Harnessing Heterogeneity: Improving Convergence Through Partial Variance Control in Federated Learning

Anonymous authors
Paper under double-blind review

Abstract

Federated Learning (FL) has emerged as a promising paradigm for collaborative model training without sharing local data. However, a significant challenge in FL arises from the heterogeneous data distributions across participating clients. This heterogeneity leads to highly variable gradient norms in the model's final layers, resulting in poor generalization, slower convergence, and reduced robustness of the global model. To address these issues, we propose a novel technique that incorporates a gradient penalty term into partial variance control. Our method enables diverse representation learning from heterogeneous client data in the initial layers while modifying standard SGD in the final layers. This approach reduces variance in the classification layers, aligns gradients, and mitigates the effects of data heterogeneity on image classification tasks. Through theoretical analysis, we establish convergence rate bounds for the proposed algorithm, demonstrating its potential for competitive convergence compared to current FL methods in highly heterogeneous data settings. Empirical evaluations on three computer vision image classification datasets validate our approach, showing enhanced performance and faster convergence over state-of-the-art baselines across various levels of data heterogeneity. Our code is available at `https://anonymous.4open.science/r/FedPGVC-7F18`.

1 Introduction

Federated learning (FL) facilitates collaborative training of a global model across multiple clients while preserving data privacy by avoiding the need to transmit local data to a central server, in contrast to traditional centralized methods McMahan et al. (2017). With the proliferation of decentralized data sources like mobile devices, hospitals, and the Internet of Things (IoT), FL has gained traction as a solution for training deep networks across distributed environments Zhang et al. (2022). However, a significant practical obstacle encountered during federated training is data heterogeneity across clients Kairouz et al. (2021); Li et al. (2020). Diverse user behaviors can lead to significant heterogeneity in the local data across different clients, resulting in non-independent and identically distributed (non-IID) data. This heterogeneity has been shown to cause unstable convergence, slow training progress, and ultimately suboptimal or even detrimental model performance Li et al. (2022); Zhao et al. (2018). While FedAvg McMahan et al. (2017) has been widely adopted and successful across multiple applications, it frequently encounters challenges in attaining optimal accuracy and convergence, particularly in heterogeneous data distributions. This difficulty arises from client drifts Karimireddy et al. (2020), a phenomenon resulting from the varying nature of data among participating clients. Prior research has addressed the issue of client drift by introducing penalties for the divergence between client and server model Li et al. (2020; 2021a), or by employing variance reduction approaches during the client model update Karimireddy et al. (2020); Acar et al. (2021). Luo et al. (2021) tackled data heterogeneity through classifier re-training utilizing virtual features. Another study uncovers that a biased classifier significantly undermines the performance of federated training on heterogeneous data and introduces a novel algorithm by re-training the classifier with learnable features Shang et al. (2022). A recent study by Li et al. (2023) measures gradient variability across clients by calculating drift diversity, especially in deeper layers, and proposes aligning classification layers using control variates. While this approach may enhance model performance, it can increase communication costs and relies on assumptions that may not always hold in practical FL scenarios.

1.1 Empirical Observations

Based on these observations, we conducted two experiments: first, to empirically analyze gradient norms for models trained on IID and non-IID data, aiming to understand gradient behavior across layers and its impact on training stability; second, to determine which parts of the neural network are more sensitive to data heterogeneity. We have taken the CIFAR100 dataset with $\alpha = 0.5$ for non-IID data and $\alpha = 100$ to make IID data distribution and a CNN with 10 layers for both experiments. The detailed experimental setting can be found in Section 4.1. Initially, we calculate aggregate metrics, including the mean, variance, and maximum gradient norms for all layers. The Mean Gradient Norm is expressed as follows:

$$\text{Mean}(\|\nabla W\|) = \frac{1}{L}\sum_{i=1}^{L}\|\nabla W_i\| \tag{1}$$

where L is the total number of layers, and $\|\nabla W_i\|$ is the gradient norm of the i^{th} layer, defined as:

$$\|\nabla W_i\| = \sqrt{\sum_{j=1}^{n}(\nabla W_{i,j})^2} \tag{2}$$

where n is the number of weights in each layer. The Variance of the Gradient Norm is calculated as:

$$\text{Var}(\|\nabla W\|) = \frac{1}{L}\sum_{i=1}^{L}(\|\nabla W_i\| - \text{Mean}(\|\nabla W\|))^2 \tag{3}$$

The Maximum Gradient Norm is defined as:

$$\text{Max}(\|\nabla W\|) = \max_{i=1,\ldots,L}\|\nabla W_i\| \tag{4}$$

We computed these metrics for each layer of the CNN model trained with both IID and non-IID data distributions. The results, presented in Table 1, reveal that models trained on non-IID data exhibit higher average gradient norms and greater variance compared to those trained on IID data. This indicates that non-IID training leads to larger updates and potentially greater instability in the training process. In the second experiment, we analyzed the gradient norms for each layer of the CNN model to understand the impact of distribution shifts. The results, shown in Fig. 1, illustrate the variation of gradient norms across layers for both IID and non-IID cases. Initial layers exhibit higher gradient norms, which decrease significantly in subsequent layers. Both models display similar gradient patterns in the initial layers. However, the model trained on non-IID data exhibits higher gradient norms in and near the classification layer, indicating larger updates and greater instability in these regions due to data heterogeneity. These findings highlight that the classification layer, along with its neighboring layers, significantly contributes to the observed instability and slower convergence when training with non-IID data.

Table 1: Aggregate metrics for gradient norms of models trained on IID and non-IID data distribution.

Metric	IID Model	non-IID Model
Mean Gradient Norm	1.96×10^3	7.49×10^3
Variance of Gradient Norm	5.40×10^6	2.82×10^7
Maximum Gradient Norm	7.90×10^3	1.83×10^4

Inspired by the above empirical observations, we propose Federated Partial Gradient Variance Control (FedPGVC) to stabilize noisy gradient norms to mitigate data heterogeneity without incurring additional communication costs. FedPGVC calculates a gradient penalty term for each individual client, updating the

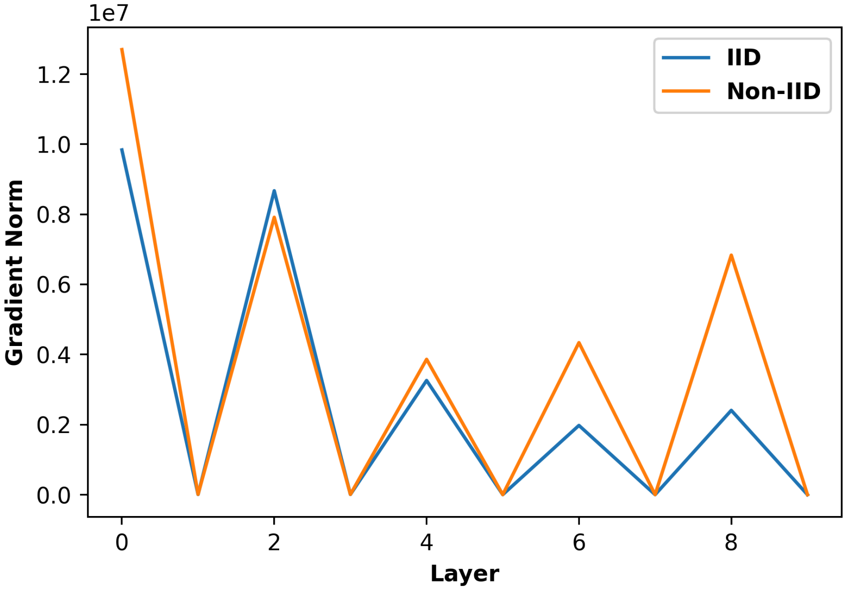


Figure 1: Comparison of gradient norm of the two models (IID and non-IID) trained using FedAvg.

last layers of the neural network while the remaining layers are updated using a Stochastic Gradient Descent (SGD) optimizer. In this work, we calculate the gradient penalty term inspired by the Wasserstein Distributionally Robust Optimization (WDRO) Gao & Kleywegt (2023). This approach addresses distributional uncertainties and deviations, enhancing the global model's resilience to non-IID data across clients and improving generalization to unseen data samples, even when they deviate from the training distribution. We have performed experiments on the three widely used datasets, MNIST, FMNIST, and CIFAR100 datasets, with varying degrees of data heterogeneity among clients. Our experimental results demonstrate that the proposed FedPGVC requires fewer communication rounds to achieve the same level of accuracy as existing approaches. Furthermore, with a fixed number of communication rounds, FedPGVC attains comparable or superior top-1 accuracy. The key contributions of this work are as follows:

- We introduce FedPGVC to tackle the challenges of data heterogeneity in federated training by incorporating a partial variance reduction technique utilizing client-specific gradient penalty terms.
- We have proposed a gradient penalty term for the weight updates of the final classification layers to mitigate client drift, stabilize gradient diversity, and accelerate convergence in federated training.
- We offer a theoretical convergence for FedPGVC in both convex and non-convex scenarios, outlining its limited reliance on measures of data heterogeneity.
- Experimental analysis shows that the proposed FedPGVC surpasses prior state-of-the-art methods in both performance and convergence efficiency across different levels of data heterogeneity and a range of diverse datasets.

2 Related work

Numerous studies have explored effective strategies for addressing the challenges of data heterogeneity in FL. We have broadly categorized these approaches into three groups: 1) client drift mitigation, which adjusts the local objectives of clients to align their models more closely with the global model; 2) aggregation schemes, which enhance the server-side fusion mechanism for model updates; 3) personalized federated learning, which focuses on training personalized models for clients rather than a shared global model. In the proposed work, we mainly focused on techniques based on client drift mitigation.

FedAvg is a predominant optimization method in FL and has witnessed widespread adoption McMahan et al. (2017). However, in heterogeneous settings where local objectives diverge significantly, FedAvg encounters performance degradation due to client drift, limiting its effectiveness in non-IID data scenarios Karimireddy et al. (2020). Li et al. (2020) introduced a proximal regularization term to manage the divergence between client and server models but fails to align global and local optimal points effectively. Li et al. (2021b) employs local batch normalization (LBN) to mitigate feature shift before server-side model averaging. Sahoo et al. (2024) introduces a novel loss function and an innovative way of calculating adaptive proximal term to tackle heterogeneous data settings. Additionally, it uses Self-organizing map (SOM) based server-side aggregation. Several techniques like FedBabu Oh et al. (2021) and TCT Yu et al. (2022) aim to enhance FL models by fine-tuning classifiers using standalone datasets or simulated features derived from client models. Similarly, Luo et al. (2021) addresses data heterogeneity by re-training classifiers with virtual features obtained from an approximated GMM model. MOON introduces a model-contrastive FL framework that aligns local client representations with the global model using a contrastive loss Li et al. (2021a). It employs a momentum encoder to provide a stable target for the contrastive loss, acting as a temporal ensemble of the global model and mitigating client drift. Stochastic variance reduction based methods like SVRG Johnson & Zhang (2013), SAGA Defazio et al. (2014), and their variations utilize control variates to mitigate the variance inherent in traditional SGD, enabling linear convergence rates for strongly convex optimization problems. SCAFFOLD Karimireddy et al. (2020) and DANE Shamir et al. (2014) have incorporated variance reduction techniques for the whole model on convex problems without exploring their performance in non-convex setups. Despite potential benefits, these approaches incur higher communication costs due to transmitting additional control variates, posing challenges for resource-constrained IoT devices Halgamuge et al. (2009). Additionally, the existing methods have shown rapid convergence in simpler models, and their effectiveness on deep networks He et al. (2015); Huang et al. (2018), remains largely unexplored. FedPVR Li et al. (2023) offers a novel perspective on FedAvg's performance in deep neural networks, uncovering substantial heterogeneity in client-specific final classification layers. By introducing targeted variance reduction exclusively for these last layers, FedPVR achieves remarkable improvements over established benchmarks. Motivated by the previous observations, our research focuses on improving the partial variance control of individual clients to mitigate data heterogeneity problems.

3 Method

3.1 Problem Statement

The primary aim of this work is to develop a robust model that can learn collaboratively from decentralized clients without the need for data sharing. The focus is on enhancing performance during federated training, particularly in scenarios with non-IID data. Given K clients, where each client $k \in \{1, \dots, K\}$ possesses a local dataset D_k, the aim is to learn a generalized global model over $D = \bigcup_{k=1}^{K} D_k$. The global objective function is defined as:

$$\arg\min_{w} L(w) = \sum_{k=1}^{K} \frac{|D_k|}{|D|} L_k(w), \tag{5}$$

where the local objective function $L_k(w)$ for client k measures the local empirical loss over the data distribution D_k and is given by:

$$L_k(w) = \mathbb{E}_{x \sim D_k}[\ell_k(w; x)], \tag{6}$$

Here, ℓ_k is the loss function for client k, while w denotes the global model parameters to be optimized. This work emphasizes addressing the issue of data heterogeneity in FL due to the non-IID distribution of data across clients.

3.2 Theoretical Analysis

As mentioned in the introduction, non-IID data distributions among clients in federated learning environment result in increased gradient diversity, particularly in the last layers of the network. To tackle this challenge, we suggest a straightforward but powerful enhancement to the standard FedAvg algorithm. Our approach introduces a carefully designed gradient penalty term with the standard SGD to align gradient norms effectively in the final layers. This method not only reduces the effects of client data heterogeneity but also enhances model performance and accelerates convergence In the FL framework, each client k is associated with a local dataset D_k and computes the gradient of the loss function with respect to the model parameters w'. The local loss function for client k is defined by Eq. 28.

$$L_k(w) = \frac{1}{|D_k|} \sum_{i \in D_k} \ell(w'; x_i, y_i) \tag{7}$$

where $\ell(w'; x_i, y_i)$ is the loss for sample (x_i, y_i). The gradient of the local loss function for client k is given by Eq. 29.

$$\nabla L_k(w') = \frac{1}{|D_k|} \sum_{i \in D_k} \nabla \ell(w'; x_i, y_i) \tag{8}$$

In the FedAvg algorithm, the global model parameters are updated according to the Eq. 30.

$$w_{t+1} = w_t - \eta_g \frac{1}{K} \sum_{k=1}^{K} \Delta w_k^{'} \tag{9}$$

where $\Delta w_k = \nabla L_k(w_t)$ is the local update from client k and w_t is the global model parameter for round t.

In IID settings, the data distribution remains consistent across all clients, leading to similar gradient norms. Let σ_{iid} represent the standard deviation of gradient norms in IID settings presented in Eq. 31.

$$E[||\nabla L_{iid}(w)||] = \sigma_{iid} \tag{10}$$

Here, w represents the global model parameter. In non-IID settings, where data distributions vary across clients, gradient norms exhibit higher variability compared to IID settings. We represent this in Eq. 32.

$$E[||\nabla L_{non-iid}(w)||] = \sigma_{non-iid} \gg \sigma_{iid} \tag{11}$$

The deeper layers of a neural network are responsible for learning more specific features, resulting in higher gradient norms in non-IID settings, as illustrated in Fig. 1. We formalize this relationship in Eq. 33.

$$E[||\nabla L_{l,non-iid}||] \gg E[||\nabla L_{l,iid}||] \tag{12}$$

where l denotes the index for the last layers. To address this gradient diversity, we propose to use a client-specific term called gradient penalty (ρ_i) to ensure better alignment of the gradients of the last layers of the model as presented in Eq. 34.

$$\Delta w_{t+1,l} = w_t - \eta_l \rho_l \nabla L_l \tag{13}$$

where ρ_l reduces gradient norm variations. We can choose ρ_l as presented in Eq. 35.

$$\rho_l = \frac{\sigma_{iid}}{||\nabla L_{l,non-iid}||} \tag{14}$$

This ensures the alignment of the gradients so that they become similar to IID settings as presented in Eq. 15.

$$||\rho_l \nabla L_{l,non-iid}|| = \sigma_{iid} \tag{15}$$

3.3 Proposed Method

Motivated by these empirical and theoretical observations, we introduce FedPGVC, an innovative method for managing data heterogeneity in federated learning. Our algorithm (Algo. 1) includes three main components: i) client update (Eq. 18 and Eq. 20), ii) computation of client gradient penalty term (Eq. 19), and iii) server update (Eq. 21). Initially, we define a vector $e \in \mathbb{R}^d$ containing 0 or 1, with v non-zero elements ($v \ll d$) as specified in Eq. 16. This vector serves as a mask to differentiate between the initial layers of the model and those adjacent to or comprising the classifier. For the subset of indices j where $e_j = 1$ (denoted as S_{gvc} in Eq. 17), we modify the corresponding weights $y_{i,S_{gvc}}$ to minimize variance. This is achieved by introducing a client-specific gradient penalty term ($\rho_i \in \mathbb{R}^v$) as formulated in Eq. 19. Subsequently, we update the weights of the corresponding layer using Eq. 20. For the remaining indices, denoted as S_{sgd}, we update the corresponding weights $y_{i,S_{sgd}}$ using standard SGD as formulated in Eq. 18. In each communication round, the process unfolds as follows: Every client receives a copy of the server model, denoted as ω. Subsequently, each client independently executes K model updating steps, leveraging the cross-entropy loss function as the optimization objective. These updating steps are governed by the equations (refer to Eq. 18, Eq. 19, and Eq. 20), which encapsulate the core operations involved in a single step. Once the local model updates are completed, the clients transmit their updated models, represented as y_i, back to the server. The server then aggregates these individual client models through the aggregation mechanism defined in Eq. 21.

$$e := \{0, 1\}, v = \Sigma e \tag{16}$$

$$S_{gvc} := \{j : e_j = 1\}, \quad S_{sgd} := \{j : e_j = 0\} \tag{17}$$

$$y_{(i,S_{sgd})} \leftarrow y_{(i,S_{sgd})} - \eta_l g_i(y_{(i,S_{sgd})}) \tag{18}$$

$$\rho_i \leftarrow \frac{1}{B} \sum_{b=1}^{B} \ell(\theta, x_b) \nabla_\theta \ell(\theta, x_b) \tag{19}$$

$$y_{(i,S_{gvc})} \leftarrow y_{(i,S_{gvc})} - \eta_l * \rho_i * g_i(y_{(i,S_{gvc})}) \tag{20}$$

$$\omega \leftarrow (1 - \eta g)\omega + \frac{1}{N} \sum_{i \in N} y_i \tag{21}$$

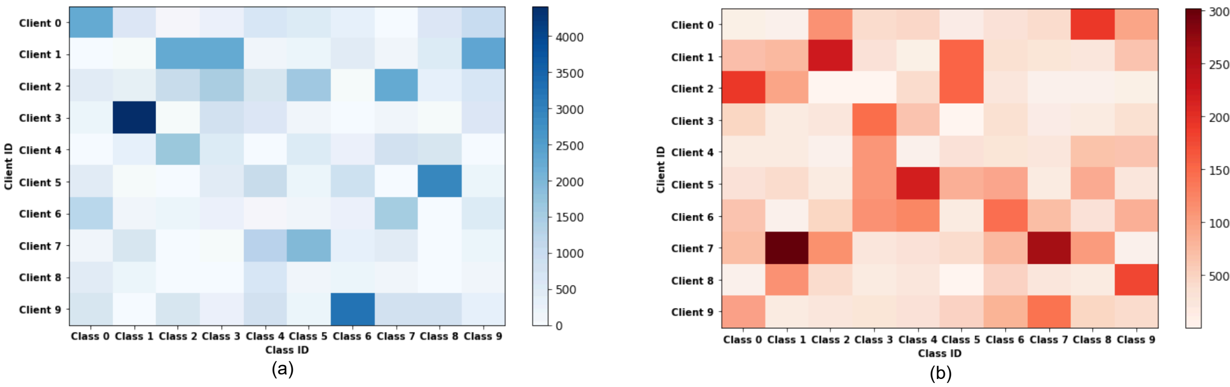
Where B is the batch size, $g_i(.)$ represents the gradient, $\ell(\theta, x_b)$ is the loss function evaluated on the b^{th} data point x_b with model parameters θ, $\nabla_\theta \ell(\theta, x_b)$ is the gradient of the loss function with respect to the model parameters θ, evaluated on the b^{th} data point x_b, η_l is the local learning rate, η_g is the global learning rate and N is the total number of clients. The convergence proof of the proposed method for both convex and non-convex settings is presented in Section 5 of the Appendix.

3.3.1 Usefulness of Introducing Gradient Penalty (ρ)

The intuition behind introducing the gradient penalty term ρ into the standard SGD for the last layers of the model is to encompass both the direction and strength of the gradients, along with the loss landscape for each client's data distribution. Prioritizing the gradients from clients which have data distributions that significantly deviate from global distribution allows us to better handle the most challenging situations within a specific range around each client's observed data distribution. Incorporating ρ into the weight updates for the final classification layers of the neural network allows us to achieve better alignment of these layers across clients, mitigating the issue of client drift caused by data heterogeneity. Specifically, we update the weights of the classification layers as presented in Eq. 19. This approach has the advantage of not requiring any additional communication overhead like prior methods such as SCAFFOLD and FedPVR Karimireddy et al. (2020); Li et al. (2023). Moreover, it strikes a balance between diversity and uniformity across the layers of the neural network, allowing the feature extraction layers to learn rich representations while ensuring better alignment of the final layers across clients.

Algorithm 1 Federated Partial Gradient Variance Control (FedPGVC)

```
Server: Initialize the global model parameters $\omega^0$, Global learning rate $\eta_g$.
Client: Initialize the local model parameters $y_i^0$, Local learning rate $\eta_l$.
Define a mask $e \in \{0,1\}^d$, where $e_j = 1$ for the last few layers and 0 for the rest layers.
Let $S_{sgd} = \{j : e_j = 0\}$ and $S_{gvc} = \{j : e_j = 1\}$.
for $r = 1, 2, \ldots, R$ do
    Server broadcasts the global model $\omega^0$ to all clients.
    for each client $i = 1, 2, \ldots, N$ in parallel do
        for $k = 1, 2, \ldots, K$ do
            $y_{i,S_{\text{sgd}}}^{(r,k)} = y_{i,S_{sgd}}^{(r,k-1)} - \eta_l \nabla_{S_{\text{sgd}}} f_i(y_i^{(r,k-1)})$
            $\rho_i^{r-1} \leftarrow \frac{1}{B} \sum_{b=1}^{B} \ell(\theta, x_b) \nabla_\theta \ell(\theta, x_b)$
            $y_{i,S_{\text{gvc}}}^{(r,k)} = y_{i,S_{gvc}}^{(r,k-1)} - \eta_l \nabla_{S_{\text{gvc}}} L_i(y_i^{(r,k-1)}, \rho_i^{r-1})$
        end for
        Client $i$ sends the updated model $y_i^{(r,K)}$ to the server.
    end for
    Server aggregates the client models and updates the global model:
    $\omega^r = (1 - \eta_g)\omega^{(r-1)} + \frac{1}{N} \sum_i y_i^{(r,K)}$
end for
```



(a) (b)

Figure 2: The distribution of MNIST data, indicating the number of images per client per class, varied according to different levels of heterogeneity, with (a) $\alpha = 0.5$ representing severe non-iid and (b) $\alpha = 1.0$ indicating moderate non-iid.

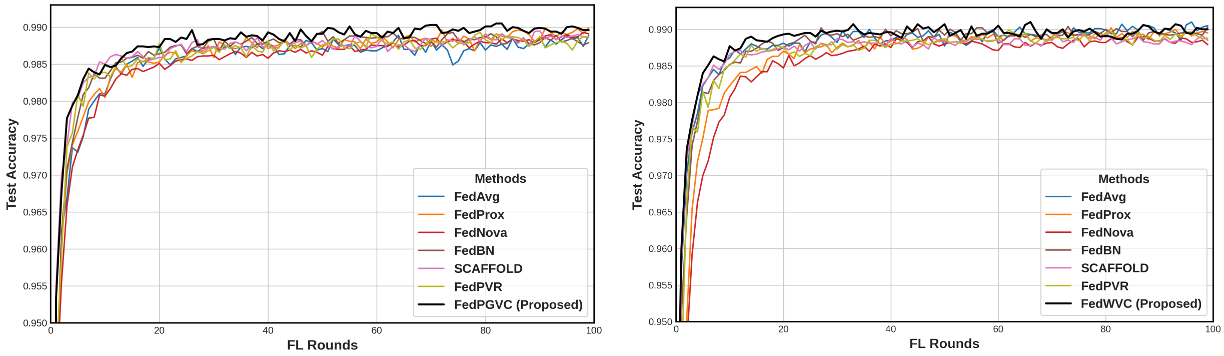
4 Experimental results

4.1 Experimental setup

To evaluate the efficacy of FedPGVC, we conducted comprehensive experiments using three widely recognized computer vision classification benchmarks: MNIST LeCun et al. (2010), FMNIST Xiao et al. (2017), and CIFAR100 Krizhevsky (2009). For robustness and reliability, all experiments were repeated three times with different seeds, and average value is reported along with standard deviation. We partitioned the entire dataset client-wise using a strategy inspired by Lin et al. (2020) to create a real-world non-IID distribution. This was achieved by distributing the data among clients using a Dirichlet distribution with a concentration parameter α, which can take any real value. The measure of data heterogeneity across clients is governed by the α with a smaller value, resulting in a more skewed data distribution, mimicking real-world scenarios where data is unevenly partitioned. Figure 2 illustrates an example of such a non-uniform data distribution for the MNIST dataset. In our experiments, we adopted α values of 0.5 and 1.0, which are commonly employed values Lin et al. (2020) to simulate varying levels of data heterogeneity. Each client possesses its own local data partition, which remains unchanged throughout the communication rounds. This static data distribution allows us to access the performance of our proposed method under realistic conditions where clients do not exchange data. To assess the classification performance of the global model, we hold out a test dataset at the server, which remains unseen during the training process. For our experiments, we utilized the well-established LeNet LeCun et al. (1998) neural network for the MNIST and FMNIST datasets. For CIFAR-100, we employed an 9-layer CNN following the approach described in Duan et al. (2023). We applied the variance reduction technique to the last two layers of the selected models to address data heterogeneity. Our experimental setup involved 10 participating clients in each communication round, with a batch size of 32, consistent with the configurations reported in prior studies Li et al. (2023) and Yu et al. (2022). In our experimental setup, each client performed two local epochs of model updating. Consistent with the configuration outlined in Karimireddy et al. (2020), we fixed the server learning rate $\eta_g = 1$. To determine the optimal client learning rate for each experiment, we conducted a grid search over $0.05, 0.01, 0.2, 0.3$. Our implementation of FedProx involved testing a range of proximal values 0.001, 0.1, 0.4, 0.7 to determine the optimal setting. For FedNova, we selected the best proximal SGD value from the set 0.001, 0.003, 0.05, 0.1, in accordance with the recommendations in Li et al. (2024). Across all experiments, we employed the Adam optimizer for consistency.

Table 2: Top-1 accuracy (%) on MNIST, FMNIST, and CIFAR100 datasets with varying degrees of data heterogeneity. The values in bold represent the highest accuracy achieved. Standard deviation values are provided in parentheses.

	MNIST		FMNIST		CIFAR100	
	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 1.0$
Fedavg	99.00 (±0.03)	98.89 (±0.05)	88.65 (±0.22)	89.14 (±0.18)	24.25 (±0.16)	25.00 (±0.14)
FedProx	98.99 (±0.04)	99.02 (±0.03)	86.96 (±0.30)	89.10 (±0.21)	24.89 (±0.15)	25.56 (±0.13)
FedNova	98.91 (±0.05)	98.79 (±0.06)	87.52 (±0.25)	88.82 (±0.23)	22.29 (±0.19)	24.92 (±0.15)
FedBN	98.94 (±0.04)	99.03 (±0.04)	88.76 (±0.20)	89.17 (±0.18)	25.12 (±0.12)	25.66 (±0.13)
SCAFFOLD	98.95 (±0.05)	98.95 (±0.05)	87.98 (±0.28)	88.41 (±0.22)	24.30 (±0.17)	25.27 (±0.16)
FedPVR	98.93 (±0.04)	98.99 (±0.05)	87.28 (±0.31)	88.37 (±0.26)	20.59 (±0.25)	17.57 (±0.20)
Proposed	**99.05 (±0.02)**	**99.04 (±0.03)**	**88.83 (±0.18)**	**89.35 (±0.17)**	**25.29 (±0.11)**	**25.74 (±0.12)**

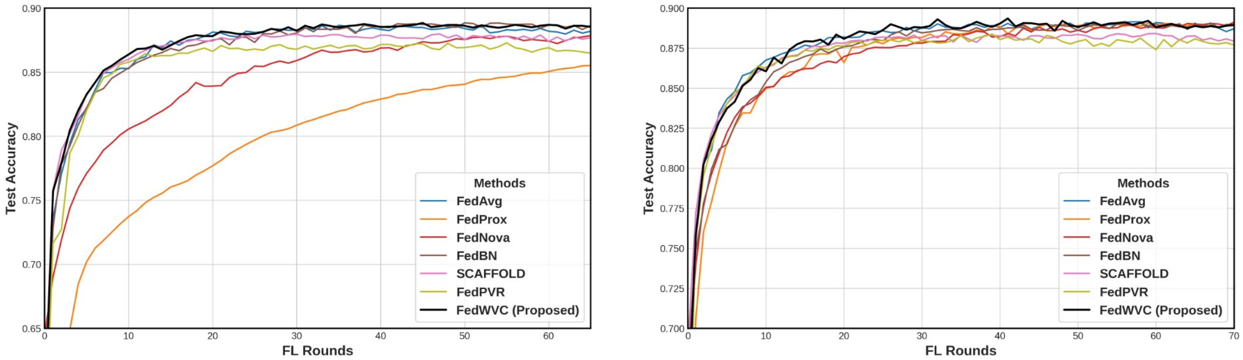


(a) (b)

Figure 3: The performance comparison of proposed FedPGVC with baseline approaches: (a) and (b) depict the graphs on the MNIST dataset for $\alpha = 0.5$ and 1.0 respectively.

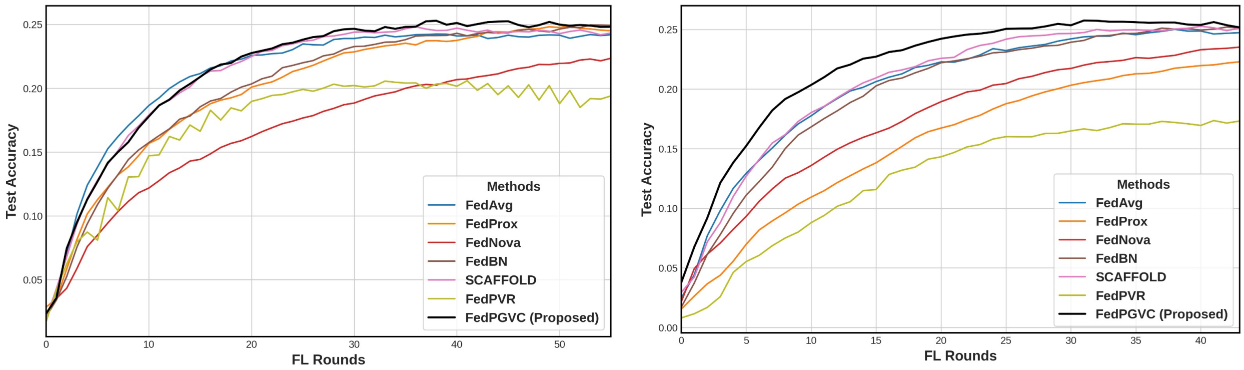
4.2 Comparison with the State-of-the-art Methods

We evaluate our proposed FedPGVC against several notable FL algorithms, including FedAvg, FedProx, FedNova, FedBN, SCAFFOLD and FedPVR. FedPGVC consistently achieves the highest accuracy across



(a) (b)

Figure 4: The performance comparison of proposed FedPGVC with baseline approaches: (a) and (b) depict the graphs on the FMNIST dataset for $\alpha = 0.5$ and 1.0 respectively.



(a) (b)

Figure 5: The performance comparison of proposed FedPGVC with baseline approaches: (a) and (b) depict the graphs on the CIFAR100 dataset for $\alpha = 0.5$ and 1.0 respectively.

varying data heterogeneity levels ($\alpha = 0.5$ and 1) on MNIST, FMNIST, and CIFAR100 datasets. On the MNIST dataset with ($\alpha = 0.5$), FedPGVC achieves a minimum accuracy improvement of 0.05% compared to FedAvg and a maximum of 0.14% compared to FedNova. With ($\alpha = 1.0$), it shows a minimum improvement of 0.01% over FedBN and a maximum of 0.25% over FedNova. For the FMNIST dataset, under $\alpha = 0.5$ settings, FedPGVC attains a minimum improvement of 0.07% over FedBN and a maximum of 1.87% over FedProx. With $\alpha = 1.0$, it achieves a minimum improvement of 0.18% over FedBN and a maximum of 0.98% over FedPVR. On the CIFAR-100 dataset with $\alpha = 0.5$, FedPGVC shows a minimum improvement of 0.17% compared to FedBN and a maximum of 4.7% compared to FedPVR. Under $\alpha = 1.0$, it achieves a minimum improvement of 0.08% over FedBN and a maximum of 8.17% over FedPVR. While our method consistently outperforms baseline approaches across all datasets, the performance gain on MNIST is less pronounced. This can be attributed to the fact that MNIST is a relatively simple and well-studied dataset with low intraclass variability and minimal noise. As a result, most modern federated learning algorithms already achieve near-optimal accuracy on MNIST, leaving little room for further improvement. The narrow margin of improvement suggests that MNIST has reached a saturation point, making even slight gains difficult. Additionally, the dataset's uniformity and simplicity reduce the impact of data heterogeneity, the primary challenge our method is designed to address. The stronger performance gains on more complex datasets like CIFAR-100, which exhibit higher intraclass variability and are more susceptible to the challenges of data heterogeneity, further highlight the strengths of our approach.

Table 3: Number of communication rounds required (speedup compared to FedAvg) to achieve specific top-1 accuracy levels (99% for MNIST, 88% for FMNIST and 24% for CIFAR100). FedPGVC outperforms other methods by requiring fewer rounds to achieve comparable accuracy. '*' denotes the algorithm failed to achieve given test accuracy.

	MNIST		FMNIST		CIFAR100	
	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 1.0$
	Number of rounds		Number of rounds		Number of rounds	
FedAvg	100 (1.0x)	100 (1.0x)	20 (1.0x)	20 (1.0x)	30 (1.0x)	30 (1.0x)
FedProx	77 (1.2x)	74 (1.3x)	*	28 (0.7x)	43 (0.69x)	*
FedNova	100 (1.0x)	*	*	34 (0.5x)	*	*
FedBN	50 (2.0x)	28 (3.5x)	26 (0.7x)	24 (0.8x)	*	33 (0.90x)
SCAFFOLD	*	*	*	25 (0.8x)	30 (1.0x)	30 (1.0x)
FedPVR	*	*	*	22 (0.9x)	*	*
Proposed	**23 (4.3x)**	**27 (3.7x)**	**18 (1.1x)**	**15 (1.3x)**	**27 (1.1x)**	**20 (1.5x)**

4.3 Convergence Analysis

Figures 3, 4, and 5 depict the learning efficiency of FedPGVC in comparison to baseline methods on the MNIST, FMNIST, and CIFAR100 datasets, respectively. FedPGVC consistently demonstrates faster learning and higher accuracy across various settings. On the MNIST dataset, FedPGVC achieves near 99% accuracy within the first 23-27 rounds (Fig. 3), outperforming baselines that require more rounds for similar performance. For FMNIST, FedPGVC reaches close to 88% accuracy in 15-18 rounds (Fig. 4), again faster than the baselines. The advantage is even more pronounced on the CIFAR-100 dataset, where FedPGVC not only converges faster but also achieves higher final accuracy (Fig. 5), significantly outperforming the baselines. Overall, FedPGVC requires fewer rounds across all experiments (refer to Table 3), achieving a speedup of 1.1 to 4.3 times compared to baselines. This efficiency is attributed to the variance reduction technique in FedPGVC, which promotes rapid convergence and mitigates the negative impact of data heterogeneity among clients.

4.4 Effect of Partial Gradient Variance Control (PGVC)

We integrated the proposed PGVC with standard FL approaches, and the results are presented in Table 4. The table shows that most approaches incorporating PGVC on the client side improved overall accuracy across datasets. FedAvg with PGVC achieves a 0.18% improvement on the FMNIST dataset and a 1.04% improvement on CIFAR100. Similarly, FedPGVC enhances accuracy with FedProx and FedNova. However, FedBN with PGVC results in lower accuracy than FedBN alone, likely due to conflicting effects on training dynamics and model regularization. These findings demonstrate that the proposed PGVC approach can be effectively integrated with existing FL algorithms. For a comprehensive view of model convergence, we have presented the learning curves in Fig. 10 and Fig. 11 of the Appendix. In terms of computational efficiency, our proposed FedPGVC method requires 18 minutes for CIFAR100 training, compared to 12 minutes for standard FedAvg. Similarly, on FMNIST, training time increases from 15 to 21 minutes. While this represents a modest increase in computational cost, we argue that the significant accuracy gains justify this trade-off, especially in scenarios where model performance is paramount.

4.5 Applying PGVC on the Different Layers of the Model

Given that the proposed approach partially applies the gradient variance control technique in the last layers of the neural network, we investigated the effects of incorporating variance reduction in different layers. We conducted experiments on the CIFAR100 dataset with $\alpha = 0.5$, as shown in Fig. 6. The results indicate that initiating variance reduction in the final layers of the model facilitates faster convergence and achieves the highest top-1 accuracy. Activating variance control in layers closer to the classifier shows minimal impact on performance while employing the technique in the initial layers results in significant performance degradation.

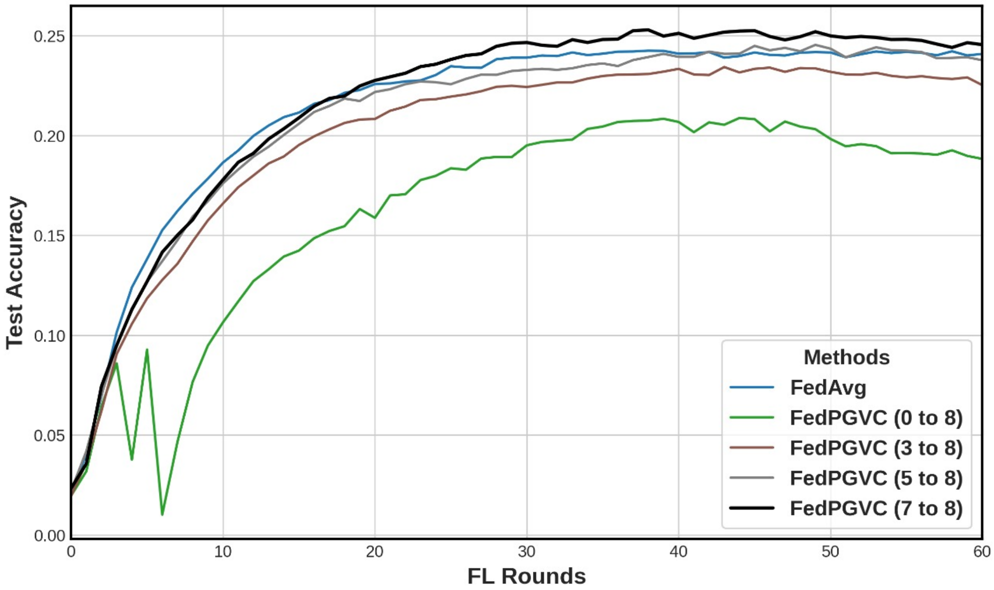


Figure 6: Performance of applying partial gradient variance control on different layers of the CNN model on the CIFAR100 dataset with $\alpha = 0.5$.

Preserving diversity in the middle and early layers enables the learning of rich feature representations while promoting uniformity in the classifier layers, which helps make less biased decisions. This balance is crucial for leveraging the collective knowledge of distributed models while mitigating the adverse effects of excessive variance. We have conducted the same experiment on the FMNIST dataset with $\alpha = 0.5$, and the results are presented in Section 6 of the Appendix.

4.6 Ablation study

We conducted two ablation studies. First, we assessed the scalability of our proposed method by varying the number of clients participating in the federated learning process. Second, we applied our method to IID data to compare its performance against the baselines.

4.6.1 Scalibility

To demonstrate the scalability of the proposed FedPGVC in practical settings, we conducted experiments on the FMNIST dataset ($\alpha = 0.5$) with varying numbers of clients: 10, 20, and 50. Figure 7 depicts the performance of the proposed model with the baselines. Notably, when the number of clients increases, the accuracy drop of FedPGVC is considerably low compared to the baselines. This observation highlights the robustness and scalability of our approach, as it can effectively harness a larger pool of clients while maintaining high accuracy.

4.6.2 Results on IID data

To assess the efficacy of the proposed FedPGVC method on IID datasets, we created an IID partition of the FMNIST dataset by setting $\alpha = 100$ and compared its performance against other baselines. The results, shown in Fig. 8, indicate that FedPGVC performs similarly to the baseline methods in the IID setting. This finding highlights that FedPGVC is not only effective for non-IID data partitions but also performs well in IID data settings.

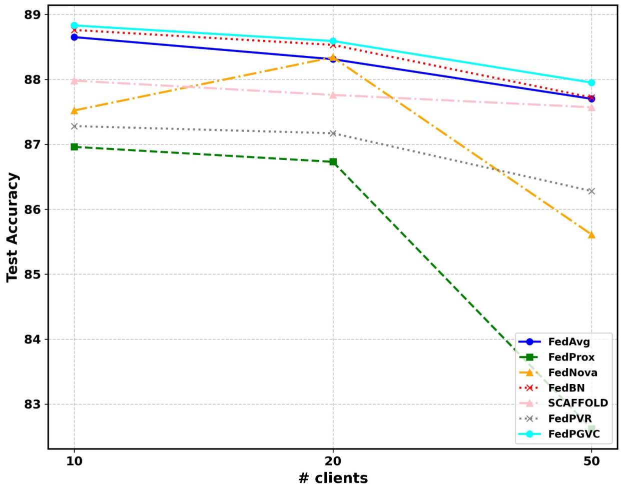


Figure 7: Performance of test accuracy with varying numbers of clients for all baseline models and the proposed approach.

Table 4: The effect of applying the proposed method to existing popular baselines on the FMNIST and CIFAR100 dataset with $\alpha = 0.5$.

Method	FMNIST	CIFAR100
FedAvg	88.65	24.25
FedAvg + PGVC	**88.83** (0.18 $\uparrow$)	**25.29** (1.04 $\uparrow$)
FedProx	86.96	**24.89**
FedProx + PGVC	**88.07** (1.11 $\uparrow$)	24.58 (0.31 $\downarrow$)
FedBN	**88.86**	**25.12**
FedBN + PGVC	88.11 (0.75 $\downarrow$)	23.86 (1.26 $\downarrow$)
FedNova	87.52	22.29
FedNova + PGVC	**87.87** (0.35 $\uparrow$)	**24.82** (2.53 $\uparrow$)

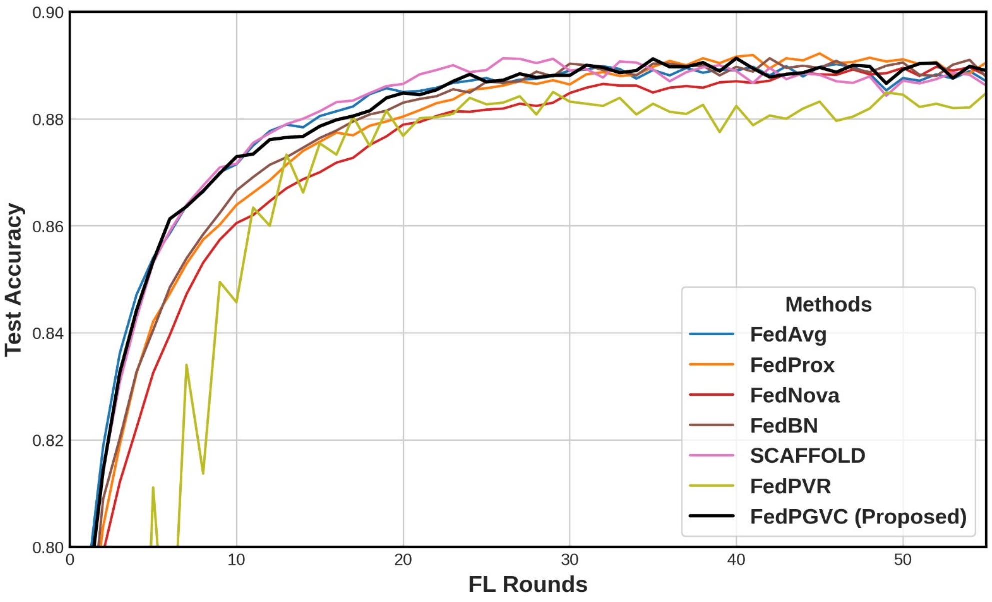


Figure 8: Performance of the proposed model compared to the baselines in FMNIST IID data settings with $\alpha = 100$.

4.7 Conclusion

This research introduces FedPGVC, an innovative FL-based approach to address the challenges posed by heterogeneous data distributions among clients. By integrating a gradient penalty term into the partial variance control strategy, FedPGVC effectively mitigates the adverse effects of data heterogeneity in federated learning environments. Extensive experiments on diverse datasets reveal FedPGVC's advantage over state-of-the-art baseline methods. Moreover, FedPGVC exhibits faster convergence rates and excellent scalability, consistently delivering performance benefits as the number of clients increases, thus positioning it as a promising solution for large-scale, real-world FL-based computer vision applications.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Jian-hui Duan, Wenzhong Li, Derun Zou, Ruichen Li, and Sanglu Lu. Federated learning with data-agnostic distribution fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8074–8083, 2023.

Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

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.

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

Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010.

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

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 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, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

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

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

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

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pp. 1000–1008. PMLR, 2014.

Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. *arXiv preprint arXiv:2204.13399*, 2022.

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

Yaodong Yu, Alexander Wei, Sai Praneeth Karimireddy, Yi Ma, and Michael Jordan. Tct: Convexifying federated learning using bootstrapped neural tangent kernels. *Advances in Neural Information Processing Systems*, 35:30882–30897, 2022.

Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and Salman Avestimehr. Federated learning for internet of things: Applications, challenges, and opportunities, 2022. URL `https://arxiv.org/abs/2111.07494`.

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

5 Convergence Proof

We perform an analysis of the convergence rate for the proposed FedPGVC, considering both convex and non-convex scenarios. To enable a theoretical proof, we introduce the following notations and assumptions: we consider N clients, each linked to a local objective function $f_i(x)$, where $i = 1, \ldots, N$. We impose the following assumptions on the objective functions:

Assumption 1: Lipschitz smoothness: $|\nabla f_i(x) - \nabla f_i(y)| \leq L|x-y|$ for all x, y and some constant $L > 0$.

Assumption 2: Bounded gradients: $\mathbb{E}|\nabla f_i(x; \xi_i)|^2 \leq G^2$ for all x and some constant $G > 0$, where ξ_i denotes the random variable representing the data samples used to compute stochastic gradients on client i.

Assumption 3: Non-convexity: We have first considered the non-convex setting, which is more general and applicable to deep neural networks. Additionally, we assume a measure of data heterogeneity across clients $\hat{\zeta}^2$ such that:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}|\nabla f_i(x)|^2 \leq \hat{\zeta}^2, \quad \forall x \tag{22}$$

The proof of the above three assumptions can be found in Karimireddy et al. (2020).

5.0.1 Proof of Convergence in non-convex setting:

For the non-convex setting, we can derive the following descent lemma. Let us denote the global model parameters at the beginning of round r as $x^{(r)}$, and the local model parameters on client i after k local updates as $y_i^{(r,k)}$. The modified SGD update rule for the local model on client i is given in Eq. 23.

$$y_i^{(r,k+1)} = y_i^{(r,k)} - \eta_l\left(g_i^{(r,k)} + \rho_i^{(r,k)} \odot e\right) \tag{23}$$

where $g_i^{(r,k)} = \nabla f_i(y_i^{(r,k)}; \xi_i^{(r,k)})$ is the stochastic gradient on client i, $\rho_i^{(r,k)} = \mathbb{E}[g_i^{(r,k)} \odot (g_i^{(r,k)} - e \odot x^{(r)})]$ is a vector denoting the gradient penalty term, e is the masking vector specifying which layers to apply modified SGD on, and $\odot$ denotes element-wise multiplication. The update rule for the global model after aggregating the client models is given in Eq. 24.

$$x^{(r+1)} = \frac{1}{N}\sum_{i=1}^{N} y_i^{(r,K)} \tag{24}$$

where K is the number of local updates performed by each client.

We will now derive a descent lemma that relates the expected decrease in the objective function after one round of client updates and global model aggregation. Let $F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ be the average of the client objective functions. Using the Lipschitz smoothness assumption and the update rules, we can show:

$$\begin{aligned}
\mathbb{E}[F(x^{(r+1)})] &\leq \mathbb{E}\left[F\left(\frac{1}{N}\sum_{i=1}^{N} y_i^{(r,K)}\right)\right] + \frac{L}{2N}\sum_{i=1}^{N}\mathbb{E}\left|y_i^{(r,K)} - x^{(r)}\right|^2 \\
&\leq \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[f_i\left(y_i^{(r,K)}\right)\right] + \frac{L}{2N}\sum_{i=1}^{N}\mathbb{E}\left|y_i^{(r,K)} - x^{(r)}\right|^2 \\
= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[f_i\left(y_i^{(r,0)}\right)\right] + \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}&\mathbb{E}\left[f_i\left(y_i^{(r,k+1)}\right) - f_i\left(y_i^{(r,k)}\right)\right] + \frac{L}{2N}\sum_{i=1}^{N}\mathbb{E}\left|y_i^{(r,K)} - x^{(r)}\right|^2
\end{aligned} \tag{25}$$

Using the Lipschitz smoothness assumption again and the update rules, we can further bound the second term as:

$$\begin{aligned}
\mathbb{E}\left[f_i\left(y_i^{(r,k+1)}\right) - f_i\left(y_i^{(r,k)}\right)\right] &\leq \langle\nabla f_i\left(y_i^{(r,k)}\right), \mathbb{E}\left[y_i^{(r,k+1)} - y_i^{(r,k)}\right]\rangle + \\
&\qquad \frac{L}{2}\mathbb{E}\left|y_i^{(r,k+1)} - y_i^{(r,k)}\right|^2 \\
&= -\eta_l\langle\nabla f_i\left(y_i^{(r,k)}\right), g_i^{(r,k)} + \rho_i^{(r,k)} \odot e\rangle + \\
&\qquad \frac{L\eta_l^2}{2}\mathbb{E}\left|g_i^{(r,k)} + \rho_i^{(r,k)} \odot e\right|^2
\end{aligned} \tag{26}$$

Substituting the bound obtained from Eq. 26 back into the descent lemma obtained in Eq. 25 and rearranging terms, we get Eq. 27.

$$\begin{aligned}\mathbb{E}[F(x^{(r+1)})] \leq \frac{1}{N}\sum_{i=1}^{N} f_i\left(x^{(r)}\right) - \frac{\eta_l}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\langle \nabla f_i\left(y_i^{(r,k)}\right), g_i^{(r,k)}\right\rangle\right] \\ -\frac{\eta_l}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left[\left\langle \nabla f_i\left(y_i^{(r,k)}\right), \rho_i^{(r,k)} \odot e\right\rangle\right] + \frac{L\eta_l^2}{2N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}\left|g_i^{(r,k)} + \rho_i^{(r,k)} \odot e\right|^2 \\ +\frac{L}{2N}\sum_{i=1}^{N}\mathbb{E}\left|y_i^{(r,K)} - x^{(r)}\right|^2\end{aligned} \tag{27}$$

To telescope the descent lemma obtained in Eq. 27 over multiple FL rounds, we will make use of the inequality 28, where we used the bounded gradients assumption and the data heterogeneity measure.

$$\begin{aligned}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\left\langle \nabla f_i\left(y_i^{(r,k)}\right), g_i^{(r,k)} - \nabla f_i\left(y_i^{(r,k)}\right)\right\rangle \leq \frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\left(\left|\nabla f_i\left(y_i^{(r,k)}\right)\right|^2 + \left|g_i^{(r,k)}\right|^2\right) \\ \leq \frac{KG^2}{2} + \frac{K\hat{\zeta}^2}{2}\end{aligned} \tag{28}$$

Similarly, we can bound the term involving the gradient penalty term $\rho_i^{(r,k)}$ as:

$$\sum_{i=1}^{N}\sum_{k=0}^{K-1}\left\langle \nabla f_i\left(y_i^{(r,k)}\right), \rho_i^{(r,k)} \odot e\right\rangle \quad \leq \frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\left(\left|\nabla f_i\left(y_i^{(r,k)}\right)\right|^2 + \left|\rho_i^{(r,k)} \odot e\right|^2\right) \leq \frac{K\hat{\zeta}^2}{2} + \frac{K\hat{\zeta}_e^2}{2} \tag{29}$$

where $\hat{\zeta}_p^2$ is a measure of the heterogeneity of the gradients for the layers where modified SGD is applied, which is defined in Eq. 30.

$$\hat{\zeta}p^2 = \frac{1}{N}\sum i = 1^N \mathbb{E}\left|\rho_i^{(r,k)} \odot e\right|^2 \tag{30}$$

Substituting bounds obtained from Eq. 29 and 30 and using Eq. 30 into the descent lemma obtained in Eq. 27 and telescoping over R FL rounds, we get Eq. 31.

$$\begin{aligned}\frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}[F(x^{(r)})] - F^* \leq \frac{1}{R\eta_l}\left(\frac{L\eta_l^2}{2} + \frac{L}{2N}\right)\sum_{r=0}^{R-1}\sum_{i=1}^{N}\mathbb{E}\left|y_i^{(r,K)} - x^{(r)}\right|^2 \\ + \frac{1}{2\eta_l}\left(\frac{KG^2}{N} + K\hat{\zeta}^2 + K\hat{\zeta}_p^2\right) + \frac{1}{R}\left(F(x^{(0)}) - F^*\right)\end{aligned} \tag{31}$$

where F^* is the optimal value of the average objective function $F(x)$.

To optimize the convergence rate bound, we need to choose the learning rates η_l and η_g (the global learning rate, which we have not explicitly used yet but will be needed for the final convergence rate). We can set $\eta_g = \sqrt{N}$ and $\eta_l = \min\left\{\frac{1}{26K\eta_g L}, \frac{1}{\sqrt{KL}}\right\}$ to balance the terms in the bound. With these choices, and after some algebraic simplifications, we obtain the following convergence rate bound for non-convex functions as given in Eq. 32.

$$\frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}[F(x^{(r)})] - F^* = \mathcal{O}\left(\frac{G\sqrt{K}}{\sqrt{NR}} + \frac{\hat{\zeta}\sqrt{K}}{\sqrt{R}} \quad + \frac{\hat{\zeta}_p\sqrt{K}}{\sqrt{NR}} + \frac{F(x^{(0)}) - F^*}{R}\right) \tag{32}$$

This bound shows that the convergence rate of our approach depends on the number of communication rounds R, the number of local updates K, the gradient bound G, the overall data heterogeneity $\hat{\zeta}$, and the heterogeneity of the gradients for the layers where modified SGD is applied, $\hat{\zeta}_e$. In the favorable case where $\hat{\zeta}_e$ is small (i.e. when the gradients for the layers where modified SGD is applied are more aligned across clients), the convergence rate of our approach can be comparable to or better than existing methods like FedAvg McMahan et al. (2017) or FedPVR Li et al. (2023), depending on the values of the other problem-specific constants.

5.0.2 Proof of Convergence in convex setting

For the convex setting, we make the following assumptions:

Assumption 1: Lipschitz Smoothness (as described above)

Assumption 2: Bounded Gradients (as described above)

Assumption 3: Convexity: The local objective functions $f_i(x)$ are convex for all $i = 1, \dots, N$. Using these assumptions, we can derive a different descent lemma as in Eq. 33.

$$\begin{aligned}\mathbb{E}[F(x^{(r+1)})] \leq \frac{1}{N}\sum_{i=1}^{N} f_i(x^{(r)}) - \frac{\eta_l}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}[\langle \nabla f_i(y_i^{(r,k)}), g_i^{(r,k)}\rangle] \\ -\frac{\eta_l}{N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}[\langle \nabla f_i(y_i^{(r,k)}), \rho_i^{(r,k)} \odot e\rangle] + \frac{L\eta_l^2}{2N}\sum_{i=1}^{N}\sum_{k=0}^{K-1}\mathbb{E}|g_i^{(r,k)} + \rho_i^{(r,k)} \odot e|^2 \\ +\frac{L}{2N}\sum_{i=1}^{N}\mathbb{E}|y_i^{(r,K)} - x^{(r)}|^2\end{aligned} \tag{33}$$

Following similar steps as in the non-convex case, we can iteratively apply inequality 33 over multiple FL rounds and use the convexity assumption to obtain inequality 34.

$$\begin{aligned}\frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}[F(x^{(r)})] - F^* \leq \frac{1}{R\eta_l}\left(\frac{L\eta_l^2}{2} + \frac{L}{2N}\right)\sum_{r=0}^{R-1}\sum_{i=1}^{N}\mathbb{E}|y_i^{(r,K)} - x^{(r)}|^2 + \\ \frac{1}{2\eta_l}\left(\frac{KG^2}{N} + K\hat{\zeta}^2 + K\hat{\zeta}_p^2\right)\end{aligned} \tag{34}$$

Note that in the convex case, we don't have the term $(F(x^{(0)}) - F^*)/R$ since the objective function is convex, and we can initialize the model at the optimal point. Optimizing the bound by setting $\eta_g = \sqrt{N}$ and $\eta_l = \min\left\{\frac{1}{26K\eta_g L}, \frac{1}{\sqrt{KL}}\right\}$ as before, we get the following convergence rate for convex functions as in Eq. 35.

$$\frac{1}{R}\sum_{r=0}^{R-1}\mathbb{E}[F(x^{(r)})] - F^* = \mathcal{O}\left(\frac{G\sqrt{K}}{\sqrt{NR}} + \frac{\hat{\zeta}\sqrt{K}}{\sqrt{R}} + \frac{\hat{\zeta}_p\sqrt{K}}{\sqrt{NR}}\right) \tag{35}$$

This bound is similar to the non-convex case but without the $(F(x^{(0)}) - F^*)/R$ term due to the convexity assumption. The convergence rate depends on the number of communication rounds R, the number of local updates K, the gradient bound G, the overall data heterogeneity $\hat{\zeta}$, and the heterogeneity of the gradients for the layers where modified SGD is applied, $\hat{\zeta}_e$.

The convergence rate of the proposed FedPGVC algorithm relies on factors such as data heterogeneity among clients, the frequency of local updates, and the duration of communication rounds. In non-convex scenarios, initialization also plays a crucial role. When gradients align well for the last layers (indicating small $\hat{\zeta}_e$), FedPGVC shows potential for superior convergence rates compared to FedAvg. The convergence rates demonstrate similar dependencies as FedAvg, with additional considerations for the heterogeneity of last-layer gradients that are updated with modified SGD.

6 Applying PGVC on the different layers of the model

To evaluate the impact of incorporating gradient variance control in various neural network layers, we conducted experiments on the FMNIST dataset with $\alpha = 0.5$. As illustrated in Fig. 9, our findings reveal that applying variance reduction in the final layers accelerates convergence and achieves the highest top-1 accuracy. Given that the proposed approach partially applies the gradient variance control technique in the last layers of the neural network, we investigated the effects of incorporating variance reduction in different layers. We conducted experiments on the FMNIST dataset with $\alpha = 0.5$, as shown in Fig. 9. The results indicate that initiating variance reduction in the final layers of the model facilitates faster convergence and achieves the highest top-1 accuracy.

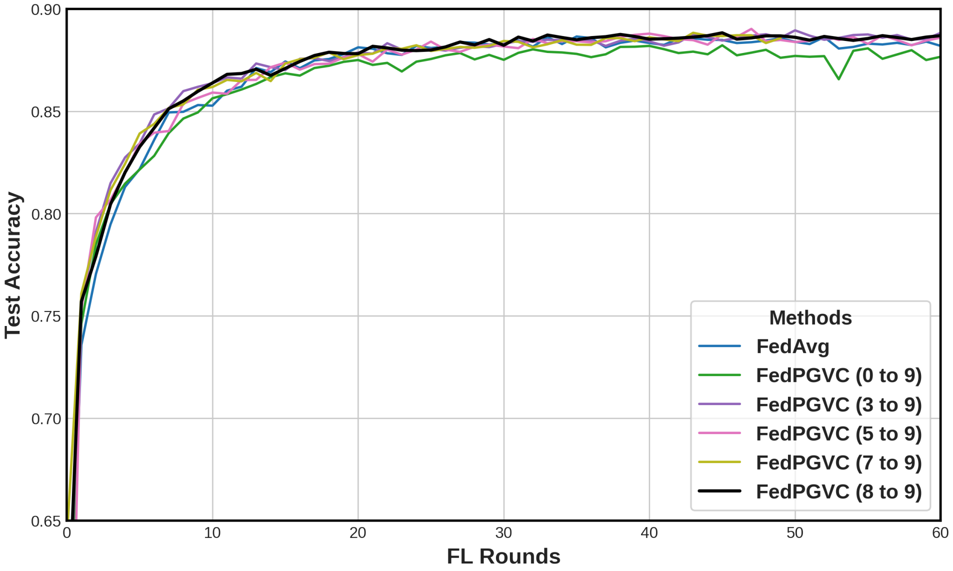


Figure 9: Performance of applying partial gradient variance control on different layers of the CNN model on the FMNIST dataset with $\alpha = 0.5$.

7 Limitation

While the proposed approach demonstrates significant improvements across various datasets, it is essential to acknowledge certain limitations. Although the method effectively reduces gradient variability in the final layers, the computation of the gradient penalty term may introduce additional processing overhead on the client side. This increased computational cost could be a constraint for resource-limited devices, even though the method is designed to minimize communication overhead. To address this challenge, future research could focus on reducing the computational complexity of the gradient penalty term without sacrificing the method's effectiveness. Approaches like model pruning or quantization may be explored to make the method more viable for devices with limited resources. Additionally, incorporating differential privacy techniques into the FedPGVC framework could broaden its applicability in privacy-sensitive areas, such as healthcare or finance. Investigating the interaction between differential privacy and the gradient penalty term, as well as its impact on model performance, would be a valuable direction for future research.

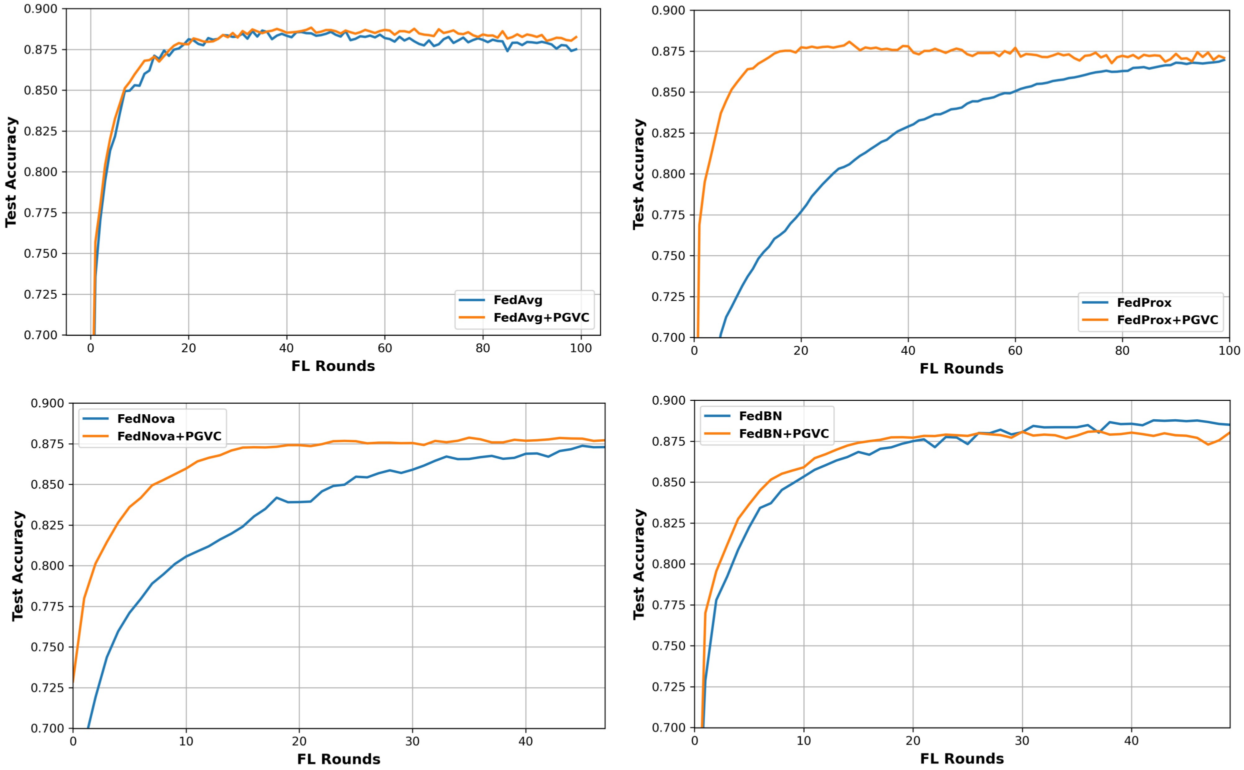


Figure 10: Learning curves illustrating the integration of the proposed PGVC technique with the existing algorithms on the FMNIST dataset at $\alpha = 0.5$.

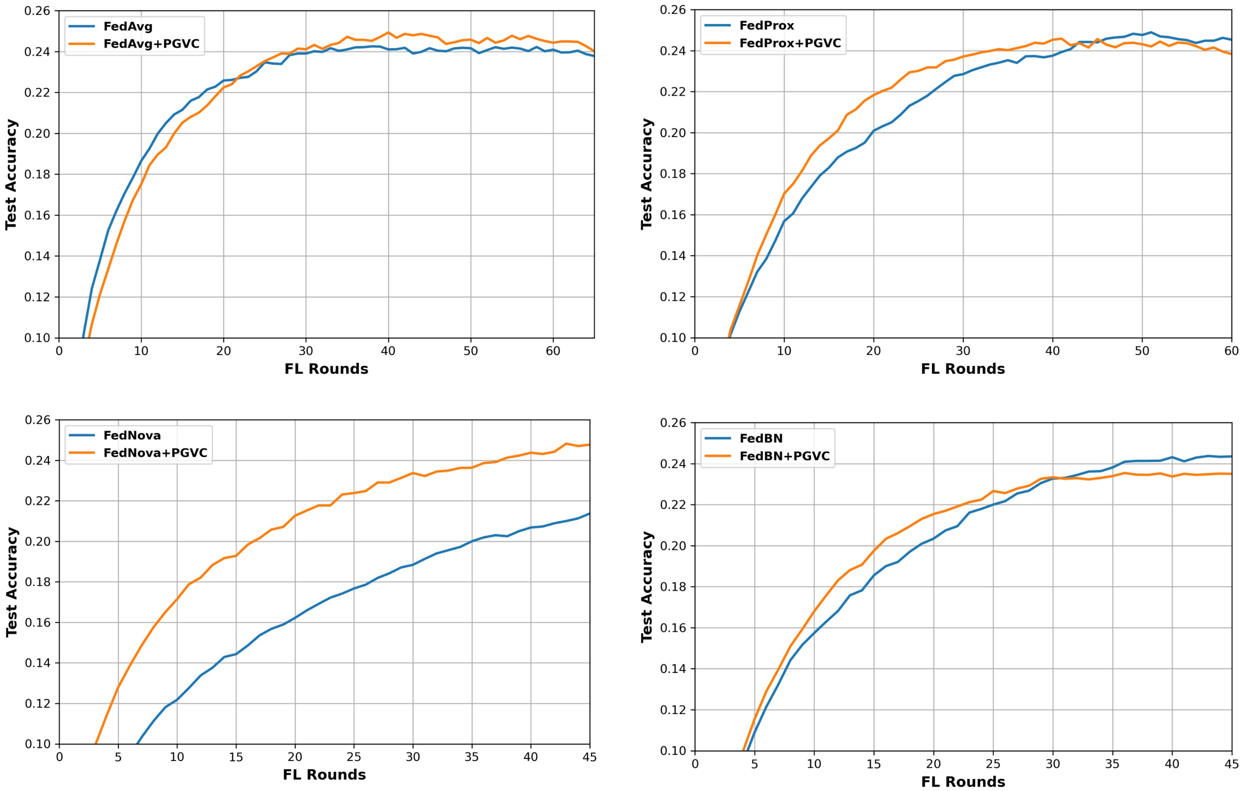


Figure 11: Learning curves illustrating the integration of the proposed PGVC technique with the existing algorithms on the CIFAR100 dataset at $\alpha = 0.5$.