_k(\theta)$ is defined in Eq. 5.

$$f_k(\theta) = E_{(x,y)\sim D_k}[\ell(f_\theta(x), y)], \tag{5}$$

where θ represents the global model parameters, $f_\theta(x)$ is the model's prediction, and ℓ is the loss function.

4.0.1 Client Side Update.

At the beginning of each round t, the server randomly selects a subset $S_t \subset K$ of clients to participate in the federated training process and subsequently shares the current global model θ_r^g to these participating clients.

Each client updates its local model by initializing with the global model parameters ($\theta_k^r = \theta_g^r$) and then updates its local model by minimizing the local objective function. For local training, we have developed an adaptive objective function that balances local loss with the divergence between local and global models. The extent of this divergence is quantified using the Kullback-Leibler (KL) divergence (Csiszár, 1975), which effectively compares the probability distributions of the local model weights $p^k(w)$ with the global model weights $q(w)$. The KL divergence is mathematically defined in Eq. 7. To obtain the probability distributions of the local and global model weights, we first flatten the weights and then apply the softmax function. This process yields the desired probability distributions (p), as specified in Eq. 6.

$$\mathrm{p} = \frac{\exp(\text{flatten weights})}{\sum \exp(\text{flatten weights})} \tag{6}$$

$$D_{\mathrm{KL}}(p^k \| q) = \sum_i p_i^k(w) \log\left(\frac{p_i^k(w)}{q_i(w)}\right) \tag{7}$$

where p_i^k and q_i are the probabilities associated with the ith component of the weight vectors. The local model must excel on local data while maintaining alignment with the global model to enhance overall generalization. This balance between minimizing local loss and aligning with the global model is defined as local adaptive function $\tilde{f}_k(\theta)$ in Eq. 8.

$$\tilde{f}_k(\theta) = (1-\beta) * f_k(\theta) + \beta * D_{\mathrm{KL}}(p^k \| q), \tag{8}$$

where $f_k(\theta)$ is cross-entropy loss for k^{th} client and β is a regularization parameter and should be adaptive to account for the performance discrepancy between the local and global models. When the local model substantially outperforms the global model, β should increase to enforce greater alignment. Conversely, if the models perform similarly, β should decrease, allowing the local model to focus more on local optimization. The definition of β is given in Eq. 9.

$$\beta = \sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}^k) \tag{9}$$

where σ is the sigmoid function, $\mathcal{A}_{\text{local}}^k$ represents the local model accuracy, and $\mathcal{A}_{\text{global}}^k$ is the global model accuracy for client k. We calculated the global model's accuracy $\mathcal{A}_{\text{global}}^k$ for client k by evaluating it on the training data of client k prior to performing local updates in the current round. Incorporating the adaptive parameter β in Eq. 8, the adaptive loss function for client k is represented in Eq. 10.

$$\begin{aligned}\mathcal{L}_{\text{adaptive}}^k = (1 - (\sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}^k)) * \mathcal{L}_{\text{local}}^k \\ + \sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}) * D_{\mathrm{KL}}(p^k \| q)\end{aligned} \tag{10}$$

After defining the adaptive loss function for each client, we optimize the local model parameters using stochastic gradient descent (SGD). The gradient update for the local model weights w_k based on the adaptive loss function is given in Eq. 11.

$$w_k^{t+1} = w_k^t - \eta \nabla_w \mathcal{L}_{\text{adaptive}}^k(w_k^t), \tag{11}$$

where η is the local learning rate. Expanding the gradient term $\nabla_w \mathcal{L}_{\text{adaptive}}^k(w_k^t)$, we obtain Eq. 12:

$$\nabla_w \mathcal{L}_{\text{adaptive}}^k(w_k^t) = (1 - (\sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}^k)) \nabla_w \mathcal{L}_{\text{local}}^k(w_k^t) + \sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}) \nabla_w D_{\mathrm{KL}}(p^k \| q). \tag{12}$$

The KL divergence term, $\sigma(\mathcal{A}_{\text{local}}^k - \mathcal{A}_{\text{global}}^k) \nabla_w D_{\mathrm{KL}}(p^k \| q)$ in Eq. 12, acts as a regularizer to keep the local model gradients aligned with the global model gradients, thereby preserving model coherence despite non-IID data. The adaptive coefficient β (Eq. 9) is dynamically computed as a function of the performance gap ($A_{\text{local}}^k - A_{\text{global}}^k$). When the global model outperforms the local one, β tends toward 0 (via the sigmoid function), thus increasing the weight on the local loss term $(1-\beta)$ and enabling the client to focus more

on its local data. Conversely, when the local model performs better than the global, β increases toward 1, strengthening the KL term to preserve global knowledge rather than blindly aligning the models. The rationale is that superior local accuracy often reflects *overfitting* to non-IID client data. A higher β in such cases regularizes the local model by constraining it to remain close to the global parameter manifold, improving overall generalization. Conversely, when the global model generalizes better, a smaller β allows the client to emphasize local learning. For example, a client trained only on classes 0–4 may achieve high local accuracy but would poorly generalize to unseen classes 5–9. The higher β and associated KL regularization preserve the global model's multi-class knowledge, thereby preventing catastrophic forgetting.

Note that we employed KL divergence on the softmax of the parameters to capture distributional alignment between local and global models rather than direct numerical differences. By transforming flattened parameters into probability distributions through softmax, we interpret model weights as expressing relative importance rather than absolute magnitude. KL divergence thus quantifies how the structural configuration of model parameters diverges between clients and the server. Compared to alternatives, logit-based distances capture only output-level similarity and overlook the underlying parameter drift that degrades generalization. Layer-wise representation distances would require computing activations on a reference dataset, introducing computational overhead and ambiguity regarding data selection in heterogeneous settings. Similarly, the L2 parameter distance (as in FedProx) reflects magnitude differences but not distributional structure which is an essential factor for understanding model behavior. Empirically, we find KL divergence advantageous because its asymmetry aligns with the local to global adaptation objective and its bounded nature ensures stable gradients during optimization.

4.0.2 Server Side Update.

After obtaining the weights from the participating clients at round t, the server calculates the Wasserstein Barycenter to effectively aggregate the weights of the last layers of the client models. Computing exact Wasserstein Barycenter can be computationally expensive, so we have approximated it using the Sinkhorn-Knopp (Knight, 2008) algorithm for efficient computation. We consider the local model weights as distributions and assign equal importance to each client in the computation of the Wasserstein Barycenter ($\bar{\mu}$). This barycenter represents the distribution that minimizes the sum of Wasserstein distances to the individual client gradient distributions, as formally defined in Eq. 13.

$$\hat{\mu} = \arg\min_{\nu} \sum_{k=1}^{K} \lambda_k W(\mu_k, \nu) \tag{13}$$

where λ_k are weights corresponding to the importance or reliability of the client k. The Wasserstein distance $W(\mu_k, \mu_j)$ between two gradient distributions μ_k and μ_j of clients j and k is defined in Eq. 14.

$$W(\mu_k, \mu_j) = \left(\inf_{\gamma \in \Gamma(\mu_k, \mu_j)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p \, d\gamma(x, y) \right)^{1/p} \tag{14}$$

where $\Gamma(\mu_k, \mu_j)$ denotes the set of all couplings (or joint distributions) γ on $\mathcal{X} \times \mathcal{X}$ with marginals μ_k and μ_j respectively, and $d(x, y)$ is the distance between points x and y in the metric space $\mathcal{X}$. After that, we use Sinkhorn-Knopp algorithm to calculate the Wasserstein Barycenter.

This barycenter is computed iteratively, starting by calculating a scaling factor γ using Eq. 15, followed by Eq. 16.

$$\gamma = \exp\left(-\frac{\mathrm{W}(\bar{p}, p_i)}{\epsilon} \right) \tag{15}$$

$$\bar{p}_{\text{new}} = \frac{\sum_i \lambda_i \gamma p_i}{\sum_i \lambda_i \gamma_i} \tag{16}$$

where $\bar{p}$ is the current estimate of the barycenter, p_i refers to the ith client's gradient distribution, ϵ is a small positive constant, and the iterations continue until convergence. After few iterations, we get the Wasserstein barycenter that is used to update the global model weights. We update the the global model weights for the last layers by substracting them from the calculated Wasserstein barycenter for effectively aggregating the updates from the last layers. Our gradient analysis (Fig. 2) indicates that non-IID data disproportionately amplifies gradient norms in the final layers, leading to heightened instability compared to IID conditions. Traditional averaging methods aggregate client updates in Euclidean space without accounting for the underlying distributional differences across clients. The Wasserstein Barycenter addresses this limitation by operating in the space of probability distributions, finding the optimal aggregation point that minimizes distributional discrepancies (Wasserstein distances) to all client updates. This geometry-aware approach provides more robust aggregation by explicitly accounting for heterogeneous client learning behaviors, reducing the bias and instability caused by non-IID data. The algorithm of proposed method FedDUAL is given in the Algorithm 1. The proof of the convergence for both convex and non-convex settings for the proposed method can be found in Section A of the Appendix.

Algorithm 1 FedDUAL

Input: Number of clients K, Number of communication rounds T, and Global model $\mathcal{G}$.
Output: Trained global model $\mathcal{G}^*$.
Define a mask $e \in \{0,1\}^d$, where $e_j = 1$ for the last few layers and 0 for the rest layers.
Let $S_{naive} = \{j : e_j = 0\}$ and $S_{dynamic} = \{j : e_j = 1\}$.
Initialize global model weights θ^g
for $t = 1$ to T **do**
  Sample a subset of clients $\mathcal{S}_t \subseteq \{1, \ldots, K\}$
  Initialize lists: local model weights $\mathcal{W} \leftarrow []$, gradients $\Delta \leftarrow []$
  for each client $k \in \mathcal{S}_t$ **do**
    Initialize local model $\mathcal{M}_k$ with global weights θ^g.
    Train $\mathcal{M}_k$ on local dataset $\mathcal{D}_k$ using adaptive loss function defined in Eq. 10.
    $\mathcal{W} \leftarrow \mathcal{W} \cup \{\theta_k\}$ ▷ Store local model weights θ_k
    Compute gradients ∇_k for $\mathcal{M}_k$
    $\Delta \leftarrow \Delta \cup \{\nabla_k\}$ ▷ Store gradients ∇_k
  end for
  for $j \in \{1, \ldots, d\}$ **do**
    if $e_j = 1$ **then** ▷ Layer belongs to $S_{dynamic}$
      Extract last layers' gradients $\{\nabla_k[j]\}$ from Δ
      Compute Wasserstein Barycenter of last layer j gradients $\bar{\nabla}_j$
      Update global model's last layer j weights $\theta^g[j] \leftarrow \theta^g[j] - \bar{\nabla}_j$
    else ▷ Layer belongs to S_{naive}
      Perform Federated Averaging for layer j:
      $\theta^g[j] \leftarrow \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} \theta_k[j]$
    end if
  end for
end for
$\mathcal{G}^* \leftarrow \theta^g$ ▷ Final trained global model

5 Experimental Results

5.1 Experimental Setup

To assess the effectiveness of the proposed FedDUAL approach, we conducted extensive experiments using three widely recognized classification benchmarks: CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky, 2009), and FMNIST (Xiao et al., 2017). To simulate real-world non-IID data distributions, we employed a client-wise partitioning strategy based on the Dirichlet distribution (Hsu et al., 2019). This distribution is governed by a concentration parameter α, which controls the degree of data

Table 1: Top-1 accuracy (%) on CIFAR10, CIFAR100, and FMNIST datasets. The values in bold represent the highest accuracy achieved. '*' denotes algorithms that failed to achieve convergence.

	CIFAR10	CIFAR100	FMNIST
FedAvg	46.68 ± 0.25	26.88 ± 0.18	81.70 ± 0.20
FedProx	47.58 ± 0.30	26.89 ± 0.22	80.54 ± 0.28
FedNova	48.44 ± 0.35	*	*
FedBN	*	26.88 ± 0.19	81.36 ± 0.23
FedDyn	43.97 ± 0.40	18.27 ± 0.32	71.86 ± 0.45
MOON	46.57 ± 0.28	28.50 ± 0.25	80.09 ± 0.27
SCAFFOLD	*	*	*
FedPVR	42.26 ± 0.42	23.78 ± 0.31	80.32 ± 0.33
Proposed	**48.70 ± 0.20**	**29.15 ± 0.24**	**81.99 ± 0.21**

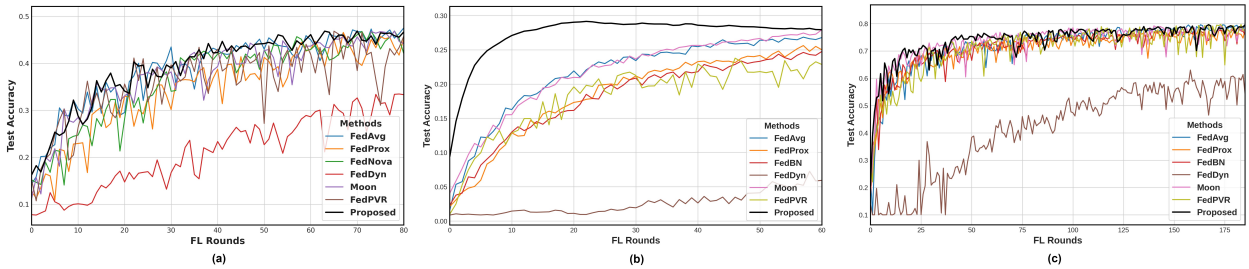


Figure 3: Learning curves comparing the proposed method with baselines across various datasets: (a) CIFAR-10, (b) CIFAR-100, and (c) FMNIST.

heterogeneity among clients. Lower α values result in more skewed data distributions, closely mimicking uneven data partitions. In all experiments, we set $\alpha = 0.01$ to simulate severe data heterogeneity, closely approximating real-world conditions. Throughout the communication rounds, each client retains a fixed local data partition. To evaluate the global model's classification performance, we use a separate test dataset maintained at the server, which remains unseen during training. For our experiments, we used LeNet (LeCun et al., 1998) for FMNIST dataset and a pre-trained VGG16 (Simonyan & Zisserman, 2015) for CIFAR-10 and CIFAR-100 dataset, following the methodology outlined in (Hu et al., 2024). We applied the proposed dynamic aggregation mechanism only to the last two layers of these models. Our setup involved 100 clients, with 10% randomly sampled per communication round, and a batch size of 32. Each client performed three

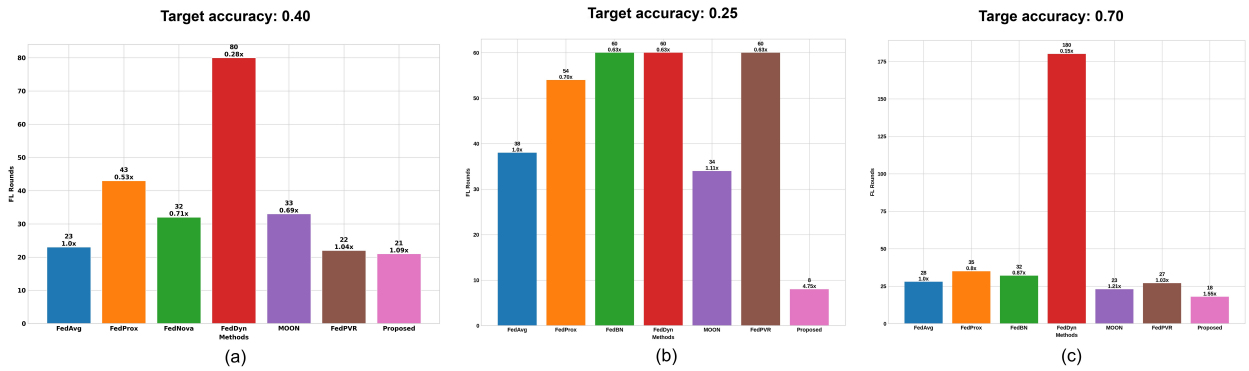


Figure 4: Number of FL rounds required to reach the target accuracy for the proposed method and other baselines on different datasets: (a) CIFAR-10, (b) CIFAR-100, and (c) FMNIST.

local epochs of model updates. We have computed each result three times with different seed values and reported the mean value with standard deviation. To determine the optimal client learning rate for each experiment, we conducted a grid search over $0.05, 0.01, 0.2, 0.3$. For the baseline FedProx, we tested proximal values of 0.001, 0.1, 0.4, 0.7 to find the optimal setting, and for FedNova, we evaluated proximal SGD values from 0.001, 0.003, 0.05, 0.1, following the recommendations in Li et al. (2024). Across all experiments, we used the Adam optimizer for consistency. We have run each algorithm three times and reported the average outcome. The experimental setup utilized an NVIDIA Quadro RTX 4000 GPU boasting 40GB of memory. The implementation was crafted using Python [1], leveraging the TensorFlow framework [2] utilizing Windows 11.

5.2 Comparison with the State-of-the-art Methods

5.2.1 Baseline.

We evaluate the proposed FedDUAL method against eight notable state-of-the-art (SOTA) FL baselines, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedNova (Wang et al., 2020b), SCAFFOLD (Karimireddy et al., 2020), FedBN (Li et al., 2021b), FedDyn (Acar et al., 2021), MOON (Li et al., 2021a) and FedPVR (Li et al., 2023).

5.2.2 Comparison of Accuracy.

The results, detailed in Table 1, reveal that many recent FL methods often fall short compared to the standard FedAvg baseline. In contrast, our proposed method consistently achieves SOTA performance, surpassing FedAvg along with other baselines across all evaluated scenarios. Furthermore, our approach exhibits remarkable adaptability across diverse datasets. Unlike some algorithms that excel on specific datasets but falter on others, the proposed FedDUAL consistently outperforms baselines across a wide range of data environments. This improvement suggests that our method addresses fundamental challenges in FL, potentially offering a more generalizable solution to the issues posed by data heterogeneity in federated settings. We also observed that FedNova, FedBN, and Scaffold did not perform effectively in our experimental setup.

5.2.3 Comparison of Convergence.

Figure 3 compares the learning curves of our method with baselines, while Fig. 10 in the Appendix includes the corresponding curves with error bars. Across all datasets, our method consistently converges faster and achieves higher final accuracy. Although the number of communication rounds varies by dataset, performance generally saturates by the final round. Notably, our method not only attains a more robust final model but also displays markedly faster convergence across all datasets examined. This effectiveness is further highlighted in Fig. 4, where it consistently reaches target accuracy with far fewer communication rounds compared to baseline approaches. However, the proposed method incurs a higher computation cost due to Wasserstein barycenter based aggregation at the server side. We have provided a thorough computational complexity analysis and runtime analysis in the Section C and the Section D of the appendix respectively.

5.3 Validation of the Motivation

To substantiate our claim that the proposed method yields models in flatter loss landscapes compared to FedAvg, we conducted a comparative analysis. Using VGG-16 models trained on the FMNIST dataset under non-IID conditions ($\alpha = 0.01$), we visualized their respective loss landscapes following the approach outlined in Li et al. (2018). Figure 6 in the Appendix depicts these landscapes, with each model centrally located within its respective terrain. The visualization reveals that our proposed method situates the model in a notably flatter region compared to FedAvg. This finding supports our assertion that our approach guides federated training towards more stable and generalizable solutions, characterized by flatter loss landscapes. The performance improvement of the proposed models stems from two key innovations: a Wasserstein

[1] https://www.python.org/
[2] https://www.tensorflow.org/

Barycenter-based aggregation for final layer gradients, mitigating client drift in heterogeneous data environments, and an adaptive loss function balancing local optimization with global consistency during client training. This synergistic approach preserves global knowledge while promoting client-specific optimization, addressing fundamental FL challenges.

6 Ablation Study

In our ablation study, we performed all experiments on the FMNIST dataset with $\alpha = 0.01$. The study comprised four types of experiments: (1) performance analysis of the individual modules, (2) assessment of the impact of dynamic aggregation across different neural network layers, (3) hyperparameter analysis, and (4) evaluation of various levels of data heterogeneity.

6.0.1 Performance Analysis of Individual Modules

To assess the effectiveness of the proposed adaptive loss and dynamic aggregation techniques, we conducted three ablation experiments across FMNIST, CIFAR-10, and CIFAR-100. The results for FMNIST are shown in Table 2. In the first experiment, we employed only the adaptive loss alongside standard server-side aggregation. Notably, this configuration underperforms FedAvg across all datasets, indicating that adaptive loss alone cannot effectively address data heterogeneity—likely due to its limited capacity to improve generalization despite fostering local-global alignment. The second experiment implemented our dynamic aggregation technique at the server, while retaining the conventional cross-entropy loss function locally. Finally, the third experiment combined both proposed methods: the adaptive loss function and the dynamic aggregation technique. As evidenced by Table 2, the integration of both proposed approaches in the third experiment yielded the highest accuracy, highlighting the impact of our dual strategy on model performance. The learning curves for these experiments using FMNIST dataset are illustrated in Fig. 7 of the Appendix.

6.0.2 Impact of Dynamic Aggregation on Different Network Layers

To substantiate our decision to apply dynamic aggregation technique selectively to last layers, we examined its impact across various layers of the neural network. Our earlier findings highlighted that data heterogeneity primarily affects last layers of the network. Figure 5 illustrates that random utilization of the dynamic aggregation to all layers diminishes performance. Conversely, targeted implementation on layers proximal to the classifier yielded optimal accuracy and convergence. These outcomes validate our hypothesis and demonstrate the method's efficacy in mitigating heterogeneity-induced issues. By focusing our dynamic aggregation technique on the most susceptible layers, we directly address the core challenge of data heterogeneity in federated training, resulting in enhanced model performance and faster convergence.

Table 2: Ablation study of FedDUAL across different datasets.

Adaptive Loss	Dynamic Agg.	Dataset	Acc. (%)
✓	✗		80.70 ± 0.22
✗	✓	FMNIST	80.91 ± 0.25
✓	✓		**81.99 ± 0.18**
✓	✗		41.05 ± 0.12
✗	✓	CIFAR10	46.50 ± 0.15
✓	✓		**48.70 ± 0.20**
✓	✗		25.05 ± 0.11
✗	✓	CIFAR100	27.01 ± 0.17
✓	✓		**29.15 ± 0.24**

6.0.3 Hyperparameter Analysis

In the proposed architecture, there are two key hyperparameters to consider: the scaling factor (γ) and the number of iterations used to compute the Wasserstein Barycenter. The proposed FedDUAL approach utilizes dynamic server-side aggregation by applying the Wasserstein Barycenter concept to combine the weights of the final layers from local models. This iterative process involves a small positive constant (ϵ) to determine the scaling factor (γ). To optimize performance, we conducted two sets of experiments on the FMNIST dataset with $\alpha = 0.01$, each exploring a range of values for these crucial hyperparameters. The hyperparameter ϵ influences the sensitivity of the barycenter calculation to variations in Wasserstein distance. A smaller ϵ makes the barycenter more responsive to differences in Wasserstein distance, while a larger ϵ diminishes this sensitivity. This impacts how the barycenter integrates each distribution according to its distance from the current estimate. During the iterative update of the barycenter, ϵ affects the scaling factor γ applied to each

distribution. An excessively small ϵ can result in slow or potentially non-existent convergence due to minimal scaling factor, whereas a too-large ϵ may cause oversmoothing, reducing the barycenter's effectiveness in accurately representing the distributions. For this setting we have fixed the number of iterations to compute Wasserstein Barycenter as 150. Figure 8 in the Appendix shows test accuracy across different ϵ values, indicating that larger ϵ can degrade performance or hinder convergence. Figure 9 in the Appendix presents the corresponding learning curves for these settings. The number of iterations in the Wasserstein Barycenter function is another critical hyperparameter that affects both the accuracy and efficiency of the barycenter computation. Generally, more iterations enhance convergence and accuracy, ensuring that the barycenter more closely approximates the optimal value. However, increasing the number of iterations also prolongs computation time, necessitating a balance between accuracy and efficiency. Finding the optimal number of iterations involves a trade-off: too few iterations may result in suboptimal outcomes, while too many can yield diminishing returns in accuracy. To achieve the best performance, begin with a reasonable default value, monitor convergence by observing changes in the barycenter, and adjust iteratively based on empirical results and available computational resources. For this setting, we fixed the *epsilon* value as 0.0001, which yields the highest results in previous experiment. Figure 11 illustrates the test accuracy for different values of iterations to calculate Wasserstein Barycenter, suggesting that larger iterations may adversely affect performance. Figure 12 presents the corresponding learning curves for these settings. From both experiments, we observe that the highest performance is achieved with $\epsilon = 0.00001$ and 150 iterations. Therefore, to optimize performance, it is advisable to set ϵ to a smaller value while keeping the number of iterations between 100 and 150.

6.0.4 Experiment on Different Level of Data Heterogeneity

Figure 13 illustrates the accuracy of the proposed method and various baselines across different levels of data heterogeneity on the FMNIST dataset. In this context, heterogeneity is quantified by α, with lower values indicating greater data heterogeneity. The results show that as α decreases, the test accuracy for all models increases, because data heterogeneity among clients is decreased. Remarkably, the proposed method consistently achieves the highest test accuracy and exhibits the slowest performance decline compared to other algorithms, demonstrating superior performance of the proposed method on varying degrees of non-IID data partitioning. The learning curve is presented in Fig. 14.

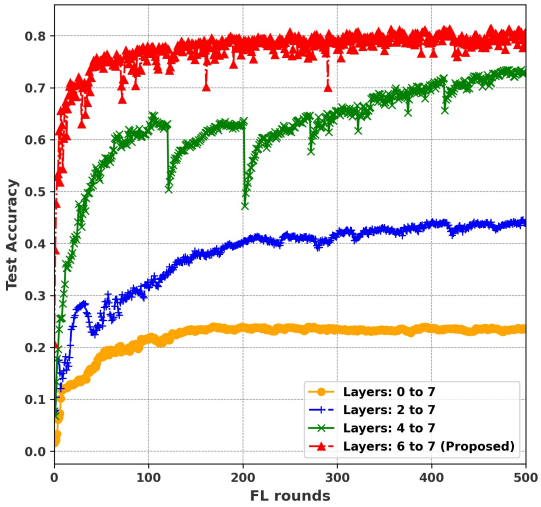


Figure 5: Illustration of the Dynamic aggregation method applied across various layers of the neural network.

7 Conclusion

This research presents a novel approach to address the challenges posed by data heterogeneity among clients in the federated approach. We systematically analyze the factors contributing to federated model performance degradation under severe data heterogeneity and propose an architecture incorporating dual-strategy innovations. First, we implement an adaptive loss function for client-side training. Second, we create a dynamic aggregation strategy for server side aggregation, tailored to client-specific learning behaviors. The proposed FedDUAL effectively overcomes the challenges of heterogeneous data, outperforming eight SOTA baselines. It demonstrates faster convergence and consistently improved performance, making it an excellent solution for large-scale FL applications in real-world scenarios. Our approach's flexibility paves the way for research into hybrid federated learning models that adapt to changing client environments and data. Future studies will focus on integrating personalized learning paths to enhance model adaptability and efficiency across various datasets.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization, 2021.

Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin. Federated user representation learning. *arXiv preprint arXiv:1909.12535*, 2019.

Dengsheng Chen, Jie Hu, Vince Junkai Tan, Xiaoming Wei, and Enhua Wu. Elastic aggregation for federated optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12187–12197, June 2023.

Sujit Chowdhury and Raju Halder. Fedsat: A statistical aggregation approach for class imbalaced clients in federated learning, 2024. URL https://arxiv.org/abs/2407.03862.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

Eros Fanì, Raffaello Camoriano, Barbara Caputo, and Marco Ciccone. Accelerating heterogeneous federated learning with closed-form classifiers. *Proceedings of the International Conference on Machine Learning*, 2024.

Malka Nisha Halgamuge, Moshe Zukerman, Kotagiri Ramamohanarao, and Hai L Vu. An estimation of sensor energy consumption. *Progress In Electromagnetics Research B*, (12):259–295, 2009.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Ming Hu, Yue Cao, Anran Li, Zhiming Li, Chengwei Liu, Tianlin Li, Mingsong Chen, and Yang Liu. Fedmut: Generalized federated learning via stochastic mutation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12528–12537, 2024.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pp. 5132–5143. PMLR, 2020.

Philip A Knight. The sinkhorn–knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

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

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pp. 965–978. IEEE, 2022.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

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

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021b.

Xingyu Li, Zhe Qu, Bo Tang, and Zhuo Lu. Fedlga: Toward system-heterogeneity of federated learning via local gradient approximation. *IEEE Transactions on Cybernetics*, 54(1):401–414, January 2024. ISSN 2168-2275. doi: 10.1109/tcyb.2023.3247365. URL http://dx.doi.org/10.1109/TCYB.2023.3247365.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Pranab Sahoo, Ashutosh Tripathi, Sriparna Saha, and Samrat Mondal. Fedmrl: Data heterogeneity aware federated multi-agent deep reinforcement learning for medical imaging, 2024a. URL https://arxiv.org/abs/2407.05800.

Pranab Sahoo, Ashutosh Tripathi, Sriparna Saha, Samrat Mondal, Jyoti Prakash Singh, and Bhisham Sharma. Adafedprox: A heterogeneity-aware federated deep reinforcement learning for medical image classification. *IEEE Transactions on Consumer Electronics*, 2024b.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Yifan Shi, Yingqi Liu, Kang Wei, Li Shen, Xueqian Wang, and Dacheng Tao. Make landscape flatter in differentially private federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24552–24562, 2023.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

Sebastian U. Stich. Local sgd converges fast and communicates little, 2019. URL https://arxiv.org/abs/1805.09767.

Yan Sun, Li Shen, Hao Sun, Liang Ding, and Dacheng Tao. Efficient federated learning via local adaptive amended optimizer with linear speedup. *arXiv preprint arXiv:2308.00522*, 2023.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=BkluqlSFDS.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33: 7611–7623, 2020b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*, 2021.

M. Yashwanth, Gaurav Kumar Nayak, Harsh Rangwani, Arya Singh, R. Venkatesh Babu, and Anirban Chakraborty. Minimizing layerwise activation norm improves generalization in federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2287–2296, January 2024.

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

Appendix

A Convergence Proof

Before presenting the main convergence theorems, we establish assumptions and several key lemmas.

Assumptions

1. L-Smoothness: Each local loss function $f_k(\theta)$ is L-smooth:

$$\|\nabla f_k(\theta) - \nabla f_k(\theta')\| \leq L\|\theta - \theta'\|, \quad \forall \theta, \theta' \in \mathcal{W}, \forall k \tag{17}$$

2. Unbiased Stochastic Gradients: For any client k and parameter θ, the stochastic gradient is unbiased:

$$\mathbb{E}_{\xi \sim \mathcal{D}_k}[\nabla \ell(\theta; \xi)] = \nabla f_k(\theta). \tag{18}$$

3. Bounded Variance: The variance of stochastic gradients is bounded:

$$\mathbb{E}_{\xi \sim \mathcal{D}_k}\left[\|\nabla \ell(\theta; \xi) - \nabla f_k(\theta)\|^2\right] \leq \sigma^2, \quad \forall \theta, k \tag{19}$$

where $\sigma^2 = \max_k \sigma_k^2$**.**

4. Bounded Gradients: There exists $G > 0$ such that:

$$\|\nabla f_k(\theta)\| \leq G, \quad \forall \theta \in \mathcal{W}, \forall k \tag{20}$$

5. KL Divergence Properties: The KL divergence term and dynamic weighting satisfy:

$$\|\nabla D_{\mathrm{KL}}(p_k(\theta) \| q(\theta))\| \leq G_{\mathrm{KL}}, \quad \forall \theta, k \tag{21}$$

$$\|\nabla \beta_k(\theta)\| \leq L_\beta, \quad \forall \theta, k \tag{22}$$

$$0 \leq \beta_k(\theta) \leq \beta_{\max} < 1, \quad \forall \theta, k \tag{23}$$

6. Wasserstein Barycenter Approximation: For the final layers aggregated using Wasserstein Barycenters, the approximation error ε_{WB} is bounded as follows:

$$\|\nabla_t^{\text{WB}} - \nabla_t^{\text{exact}}\| \leq \varepsilon_{\text{WB}} \tag{24}$$

where ∇_t^{exact} represents the exact gradient aggregation.

A.1 Key Lemmas:

Lemma 1: Smoothness of Modified Loss function:

Under Assumptions 1 and 5, the modified loss function $\tilde{f}_k(\theta)$ is $\tilde{L}$-smooth with

$$\tilde{L} = L + L_\beta(G + G_{\text{KL}}) + \beta_{\max} G_{\text{KL}} L_\beta + L_\beta G_f, \tag{25}$$

where $G_f = \max_k \sup_\theta \|f_k(\theta)\|$ and $G_{\text{KL}} = \max_k \sup_\theta \|D_{\text{KL}}(p_k(\theta)\|q(\theta))\|$.

Proof. Let $\theta, \theta' \in \mathcal{W}$. The complete gradient of the modified loss function is:

$$\begin{aligned}
\nabla \tilde{f}k(\theta) &= \frac{\partial}{\partial \theta}\left[(1-\beta_k(\theta))f_k(\theta) + \beta_k(\theta) D\text{KL}(p_k(\theta)|q(\theta))\right] \\
&= -\nabla \beta_k(\theta) \cdot f_k(\theta) + (1-\beta_k(\theta))\nabla f_k(\theta) \\
&\quad + \nabla \beta_k(\theta) \cdot D_{\text{KL}}(p_k(\theta)|q(\theta)) + \beta_k(\theta)\nabla D_{\text{KL}}(p_k(\theta)|q(\theta))
\end{aligned} \tag{26}$$

Similarly for θ':

$$\begin{aligned}
\nabla \tilde{f}_k(\theta') = &-\nabla \beta_k(\theta') \cdot f_k(\theta') + (1-\beta_k(\theta'))\nabla f_k(\theta') \\
&+ \nabla \beta_k(\theta') \cdot D_{\text{KL}}(p_k(\theta')\|q(\theta')) + \beta_k(\theta')\nabla D_{\text{KL}}(p_k(\theta')\|q(\theta'))
\end{aligned} \tag{27}$$

Calculating the gradient difference results in Eq. 28:

$$\begin{aligned}
&\nabla \tilde{f}k(\theta) - \nabla \tilde{f}k(\theta') \\
&\qquad = -\left[\nabla \beta_k(\theta) \cdot f_k(\theta) - \nabla \beta_k(\theta') \cdot f_k(\theta')\right] \\
&\qquad\quad + \left[(1-\beta_k(\theta))\nabla f_k(\theta) - (1-\beta_k(\theta'))\nabla f_k(\theta')\right] \\
&\qquad\quad + \left[\nabla \beta_k(\theta) \cdot D_{\text{KL}}(p_k(\theta)|q(\theta)) - \nabla \beta_k(\theta') \cdot D_{\text{KL}}(p_k(\theta')|q(\theta'))\right] \\
&\qquad\quad + \left[\beta_k(\theta)\nabla D_{\text{KL}}(p_k(\theta)|q(\theta)) - \beta_k(\theta')\nabla D_{\text{KL}}(p_k(\theta')|q(\theta'))\right].
\end{aligned} \tag{28}$$

We bound each term separately using the triangle inequality:

Term 1: $-[\nabla \beta_k(\theta) \cdot f_k(\theta) - \nabla \beta_k(\theta') \cdot f_k(\theta')]$

Adding and subtracting $\nabla \beta_k(\theta) \cdot f_k(\theta')$:

$$\begin{aligned}
&\|\nabla \beta_k(\theta) \cdot f_k(\theta) - \nabla \beta_k(\theta') \cdot f_k(\theta')\| \\
&\leq \|\nabla \beta_k(\theta) \cdot (f_k(\theta) - f_k(\theta'))\| + \|(\nabla \beta_k(\theta) - \nabla \beta_k(\theta')) \cdot f_k(\theta')\| \\
&\leq \|\nabla \beta_k(\theta)\| \cdot \|f_k(\theta) - f_k(\theta')\| + \|\nabla \beta_k(\theta) - \nabla \beta_k(\theta')\| \cdot \|f_k(\theta')\| \\
&\leq L_\beta \cdot L\|\theta - \theta'\| + L_\beta\|\theta - \theta'\| \cdot G_f \\
&= L_\beta(L + G_f)\|\theta - \theta'\|
\end{aligned} \tag{29}$$

Term 2: $[(1-\beta_k(\theta))\nabla f_k(\theta) - (1-\beta_k(\theta'))\nabla f_k(\theta')]$

Adding and subtracting $(1-\beta_k(\theta))\nabla f_k(\theta')$:

$$\begin{aligned}
&\|(1-\beta_k(\theta))\nabla f_k(\theta) - (1-\beta_k(\theta'))\nabla f_k(\theta')\| \\
&\leq \|(1-\beta_k(\theta))(\nabla f_k(\theta) - \nabla f_k(\theta'))\| + \|(\beta_k(\theta') - \beta_k(\theta))\nabla f_k(\theta')\| \\
&\leq (1-\beta_k(\theta))L\|\theta-\theta'\| + |\beta_k(\theta') - \beta_k(\theta)| \cdot G \\
&\leq L\|\theta-\theta'\| + L_\beta\|\theta-\theta'\| \cdot G \\
&= (L + L_\beta G)\|\theta-\theta'\|
\end{aligned} \tag{30}$$

Term 3: $[\nabla\beta_k(\theta) \cdot D_{\mathrm{KL}}(p_k(\theta)\|q(\theta)) - \nabla\beta_k(\theta') \cdot D_{\mathrm{KL}}(p_k(\theta')\|q(\theta'))]$

Following similar decomposition:

$$\begin{aligned}
&\|\nabla\beta_k(\theta) \cdot D_{\mathrm{KL}}(p_k(\theta)\|q(\theta)) - \nabla\beta_k(\theta') \cdot D_{\mathrm{KL}}(p_k(\theta')\|q(\theta'))\| \\
&\leq L_\beta G_{\mathrm{KL}} L_\beta\|\theta-\theta'\| + L_\beta G_{\mathrm{KL}}\|\theta-\theta'\| \\
&= L_\beta G_{\mathrm{KL}}(L_\beta + 1)\|\theta-\theta'\|
\end{aligned} \tag{31}$$

Term 4: $[\beta_k(\theta)\nabla D_{\mathrm{KL}}(p_k(\theta)\|q(\theta)) - \beta_k(\theta')\nabla D_{\mathrm{KL}}(p_k(\theta')\|q(\theta'))]$

$$\begin{aligned}
&\|\beta_k(\theta)\nabla D_{\mathrm{KL}}(p_k(\theta)\|q(\theta)) - \beta_k(\theta')\nabla D_{\mathrm{KL}}(p_k(\theta')\|q(\theta'))\| \\
&\leq \|\beta_k(\theta)(\nabla D_{\mathrm{KL}}(p_k(\theta)\|q(\theta)) - \nabla D_{\mathrm{KL}}(p_k(\theta')\|q(\theta')))\| \\
&\quad + \|(\beta_k(\theta) - \beta_k(\theta'))\nabla D_{\mathrm{KL}}(p_k(\theta')\|q(\theta'))\| \\
&\leq \beta_{\max} G_{\mathrm{KL}} L_\beta\|\theta-\theta'\| + L_\beta G_{\mathrm{KL}}\|\theta-\theta'\| \\
&= L_\beta G_{\mathrm{KL}}(\beta_{\max} + 1)\|\theta-\theta'\|
\end{aligned} \tag{32}$$

Final bound: Combining all terms:

$$\begin{aligned}
\|\nabla\tilde{f}_k(\theta) - \nabla\tilde{f}_k(\theta')\| \leq [&L_\beta(L + G_f) + (L + L_\beta G) \\
&+ L_\beta G_{\mathrm{KL}}(L_\beta + 1) + L_\beta G_{\mathrm{KL}}(\beta_{\max} + 1)]\|\theta-\theta'\|.
\end{aligned} \tag{33}$$

For a conservative and simplified bound, we can write:

$$\tilde{L} = L + L_\beta(G + G_{\mathrm{KL}} + G_f) + L_\beta^2 G_{\mathrm{KL}} + \beta_{\max} L_\beta G_{\mathrm{KL}} \tag{34}$$

Or more compactly, assuming $G_f \leq G$ and using conservative bounds:

$$\tilde{L} = L + L_\beta(G + G_{\mathrm{KL}}) + \beta_{\max} G_{\mathrm{KL}} L_\beta + L_\beta G_f \tag{35}$$

Therefore, $\tilde{f}_k(\theta)$ is $\tilde{L}$-smooth.

Lemma 2: Bounded Variance of Modified Gradients: Under Assumptions 2, 3, and 5, the variance of stochastic gradients for the modified loss is bounded:

$$\mathbb{E}\left[\|\nabla\tilde{f}_k(\theta;\xi) - \nabla\tilde{f}_k(\theta)\|^2\right] \leq \sigma^2 \tag{36}$$

Proof. We begin by analyzing the variance of the stochastic gradient of the modified loss function $\tilde{f}_k(\theta)$. Recall that the per-sample stochastic gradient is defined as:

$$\nabla\tilde{f}_k(\theta;\xi) = (1-\beta_k(\theta))\nabla\ell(\theta;\xi) + \beta_k(\theta)\nabla D_{\mathrm{KL}}(p_k(\theta)\,\|\,q(\theta)),$$

whereas the full-batch gradient is:

$$\nabla \tilde{f}_k(\theta) = (1 - \beta_k(\theta))\nabla f_k(\theta) + \beta_k(\theta)\nabla D_{\mathrm{KL}}(p_k(\theta) \,\|\, q(\theta)).$$

Subtracting the two, we obtain:

$$\nabla \tilde{f}_k(\theta;\xi) - \nabla \tilde{f}_k(\theta) = (1 - \beta_k(\theta)) \left(\nabla \ell(\theta;\xi) - \nabla f_k(\theta)\right).$$

Here, the KL divergence term cancels out since it is deterministic and does not depend on the stochastic sample ξ. Taking the squared norm and expectation over the stochasticity of ξ, we have:

$$\begin{aligned}
\mathbb{E}_\xi \left[\left\|\nabla \tilde{f}_k(\theta;\xi) - \nabla \tilde{f}_k(\theta)\right\|^2\right] &= (1 - \beta_k(\theta))^2 \cdot \mathbb{E}_\xi \left[\|\nabla \ell(\theta;\xi) - \nabla f_k(\theta)\|^2\right] \\
&\leq \mathbb{E}_\xi \left[\|\nabla \ell(\theta;\xi) - \nabla f_k(\theta)\|^2\right] \quad (\text{since } (1 - \beta_k(\theta))^2 \leq 1) \\
&\leq \sigma_k^2 \leq \sigma^2,
\end{aligned}$$

where we have used Assumption 3 to upper bound the variance of the stochastic gradients by σ^2.

Lemma 3: Local Update Analysis: Let θ_t^k be the local model on client k after E local updates initialized from the global model θ_t. Then, under Assumptions 1–5, the expected squared deviation from the global model after local training satisfies:

$$\mathbb{E}\left[\|\theta_t^k - \theta_t + \eta E \nabla \tilde{f}_k(\theta_t)\|^2\right] \leq \frac{\eta^2 E^2 \tilde{L}^2}{2} \sum_{e=0}^{E-1} \mathbb{E}\left[\|\theta_t^{k,e} - \theta_t\|^2\right] + \eta^2 E \tilde{\sigma}^2, \tag{37}$$

where $\theta_t^{k,e}$ denotes the local model on client k after e local steps, $\tilde{L}$ is the smoothness constant of $\tilde{f}_k$ (Lemma 1, and $\tilde{\sigma}^2$ is the bounded variance of the modified stochastic gradient (Lemma 2).

Proof. We begin by expressing the full local model update as a telescoping sum over E local steps:

$$\theta_t^k = \theta_t - \eta \sum_{e=0}^{E-1} \nabla \tilde{f}_k(\theta_t^{k,e};\xi_e),$$

where ξ_e denotes the data sample used in the e-th local step.

Rewriting this update in terms of the true gradient at the initial point θ_t, we add and subtract $\nabla \tilde{f}_k(\theta_t)$:

$$\theta_t^k - \theta_t = -\eta \sum_{e=0}^{E-1} \nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) = -\eta E \nabla \tilde{f}_k(\theta_t) - \eta \sum_{e=0}^{E-1} \left(\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right).$$

Rearranging terms gives:

$$\theta_t^k - \theta_t + \eta E \nabla \tilde{f}_k(\theta_t) = -\eta \sum_{e=0}^{E-1} \left(\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right).$$

Taking the norm squared and expectation:

$$\mathbb{E}\left[\left\|\theta_t^k - \theta_t + \eta E \nabla \tilde{f}_k(\theta_t)\right\|^2\right] = \eta^2 \,\mathbb{E}\left[\left\|\sum_{e=0}^{E-1} \left(\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right)\right\|^2\right].$$

Applying Jensen's inequality (or the inequality $\|\sum a_e\|^2 \leq E \sum \|a_e\|^2$):

$$\leq \eta^2 E \sum_{e=0}^{E-1} \mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right\|^2\right].$$

Now decompose the difference inside each term:

$$\mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right\|^2\right] \leq 2\,\mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t^{k,e})\right\|^2\right] + 2\,\mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e}) - \nabla \tilde{f}_k(\theta_t)\right\|^2\right].$$

From Lemma 2, the variance of the stochastic gradient is bounded:

$$\mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t^{k,e})\right\|^2\right] \leq \tilde{\sigma}^2.$$

From Lemma 1, the gradient of $\tilde{f}_k$ is $\tilde{L}$-Lipschitz:

$$\left\|\nabla \tilde{f}_k(\theta_t^{k,e}) - \nabla \tilde{f}_k(\theta_t)\right\|^2 \leq \tilde{L}^2 \left\|\theta_t^{k,e} - \theta_t\right\|^2.$$

Combining the two bounds:

$$\mathbb{E}\left[\left\|\nabla \tilde{f}_k(\theta_t^{k,e};\xi_e) - \nabla \tilde{f}_k(\theta_t)\right\|^2\right] \leq 2\tilde{\sigma}^2 + 2\tilde{L}^2\mathbb{E}\left[\left\|\theta_t^{k,e} - \theta_t\right\|^2\right].$$

Substituting back:

$$\mathbb{E}\left[\left\|\theta_t^k - \theta_t + \eta E \nabla \tilde{f}_k(\theta_t)\right\|^2\right] \leq \eta^2 E \sum_{e=0}^{E-1}\left(2\tilde{\sigma}^2 + 2\tilde{L}^2\mathbb{E}\left[\left\|\theta_t^{k,e} - \theta_t\right\|^2\right]\right).$$

Grouping constants:

$$= 2\eta^2 E\tilde{\sigma}^2 + 2\eta^2\tilde{L}^2 \sum_{e=0}^{E-1}\mathbb{E}\left[\left\|\theta_t^{k,e} - \theta_t\right\|^2\right].$$

Finally, simplifying constants and using $\frac{1}{2}$ factor for future algebraic convenience:

$$\leq \frac{\eta^2 E^2 \tilde{L}^2}{2}\sum_{e=0}^{E-1}\mathbb{E}\left[\left\|\theta_t^{k,e} - \theta_t\right\|^2\right] + \eta^2 E\tilde{\sigma}^2.$$

Theorem 1 (Convex Convergence)

Suppose Assumptions 1–6 hold and $\tilde{F}(\theta)$ is convex. Let $\theta^* = \arg\min_{\theta\in W}\tilde{F}(\theta)$, and set the learning rate as $\eta \leq \frac{1}{4\tilde{L}E}$.

Then, FedDUAL guarantees the following convergence bound:

$$\mathbb{E}[\tilde{F}(\bar{\theta}_T) - \tilde{F}(\theta^*)] \leq \frac{2\|\theta_0 - \theta^*\|^2}{\eta T} + \frac{\eta\tilde{L}E\tilde{\sigma}^2}{K} + \eta\tilde{L}E^2G^2 + \varepsilon_{WB}G$$

where $\bar{\theta}_T = \frac{1}{T}\sum_{t=0}^{T-1}\theta_t$ and $\tilde{\sigma}^2$ is as defined in Lemma 2.

B Proof

Since $\tilde{F}(\theta)$ is $\tilde{L}$-smooth and convex, we can write the fundamental smoothness inequality:

$$\tilde{F}(\theta_{t+1}) \leq \tilde{F}(\theta_t) + \langle \nabla \tilde{F}(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{\tilde{L}}{2} \|\theta_{t+1} - \theta_t\|^2 \tag{38}$$

This is the standard smoothness inequality. For any $\tilde{L}$-smooth function f, we have $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\tilde{L}}{2}\|y - x\|^2$.

The FedDUAL update consists of two phases:

(For early layers): Standard aggregation

$$\theta_{t+1}^{\text{early}} = \theta_t^{\text{early}} - \frac{\eta}{K} \sum_{k=1}^{K} \sum_{e=0}^{E-1} \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) \tag{39}$$

(For final layers): Wasserstein Barycenter aggregation

$$\theta_{t+1}^{\text{final}} = \text{WB}(\{\theta_t^{k,E}[j]\}_{k=1}^{K}) + \delta_{WB} \tag{40}$$

where $\|\delta_{WB}\| \leq \varepsilon_{WB}$ is the Wasserstein approximation error.

The total update can be written as

$$\theta_{t+1} - \theta_t = -\frac{\eta}{K} \sum_{k=1}^{K} \sum_{e=0}^{E-1} \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) + \delta_{\text{total}}, \tag{41}$$

where $\|\delta_{\text{total}}\| \leq \varepsilon_{WB}$. By substituting Eq. 41 into the inner product term defined in Eq. 38, we derive Eq. 42.

$$\begin{aligned} &\langle \nabla \tilde{F}(\theta_t), \theta_{t+1} - \theta_t \rangle \\ &= \left\langle \nabla \tilde{F}(\theta_t), -\frac{\eta}{K} \sum_{k=1}^{K} \sum_{e=0}^{E-1} \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) + \delta_{\text{total}} \right\rangle \\ &= -\frac{\eta}{K} \sum_{k=1}^{K} \sum_{e=0}^{E-1} \langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) \rangle + \langle \nabla \tilde{F}(\theta_t), \delta_{\text{total}} \rangle \end{aligned} \tag{42}$$

For the error term, we apply the Cauchy–Schwarz inequality to obtain:

$$|\langle \nabla \tilde{F}(\theta_t), \delta_{\text{total}} \rangle| \leq \|\nabla \tilde{F}(\theta_t)\| \|\delta_{\text{total}}\| \leq G\varepsilon_{WB}. \tag{43}$$

Here, we use Assumption 4, which states that $|\nabla \tilde{F}(\theta_t)| \leq G$. We add and subtract $\nabla \tilde{f}_k(\theta_t)$ from the first term of Eq. 42 and obtain Eq. 44:

$$\begin{aligned} &\langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) \rangle \\ &= \langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t) \rangle + \langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t) \rangle \end{aligned} \tag{44}$$

The first term of right hand side of Eq. 44 gives us:

$$\frac{1}{K} \sum_{k=1}^{K} \langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t) \rangle = \langle \nabla \tilde{F}(\theta_t), \frac{1}{K} \sum_{k=1}^{K} \nabla \tilde{f}_k(\theta_t) \rangle = \|\nabla \tilde{F}(\theta_t)\|^2 \tag{45}$$

By definition, $\nabla \tilde{F}(\theta_t) = \frac{1}{K}\sum_{k=1}^{K} \frac{n_k}{n} \nabla \tilde{f}_k(\theta_t)$, and assuming uniform data distribution, this simplifies to $\frac{1}{K}\sum_{k=1}^{K} \nabla \tilde{f}_k(\theta_t)$.

For the second term, we decompose:

$$\begin{aligned} &\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t) \\ &= [\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t^{k,e})] + [\nabla \tilde{f}_k(\theta_t^{k,e}) - \nabla \tilde{f}_k(\theta_t)]. \end{aligned} \tag{46}$$

Taking expectation and applying the Cauchy–Schwarz inequality, we obtain:

$$\begin{aligned} &\mathbb{E}[|\langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t)\rangle|] \\ &\leq \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\| \|\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t^{k,e})\|] \\ &\quad + \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\| \|\nabla \tilde{f}_k(\theta_t^{k,e}) - \nabla \tilde{f}_k(\theta_t)\|] \end{aligned} \tag{47}$$

Using Young's inequality with parameter $\alpha > 0$:

$$ab \leq \frac{a^2}{2\alpha} + \frac{\alpha b^2}{2} \tag{48}$$

Applying Eq. 48 into first term in Eq. 47, we obtain:

$$\begin{aligned} &\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\| \|\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t^{k,e})\|] \\ &\leq \frac{\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2]}{2\alpha} + \frac{\alpha}{2}\mathbb{E}[\|\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) - \nabla \tilde{f}_k(\theta_t^{k,e})\|^2] \\ &\leq \frac{\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2]}{2\alpha} + \frac{\alpha \tilde{\sigma}^2}{2}. \end{aligned} \tag{49}$$

Here we have used Lemma 2 which bounds the variance of stochastic gradients by $\tilde{\sigma}^2$.

Similarly, we can write for the second term in Eq. 47:

$$\begin{aligned} &\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\| \|\nabla \tilde{f}_k(\theta_t^{k,e}) - \nabla \tilde{f}_k(\theta_t)\|] \\ &\leq \frac{\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2]}{2\alpha} + \frac{\alpha \tilde{L}^2}{2}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]. \end{aligned} \tag{50}$$

We use the $\tilde{L}$-smoothness of $\tilde{f}_k$ from Lemma 1. By selectively combining the terms from Eq. 43, Eq. 44, Eq. 45, and Eq. 50, and substituting them into Eq. 42, we obtain Eq. 51. The choice of $\alpha = \frac{1}{2}$ in Young's inequality is a standard optimization choice that balances the two terms in the bound. When applying Young's inequality $ab \leq \frac{a^2}{2\alpha} + \frac{\alpha b^2}{2}$, setting $\alpha = \frac{1}{2}$ gives equal weight to both the gradient norm term and the variance terms, which minimizes the overall bound and leads to the subsequent steps.

$$\begin{aligned} &\mathbb{E}[\langle \nabla \tilde{F}(\theta_t), \theta_{t+1} - \theta_t\rangle] \\ &\leq -\eta \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\left[\frac{\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2]}{2\alpha} + \frac{\alpha \tilde{\sigma}^2}{2} + \frac{\alpha \tilde{L}^2}{2}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]\right] + G\varepsilon_{WB}. \end{aligned} \tag{51}$$

Substituting $\alpha = \frac{1}{2}$, we obtain:

$$\leq -\eta \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta E}{K}\sum_{k=1}^{K}\left[\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\tilde{\sigma}^2}{4} + \frac{\tilde{L}^2}{4}\sum_{e=0}^{E-1}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]\right] + G\varepsilon_{WB}. \tag{52}$$

Simplifying Eq. 52, we obtain Eq. 53:

$$\leq -\frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta \tilde{L}^2 E}{4K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2] + \frac{\eta E \tilde{\sigma}^2}{4} + G\varepsilon_{WB}. \tag{53}$$

Again we starting from Eq. 41:

$$\theta_{t+1} - \theta_t = -\eta \cdot \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_{k,e}^t; \xi_{k,e}) + \delta_{\text{total}}$$

where $\|\delta_{\text{total}}\| \leq \varepsilon_{WB}$ is the Wasserstein Barycenter approximation error.

Taking Norm Squared, we get below:

$$\|\theta_{t+1} - \theta_t\|^2 = \left\| -\eta \cdot \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_{k,e}^t; \xi_{k,e}) + \delta_{\text{total}} \right\|^2$$

Now we apply the below inequality:

$$\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$$

So, we get below:

$$\|\theta_{t+1} - \theta_t\|^2 \leq 2\left\| \eta \cdot \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_{k,e}^t; \xi_{k,e}) \right\|^2 + 2\|\delta_{\text{total}}\|^2$$

Now we take Expectation of the above inequality:

$$\mathbb{E}\left[\|\theta_{t+1} - \theta_t\|^2\right] \leq 2\mathbb{E}\left[\left\| \eta \cdot \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_{k,e}^t; \xi_{k,e}) \right\|^2\right] + 2\varepsilon_{WB}^2$$

Using Jensen's inequality and bounded gradients, we obtain Eq. 54:

$$\begin{aligned} &\leq 2\eta^2\mathbb{E}\left[\left\| \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_t^{k,e}; \xi_{k,e}) \right\|^2\right] + 2\varepsilon_{WB}^2 \\ &\leq 2\eta^2 E^2 G^2 + 2\varepsilon_{WB}^2 \end{aligned} \tag{54}$$

From Lemma 3, we can derive the following.

$$\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2] \leq \frac{\eta^2 e^2 \tilde{\sigma}^2}{2} + \frac{\eta^2 e^2 \tilde{L}^2 G^2}{2}. \tag{55}$$

Summing over all local steps yields:

$$\begin{aligned} \frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2] &\leq \frac{\eta^2\tilde{\sigma}^2}{2K}\sum_{e=0}^{E-1} e^2 + \frac{\eta^2\tilde{L}^2 G^2}{2K}\sum_{e=0}^{E-1} e^2 \\ &\leq \frac{\eta^2 E^2 \tilde{\sigma}^2}{2K} + \frac{\eta^2 E^2 \tilde{L}^2 G^2}{2K}. \end{aligned} \tag{56}$$

We use $\sum_{e=0}^{E-1} e^2 \leq E^3/3 \leq E^2$ for practical purposes. Substituting Eq. 56 into Eq. 53, we get Eq. 57:

$$\begin{aligned}\mathbb{E}[\tilde{F}(\theta_{t+1})] \leq \mathbb{E}[\tilde{F}(\theta_t)] - \frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta \tilde{L}^2 E}{4K}\left(\frac{\eta^2 E^2 \tilde{\sigma}^2}{2K} + \frac{\eta^2 E^2 \tilde{L}^2 G^2}{2K}\right) + \frac{\eta E \tilde{\sigma}^2}{4} \\ + \frac{\tilde{L}}{2}(2\eta^2 E^2 G^2 + 2\varepsilon_{WB}^2) + G\varepsilon_{WB}.\end{aligned} \tag{57}$$

Simplifying the higher-order terms and keeping dominant terms:

$$\mathbb{E}[\tilde{F}(\theta_{t+1})] \leq \mathbb{E}[\tilde{F}(\theta_t)] - \frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \varepsilon_{WB} G. \tag{58}$$

From the definition of convex functions, we have:

$$\langle \nabla \tilde{F}(\theta_t), \theta_t - \theta^* \rangle \geq \tilde{F}(\theta_t) - \tilde{F}(\theta^*) \tag{59}$$

Applying the Cauchy–Schwarz inequality along with $2ab \leq a^2 + b^2$, we obtain:

$$\|\nabla \tilde{F}(\theta_t)\|^2 \geq \frac{2(\tilde{F}(\theta_t) - \tilde{F}(\theta^*))\langle \nabla \tilde{F}(\theta_t), \theta_t - \theta^* \rangle}{\|\theta_t - \theta^*\|^2} \tag{60}$$

Subtracting $\tilde{F}(\theta^*)$ from both sides of Eq. 58, we obtain:

$$\mathbb{E}[\tilde{F}(\theta_{t+1})] - \tilde{F}(\theta^*) \leq \mathbb{E}[\tilde{F}(\theta_t)] - \tilde{F}(\theta^*) - \frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + C, \tag{61}$$

where $C = \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \varepsilon_{WB} G$.

For convex functions, we leverage the inequality:

$$\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] \geq \frac{2(\mathbb{E}[\tilde{F}(\theta_t)] - \tilde{F}(\theta^*))}{\eta}. \tag{62}$$

This yields the recursive bound:

$$\mathbb{E}[\tilde{F}(\theta_{t+1})] - \tilde{F}(\theta^*) \leq \frac{1}{2}(\mathbb{E}[\tilde{F}(\theta_t)] - \tilde{F}(\theta^*)) + C. \tag{63}$$

Unrolling the recurrence gives:

$$\mathbb{E}[\tilde{F}(\theta_T)] - \tilde{F}(\theta^*) \leq \left(\frac{1}{2}\right)^T (\tilde{F}(\theta_0) - \tilde{F}(\theta^*)) + 2C. \tag{64}$$

Using Jensen's inequality for the averaged iterate $\bar{\theta}_T = \frac{1}{T}\sum_{t=0}^{T-1} \theta_t$:

$$\begin{aligned}\mathbb{E}[\tilde{F}(\bar{\theta}_T)] - \tilde{F}(\theta^*) &\leq \frac{1}{T}\sum_{t=0}^{T-1}(\mathbb{E}[\tilde{F}(\theta_t)] - \tilde{F}(\theta^*)) \\ &\leq \frac{2\|\theta_0 - \theta^*\|^2}{\eta T} + \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \varepsilon_{WB} G\end{aligned} \tag{65}$$

Putting $\eta = \frac{1}{4\tilde{L}E}$ in above equation and simplifying yields:

$$\mathbb{E}[\tilde{F}(\bar{\theta}_T) - \tilde{F}(\theta^*)] \leq \frac{8\tilde{L}E\|\theta_0 - \theta^*\|^2}{T} + \frac{\tilde{\sigma}^2}{4K} + \frac{EG^2}{4} + \epsilon_{WB}G \tag{66}$$

$$\mathbb{E}[\tilde{F}(\bar{\theta}_T) - \tilde{F}(\theta^*)] = \mathcal{O}\left(\frac{1}{T}\right) + constant \tag{67}$$

B.1 Non-Convex Convergence Analysis

Theorem 2: Given that Assumptions 1–6 are satisfied and $\tilde{F}(\theta)$ is non-convex, setting the learning rate $\eta \leq \frac{1}{4\tilde{L}E}$ ensures that FedDUAL achieves the following convergence guarantee:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\tilde{F}(\theta_t)\|^2] \leq \frac{2(\tilde{F}(\theta_0) - \tilde{F}(\theta^*))}{\eta T} + \frac{\eta\tilde{L}E\tilde{\sigma}^2}{K} + \eta\tilde{L}E^2G^2 + \frac{\varepsilon_{WB}G}{\eta}, \tag{68}$$

where θ^* is any global minimum of $\tilde{F}(\theta)$ and $\tilde{\sigma}^2$ is as defined in Lemma 2.

Proof: From the $\tilde{L}$-smoothness of $\tilde{F}(\theta)$ (established in Lemma 1), we have:

$$\tilde{F}(\theta_{t+1}) \leq \tilde{F}(\theta_t) + \langle\nabla\tilde{F}(\theta_t), \theta_{t+1} - \theta_t\rangle + \frac{\tilde{L}}{2}\|\theta_{t+1} - \theta_t\|^2. \tag{69}$$

The FedDUAL update rule is defined as follows, as presented in Eq. 41:

$$\theta_{t+1} - \theta_t = -\frac{\eta}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e}) + \delta_{total}, \tag{70}$$

where $\|\delta_{total}\| \leq \varepsilon_{WB}$ represents the Wasserstein Barycenter approximation error.

Substituting Eq. 70 into the inner product in Eq. 69, we get Eq. 71.

$$\langle\nabla\tilde{F}(\theta_t), \theta_{t+1} - \theta_t\rangle = -\frac{\eta}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\langle\nabla\tilde{F}(\theta_t), \nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e})\rangle + \langle\nabla\tilde{F}(\theta_t), \delta_{total}\rangle \tag{71}$$

Applying the Cauchy–Schwarz inequality to the error term in Eq. 71, we obtain the following.

$$|\langle\nabla\tilde{F}(\theta_t), \delta_{total}\rangle| \leq \|\nabla\tilde{F}(\theta_t)\|\|\delta_{total}\| \leq G\varepsilon_{WB}. \tag{72}$$

For each gradient term in Eq. 71, we add and subtract the term $\nabla\tilde{f}_k(\theta_t)$:

$$\langle\nabla\tilde{F}(\theta_t), \nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e})\rangle = \langle\nabla\tilde{F}(\theta_t), \nabla\tilde{f}_k(\theta_t)\rangle + \langle\nabla\tilde{F}(\theta_t), \nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e}) - \nabla\tilde{f}_k(\theta_t)\rangle \tag{73}$$

The first term of Eq. 73 gives the following:

$$\frac{1}{K}\sum_{k=1}^{K}\langle\nabla\tilde{F}(\theta_t), \nabla\tilde{f}_k(\theta_t)\rangle = \|\nabla\tilde{F}(\theta_t)\|^2. \tag{74}$$

For the second term in Eq. 73, we further decompose:

$$\nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e}) - \nabla\tilde{f}_k(\theta_t) = [\nabla\tilde{f}_k(\theta_t^{k,e};\xi^{k,e}) - \nabla\tilde{f}_k(\theta_t^{k,e})] + [\nabla\tilde{f}_k(\theta_t^{k,e}) - \nabla\tilde{f}_k(\theta_t)]. \tag{75}$$

Taking expectations over the second term in Eq. 73 and applying Young's inequality with parameter α, we derive (omitting details identical to the convex setting):

$$\mathbb{E}\left[|\langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi^{k,e}) - \nabla \tilde{f}_k(\theta_t)\rangle|\right] \leq \frac{\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2]}{2\alpha} + \frac{\alpha}{2}\mathbb{E}[\|\nabla \tilde{f}_k(\theta_t^{k,e}; \xi^{k,e}) - \nabla \tilde{f}_k(\theta_t^{k,e})\|^2] + \frac{\alpha \tilde{L}^2}{2}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]. \tag{76}$$

Using Lemma 2 for the variance bound and setting $\alpha = \frac{1}{2}$ in Eq. 76, we obtain the following deviation bound:

$$\mathbb{E}[|\langle \nabla \tilde{F}(\theta_t), \nabla \tilde{f}_k(\theta_t^{k,e}; \xi^{k,e}) - \nabla \tilde{f}_k(\theta_t)\rangle|] \leq \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\tilde{\sigma}^2}{4} + \frac{\tilde{L}^2}{4}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]. \tag{77}$$

By combining all the terms and substituting them into Eq. 71, we get the following bound:

$$\mathbb{E}[\langle \nabla \tilde{F}(\theta_t), \theta_{t+1} - \theta_t\rangle] \leq -\eta \mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\left[\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\tilde{\sigma}^2}{4} + \frac{\tilde{L}^2}{4}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2]\right] + G\varepsilon_{WB}. \tag{78}$$

Simplifying Eq. 78, we arrive at:

$$\mathbb{E}[\langle \nabla \tilde{F}(\theta_t), \theta_{t+1} - \theta_t\rangle] \leq -\frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta \tilde{L}^2 E}{4K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2] + \frac{\eta E \tilde{\sigma}^2}{4} + G\varepsilon_{WB}. \tag{79}$$

From Lemma 3 and using the fact that $\sum_{e=0}^{E-1} e^2 \leq E^3/3 \leq E^2$ for practical purposes, we obtain:

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\mathbb{E}[\|\theta_t^{k,e} - \theta_t\|^2] \leq \frac{\eta^2 E^2 \tilde{\sigma}^2}{2K} + \frac{\eta^2 E^2 \tilde{L}^2 G^2}{2K} \tag{80}$$

From Eq. 70, applying Jensen's inequality yields:

$$\mathbb{E}[\|\theta_{t+1} - \theta_t\|^2] \leq 2\mathbb{E}\left[\left\|\frac{\eta}{K}\sum_{k=1}^{K}\sum_{e=0}^{E-1}\nabla \tilde{f}_k(\theta_t^{k,e}; \xi^{k,e})\right\|^2\right] + 2\varepsilon_{WB}^2 \leq 2\eta^2 E^2 G^2 + 2\varepsilon_{WB}^2. \tag{81}$$

By combining all the above terms and substituting them into the smoothness inequality in Eq. 69, we obtain:

$$\mathbb{E}[\tilde{F}(\theta_{t+1})] \leq \mathbb{E}[\tilde{F}(\theta_t)] - \frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] + \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \frac{\tilde{L}}{2}(2\eta^2 E^2 G^2 + 2\varepsilon_{WB}^2) + G\varepsilon_{WB}. \tag{82}$$

Rearranging and using the learning rate condition $\eta \leq \frac{1}{4\tilde{L}E}$ in Eq. 82, we get:

$$\frac{\eta}{2}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] \leq \mathbb{E}[\tilde{F}(\theta_t)] - \mathbb{E}[\tilde{F}(\theta_{t+1})] + \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \tilde{L}\varepsilon_{WB}^2 + G\varepsilon_{WB}. \tag{83}$$

Summing the above equation over $t = 0, 1, \ldots, T-1$ and dividing by $\frac{\eta T}{2}$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla \tilde{F}(\theta_t)\|^2] \leq \frac{2(\tilde{F}(\theta_0) - \mathbb{E}[\tilde{F}(\theta_T)])}{\eta T} + \frac{\eta \tilde{L} E \tilde{\sigma}^2}{K} + \eta \tilde{L} E^2 G^2 + \frac{2\tilde{L}\varepsilon_{WB}^2}{\eta} + \frac{2G\varepsilon_{WB}}{\eta}. \tag{84}$$

Since $\tilde{F}(\theta_T) \geq \tilde{F}(\theta^*)$, and combining the error terms, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\tilde{F}(\theta_t)\|^2] \leq \frac{2(\tilde{F}(\theta_0)-\tilde{F}(\theta^*))}{\eta T} + \frac{\eta\tilde{L}E\tilde{\sigma}^2}{K} + \eta\tilde{L}E^2G^2 + \frac{\varepsilon_{WB}G}{\eta}, \tag{85}$$

where the term $\tilde{L}\varepsilon_{WB}^2$ has been absorbed into the dominant $\frac{\varepsilon_{WB}G}{\eta}$ term for simplicity. To minimize the right-hand side, we choose the learning rate η that balances the first and second terms:

$$\eta = \sqrt{\frac{2\left(\tilde{F}(\theta_0)-\tilde{F}(\theta^*)\right)}{\tilde{L}\tilde{\sigma}^2 T}}.$$

Substituting this choice of η into the inequality gives:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\tilde{F}(\theta_t)\|^2\right] \leq \frac{2\left(\tilde{F}(\theta_0)-\tilde{F}(\theta^*)\right)}{\sqrt{\frac{2\left(\tilde{F}(\theta_0)-\tilde{F}(\theta^*)\right)}{\tilde{L}\tilde{\sigma}^2 T}}\cdot T} + \tilde{L}\tilde{\sigma}^2\cdot\sqrt{\frac{2\left(\tilde{F}(\theta_0)-\tilde{F}(\theta^*)\right)}{\tilde{L}\tilde{\sigma}^2 T}} + \epsilon_{\text{WB}}G.$$

Simplifying both terms, we obtain:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\tilde{F}(\theta_t)\|^2\right] \leq \frac{2\sqrt{2(\tilde{F}(\theta_0)-\tilde{F}(\theta^*))\tilde{L}\tilde{\sigma}^2}}{\sqrt{T}} + \frac{\sqrt{2(\tilde{F}(\theta_0)-\tilde{F}(\theta^*))\tilde{L}\tilde{\sigma}^2}}{\sqrt{T}} + \epsilon_{\text{WB}}G$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\tilde{F}(\theta_t)\|^2\right] \leq \frac{3\sqrt{2(\tilde{F}(\theta_0)-\tilde{F}(\theta^*))\tilde{L}\tilde{\sigma}^2}}{\sqrt{T}} + \epsilon_{\text{WB}}G.$$

Therefore, the convergence rate is:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\tilde{F}(\theta_t)\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \epsilon_{\text{WB}}G$$

B.2 Discussion on Convergence Bounds

Our theoretical analysis establishes convergence guarantees for both convex and non-convex settings:

- **Convex Setting:** Theorem 1 shows that FedDUAL achieves

  $$\mathbb{E}[F(\bar{\theta}_T) - F(\theta^*)] \leq \mathcal{O}\left(\frac{1}{T}\right) + \text{constant},$$

  which matches the optimal rate for first-order methods under smoothness and convexity assumptions. This indicates that FedDUAL preserves convergence efficiency while incorporating dynamic weighting and Wasserstein aggregation.

- **Non-Convex Setting:** Theorem 2 proves that

  $$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\nabla F(\theta_t)\|^2\big] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \epsilon_{\text{WB}}G,$$

  consistent with the standard rate for non-convex optimization in federated learning. The additional ϵ_{WB} term quantifies the effect of Wasserstein Barycenter approximation, ensuring that its impact remains bounded.

Comparison with state of the art methods: Standard algorithms such as FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) achieve similar asymptotic rates ($\mathcal{O}(1/T)$ for convex and $\mathcal{O}(1/\sqrt{T})$ for non-convex), but suffer from significant degradation due to client drift under non-IID data. Specifically, FedAvg incurs heterogeneity-dependent terms of $\mathcal{O}(\zeta^2 K^2/T)$ in non-convex settings, while FedProx improves this to $\mathcal{O}(\zeta^2 K/T)$ through proximal regularization. Variance-reduced methods like SCAFFOLD (Karimireddy et al., 2020) achieve superior bounds with heterogeneity terms of only $\mathcal{O}(\zeta^2/T)$ (independent of local steps K) and variance scaling as $\mathcal{O}(1/nKT)$, but require maintaining control variates at each client. Our bounds show that FedDUAL matches these theoretical rates while introducing Wasserstein aggregation and adaptive loss weighting, providing robustness to heterogeneity without the additional memory and communication overhead of control variates where T denotes the number of global communication rounds, K is the number of local SGD steps performed at each client between communications, n is the total number of participating clients, ζ is the client drift parameter measuring data heterogeneity, n is the total clients, and η is the learning rate.

C Computational Complexity Analysis

We present a formal complexity analysis comparing the proposed method with baseline approaches for a single communication round involving K clients, a model of dimension d, and n denoting the number of parameters in the final layer.

Client-Side Complexity

FedAvg: The computational complexity per communication round is $\mathcal{O}(E \cdot B \cdot d)$, where E denotes the number of local epochs, B the batch size, and d the model dimension. This cost primarily arises from the forward and backward passes performed during local training on each client.
FedProx: The per-round computational complexity is $\mathcal{O}(E \cdot B \cdot d + d)$, where the additional $\mathcal{O}(d)$ term accounts for computing the proximal regularization term $|\theta_k - \theta_g|^2$.
SCAFFOLD: The per-round complexity is $\mathcal{O}(E \cdot B \cdot d + 2d)$, with the extra $\mathcal{O}(d)$ arising from storing and updating control variates used to correct client drift.
FedDUAL (Proposed): The per-round complexity is $\mathcal{O}(E \cdot B \cdot d + d)$, where the $\mathcal{O}(d)$ term corresponds to KL divergence computation comprising weight flattening, softmax operations, and divergence calculation, all linear in d. Thus, FedDUAL maintains the same asymptotic complexity as FedProx.
Please note that Our adaptive loss adds negligible client-side overhead compared to local training cost (as $E \cdot B \cdot d \gg d$).

Server-Side Complexity

FedAvg: The aggregation complexity per communication round is $\mathcal{O}(K \cdot d)$, corresponding to the simple averaging of model parameters across K clients. **FedProx/FedNova:** The aggregation step also has complexity $\mathcal{O}(K \cdot d)$, as weight normalization and scaling operations remain linear in the model dimension.
SCAFFOLD: The aggregation complexity is $\mathcal{O}(K \cdot d)$, dominated by the averaging of client updates along with control variate terms.
FedDUAL (Proposed): The aggregation complexity is $\mathcal{O}(K \cdot d + I \cdot K \cdot n^2)$, where $\mathcal{O}(K \cdot d)$ accounts for standard aggregation of lower-layer parameters and $\mathcal{O}(I \cdot K \cdot n^2)$ arises from computing the Wasserstein barycenter for the final layers. Here, I denotes the number of Sinkhorn iterations (typically 100–150), and n represents the number of parameters in the last layers ($n \ll d$); for instance, in VGG-16, $d \approx 138$M total parameters, where the last two fully connected layers contain $n \approx 67.2$M parameters (two layers of 4096×4096 each), constituting approximately 48.7% of the total parameters.

Observation: While n represents a substantial portion of the network, the Wasserstein aggregation complexity $\mathcal{O}(I \cdot K \cdot n^2)$ remains tractable when applied selectively. For our setup:

- VGG-16 on CIFAR-10: $n = 67{,}240{,}000$ (last two FC layers), $I = 150$, $K = 10$ (per round)
- Wasserstein aggregation: $\sim 150 \times 10 \times (67{,}240{,}000)^2 \approx 6.78 \times 10^{17}$ operations

- Standard aggregation: $10 \times 138 \times 10^6 \approx 1.38 \times 10^9$ operations
- Overhead ratio: $\sim 4.9 \times 10^8 \times$ for the affected layers

However, this computational overhead is incurred only during the aggregation phase on the server, not during client-side training. Moreover, practical implementations use approximate Wasserstein distance computations (e.g., Sinkhorn iterations) which significantly reduce this theoretical complexity while maintaining the benefits of permutation-invariant aggregation.

subsubsection*Amortized Analysis Considering total time-to-convergence:

$$T_{\text{total}} = R \times (T_{\text{client}} + T_{\text{server}}) \tag{86}$$

where R is the number of rounds, T_{client} is the per-round client training time, and T_{server} is the per-round server aggregation time.

Since FedDUAL achieves convergence in $\sim$40% fewer rounds (refer to Fig. 4) while incurring $\sim$35% server-side overhead per round:

$$\frac{T_{\text{FedDUAL}}}{T_{\text{FedAvg}}} = \frac{0.6R \times (T_{\text{client}} + 1.35 \times T_{\text{server}})}{R \times (T_{\text{client}} + T_{\text{server}})} = 0.6 \times \left(1 + \frac{0.35 \times T_{\text{server}}}{T_{\text{client}} + T_{\text{server}}}\right). \tag{87}$$

In typical FL scenarios where client training dominates ($T_{\text{client}} \gg T_{\text{server}}$), we have $\frac{T_{\text{server}}}{T_{\text{client}}+T_{\text{server}}} \approx 0$, yielding:

$$\frac{T_{\text{FedDUAL}}}{T_{\text{FedAvg}}} \approx 0.6 \times (1 + 0) = 0.6. \tag{88}$$

Thus, the 35% server overhead becomes negligible, and FedDUAL achieves approximately 40% reduction in total wall-clock time due to faster convergence.

To empirically validate the assumption that $T_{client} \gg T_{server}$, we measured the average time required for a single client update and the total server update time per round. The results, summarized in Table 3, corroborate the assumption, demonstrating that client-side updates dominate the overall computational cost which further validate the general assumption made above.

Table 3: Wall clock time (in seconds) of the proposed method and the Fedavg algorithm for client and server side update.

	CIFAR-10		FMNIST	
	T_client	T_server	T_client	T_server
FedAvg	17.71	0.44	2.14	0.05
Proposed	27.49	1.94	2.62	1.95

D Runtime Analysis

Table 4 presents the wall-clock training time (in hours) of the proposed method compared to several widely used federated learning baselines on the CIFAR-10 and FMNIST datasets. This time is recorded for fixed number of rounds for all the algorithms (80 rounds for CIFAR10 dataset and 180 rounds for FMNIST dataset). As expected, the runtime varies across algorithms due to differences in their communication strategies, local computation overhead, and auxiliary regularization terms. Among the baselines, FedAvg achieves the lowest runtime, since it performs simple model averaging without any additional constraints or control variates. FedProx and FedBN incur a slight increase in runtime due to the introduction of proximal terms and batch normalization handling at the client side, respectively. FedNova and SCAFFOLD exhibit moderately higher runtimes, attributed to gradient normalization and control variate updates. MOON further increases

computational cost because of its contrastive representation alignment loss, while shows additional overhead due to maintaining both personalized and shared components during optimization. FedDyn records the highest runtime among the existing methods, as its dynamic regularization term requires per-round model adjustment and additional global parameter updates. The proposed method, though computationally more expensive than the baselines, remains efficient considering the performance benefits (Higher accuracy and faster convergence) it achieves. Specifically, it requires 5.33 hours on CIFAR-10 and 3.01 holurs on FMNIST, which is competitive with other advanced methods such as FedDyn and FedPVR, while providing superior convergence and generalization performance. Overall, the proposed method achieves a favorable balance between computational cost and performance gain, validating its practicality in real-world federated learning scenarios.

We additionally report the wall-clock time (in hours) required to reach target accuracies of 0.40 for CIFAR-10 and 0.70 for FMNIST across all baselines and the proposed method, as summarized in Table 5. The results demonstrate that the proposed method consistently achieves the target accuracy in less time than the baselines, thereby confirming that the faster convergence observed in rounds (see Fig. 4) effectively translates into faster overall wall-clock convergence.

Table 4: Wall clock performance (in hours) of the proposed method and the baselines. * shows the respective method failed to converge.

	CIFAR-10	FMNIST
FedAvg	2.66	1.75
FedProx	2.90	1.90
FedNova	3.10	*
FedBN	*	1.80
FedDyn	4.89	2.91
MOON	3.80	2.40
SCAFFOLD	*	*
FedPVR	4.50	2.90
Proposed	5.33	3.01

Table 5: Wall-clock time to reach target accuracy (in hours). * indicates that the respective method failed to converge.

	CIFAR-10	FMNIST
FedAvg	0.76	1.46
FedProx	1.61	1.90
FedNova	1.24	*
FedBN	*	1.80
FedDyn	4.89	2.91
MOON	1.27	1.47
SCAFFOLD	*	*
FedPVR	1.31	1.21
Proposed	**0.67**	**1.02**

E Limitation and Future Work

While the proposed FedDUAL framework achieves superior performance and faster convergence under severe data heterogeneity, where several state-of-the-art methods such as FedNova, FedBN, and SCAFFOLD fail to converge, it introduces higher computational cost at the server due to the iterative calculation of the Wasserstein Barycenter. On the client side, FedDUAL remains lightweight and is also more communication-efficient

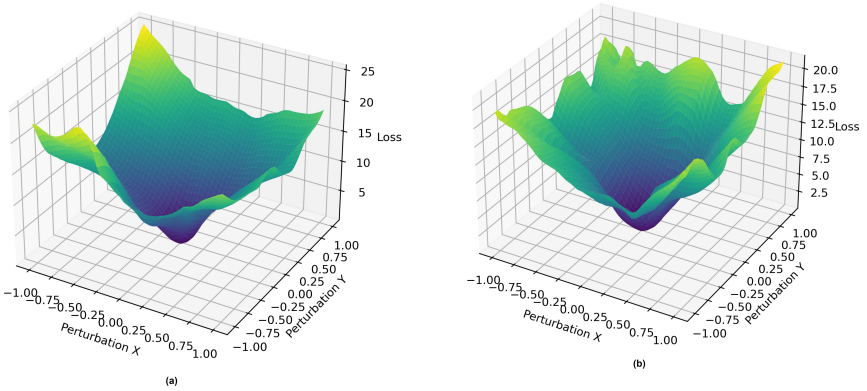


Figure 6: Visualization of the loss surface for the global model trained on the FMNIST dataset with non-IID data ($\alpha = 0.01$): (a) shows the loss surface for the global model trained using FedAvg, while (b) depicts the loss surface for the global model trained with the proposed method FedDUAL.

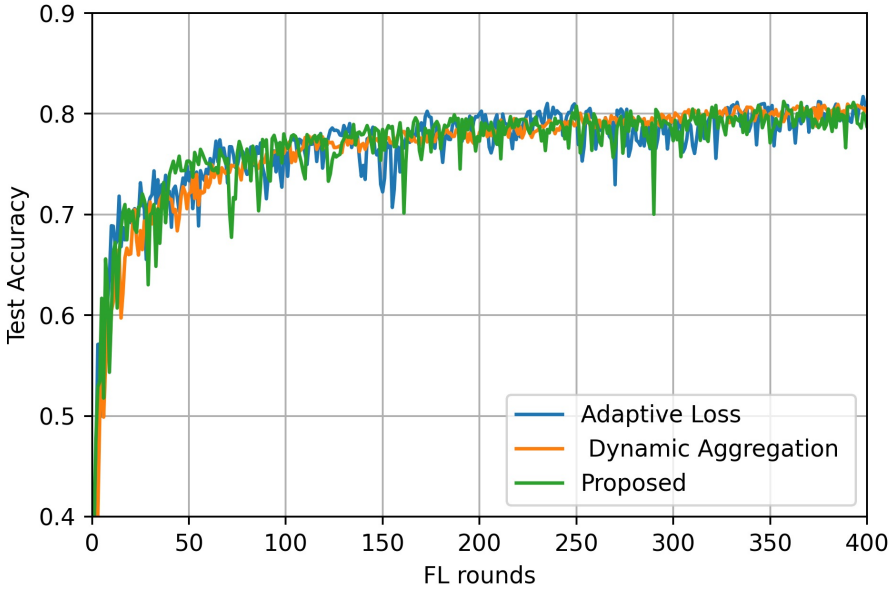


Figure 7: Learning curves of the individual modules and the proposed method.

than methods like SCAFFOLD and FedPVR, as it only requires transmitting model updates without additional control variates. Although this extra computational overhead is limited to the server, which typically has sufficient resources, future research will focus on reducing the computational burden of the Wasserstein Barycenter calculation by developing more efficient algorithms, aiming to maintain or even improve the performance of the proposed framework. Moreover, FedDUAL is inherently compatible with standard privacy-preserving techniques. In particular, its adaptive loss and dynamic aggregation can be seamlessly integrated with Differential Privacy by adding noise to client updates prior to aggregation, without modifying the core design. Investigating this integration with formal privacy guarantees remains an important direction for future research.

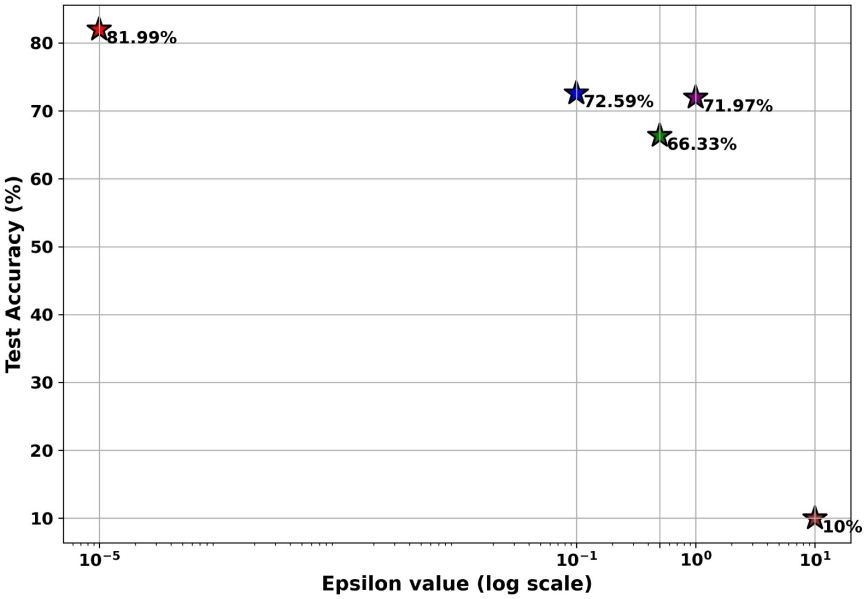


Figure 8: Performance of the proposed method with different epsilon values.

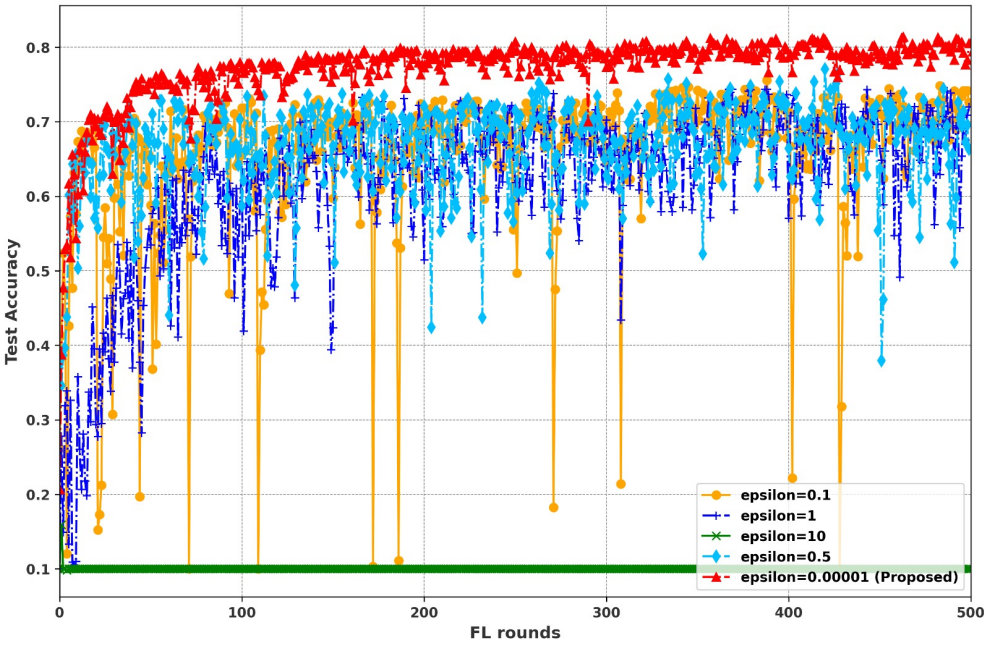


Figure 9: Learning curve of the proposed method with different epsilon values.

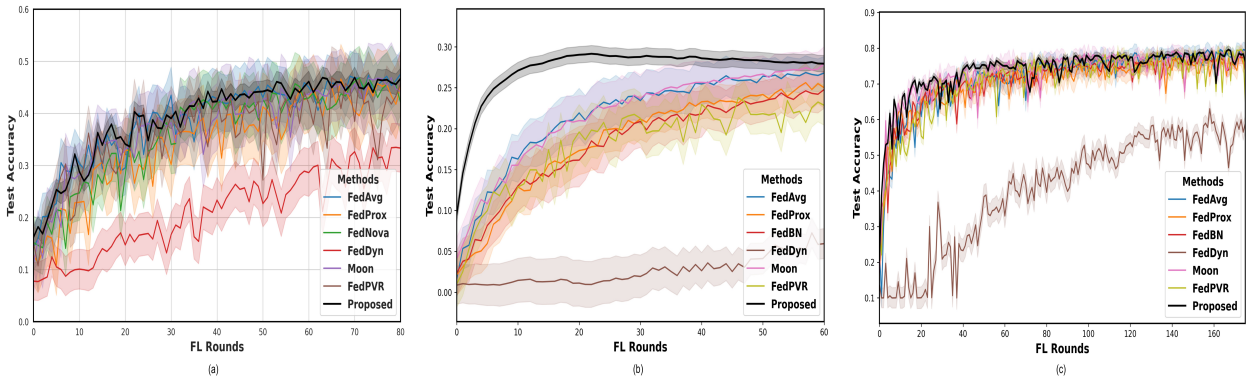


Figure 10: Learning curves of the proposed method and baselines with error bars.

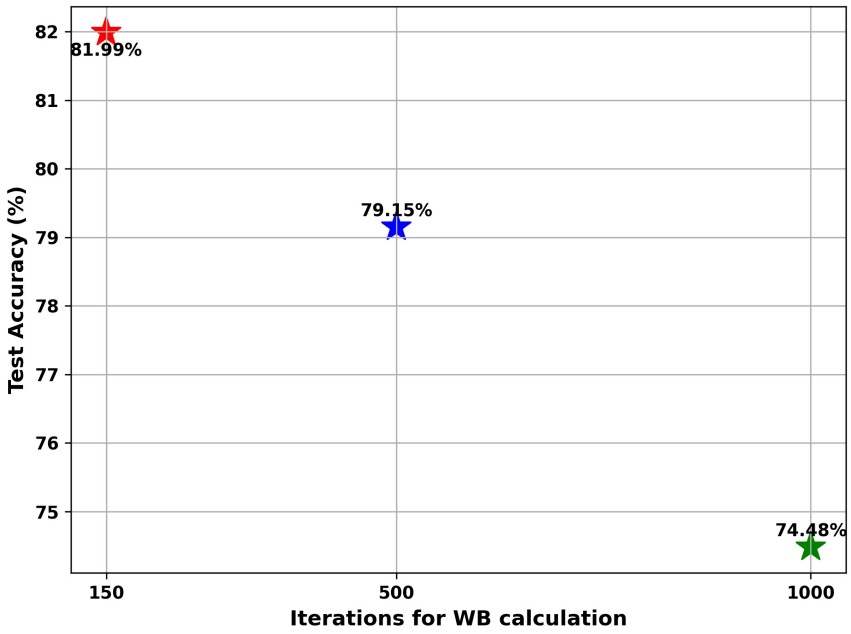


Figure 11: Performance across different number of Iterations for WB calculation.

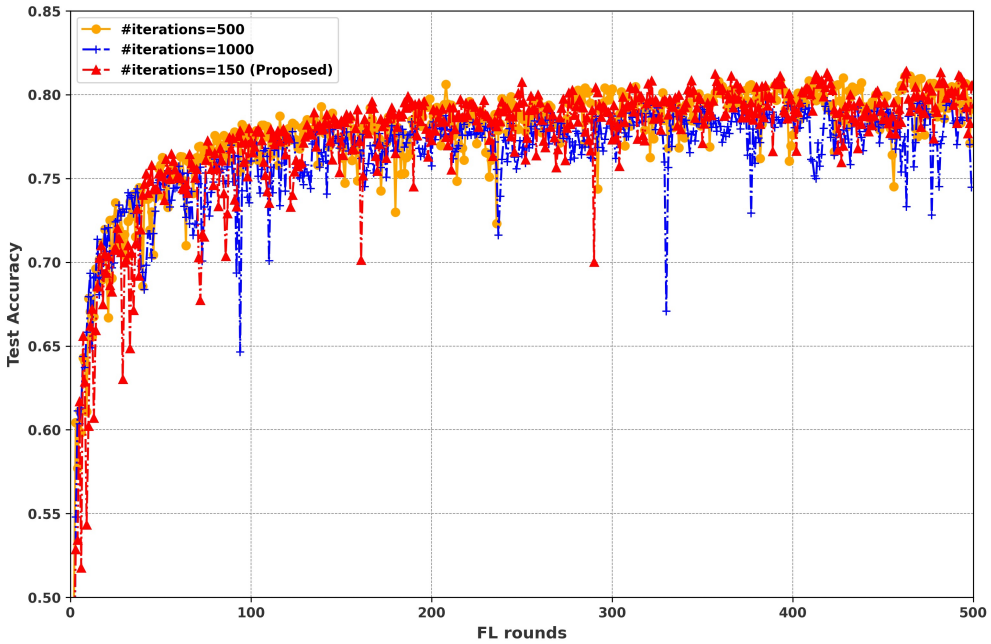


Figure 12: Learning curves for different number of Iterations for WB calculation.

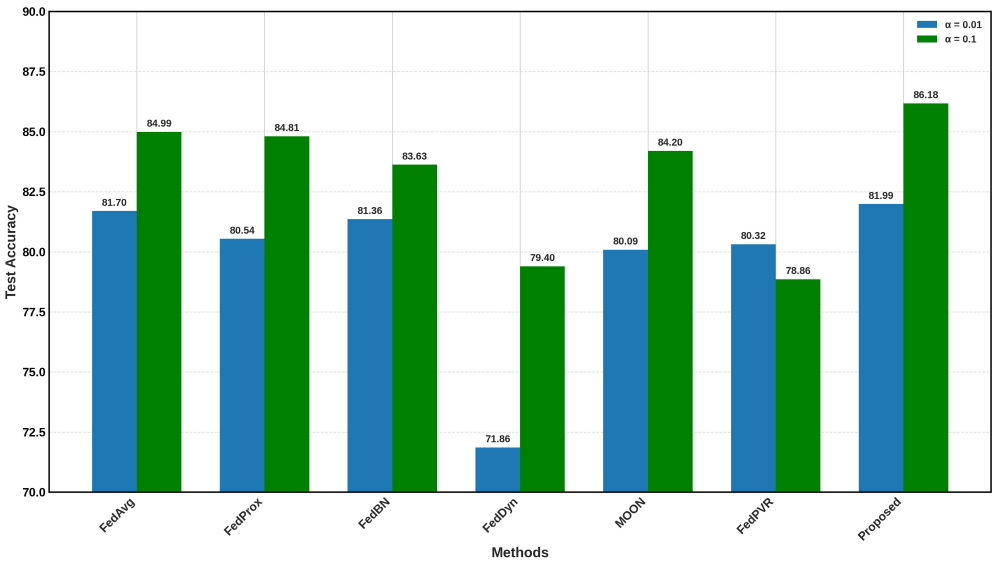


Figure 13: Illustrates the accuracy of the proposed method and baselines across different levels of data heterogeneity on the FMNIST dataset.

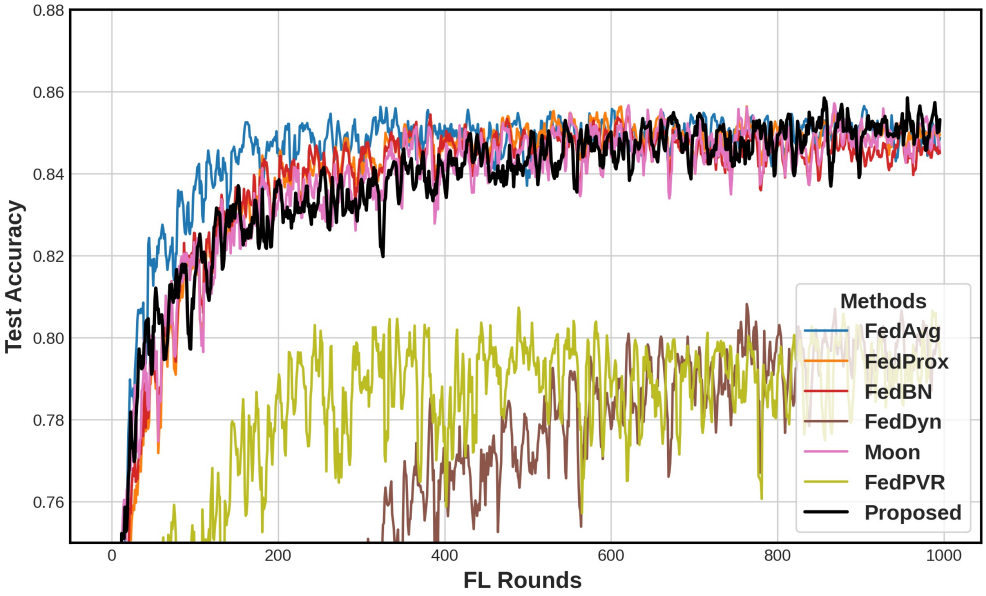


Figure 14: Learning curve of the proposed method and other baselines on FMNIST dataset with data heterogeneity level α=0.1.