# FedFN: Feature Normalization for Alleviating Data Heterogeneity Problem in Federated Learning

**Seongyoon Kim**
Dept. ISysE, KAIST
curisam@kaist.ac.kr

**Gihun Lee**
Graduate School of AI, KAIST
opcrisis@kaist.ac.kr

**Jaehoon Oh***
SAIT
jh0104.oh@gmail.com

**Se-Young Yun***
Graduate School of AI, KAIST
yunseyoung@gmail.com

## Abstract

Federated Learning (FL) is a collaborative method for training models while preserving data privacy in decentralized settings. However, FL encounters challenges related to data heterogeneity, which can result in performance degradation. In our study, we observe that as data heterogeneity increases, feature representation in the FedAVG model deteriorates more significantly compared to classifier weight. Additionally, we observe that as data heterogeneity increases, the gap between higher feature norms for observed classes, obtained from local models, and feature norms of unobserved classes widens, in contrast to the behavior of classifier weight norms. This widening gap extends to encompass the feature norm disparities between local and the global models. To address these issues, we introduce Federated Averaging with Feature Normalization Update (FedFN), a straightforward learning method. We demonstrate the superior performance of FedFN through extensive experiments, even when applied to pretrained ResNet18. Subsequently, we confirm the applicability of FedFN to foundation models.

## 1 Introduction

Federated Learning (FL) facilitates collaborative model training while preserving data privacy [16, 24]. This approach consists of four iterative stages: (1) client selection, (2) broadcasting, (3) local training, and (4) aggregation. Selected clients receive the global model, train it locally with their own data, and transmit the trained models back to the central server for aggregation. These steps are repeated to progressively enhance the performance of global model. In FL, a fundamental challenge arises from the existence of diverse data distributions among different clients, referred to as *data heterogeneity*. This leads to performance degradation of the global model [18, 19, 24].

Recently, numerous studies [4, 27, 30] have been conducted to identify the specific aspects, such as feature representations or classifier weights, that are significantly influenced by data heterogeneity. Luo et al. [23] demonstrated that classifier weights are the most sensitive to data heterogeneity, illustrating how classifiers can easily become biased depending on the data distribution. To mitigate classifier bias, algorithms have been proposed using restricted softmax loss or fixed orthogonal classifiers during local training [4, 19, 27]. Meanwhile, Shi et al. [30] demonstrates the impact of data heterogeneity on feature representations, potentially leading to dimensional collapse where only a subset of high-dimensional feature vectors is employed to represent features within the global model. In our study, we find that the primary concern lies not in the classifier weight but in the features.

Feature normalization, as utilized in various fields [4, 6, 13, 15, 25, 29, 34], enhances the discriminative power of feature representation, making it easier to distinguish data belonging to different

*corresponding authors

International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023 (FL@FM-NeurIPS'23). This workshop does not have official proceedings and this paper is non-archival.

classes. We reveal that data heterogeneity in FL leads to a substantial discrepancy in feature norms between the global model and local models. Based on this observation, we incorporate feature normalization into the FL framework. Our contributions are outlined as follows:

- In FedAVG, we find that feature representations are more adversely affected by data heterogeneity than classifier weights. Furthermore, as data heterogeneity increases, the disparity between the higher feature norms for observed classes, derived from local models, and the feature norms of unobserved classes widens, in contrast to classifier weight norms. This widening gap extends to encompass feature norm disparities between local models and the global model. **(Section 3)**
- To tackle this challenge, we introduce **Fed**erated Averaging with **F**eature **N**ormalization Update (FedFN), which effectively eliminates discrepancies in feature norms during local training. FedFN robustly maintains the quality of feature representations even in highly heterogeneous data settings. **(Section 4)**
- We incorporate the feature normalization technique into existing algorithms, and show notable performance improvements. Furthermore, this effectiveness persists even when using pretrained model. **(Section 5)**

## 2 Experimental Setup

**Datasets and Models** We conduct extensive experiments using two widely-used datasets with suitable models: VGG11 [31] and ResNet18 [7] for the CIFAR-10 dataset, and MobileNet [8] for the CIFAR-100 dataset [14].

**Federated Environments** To simulate a realistic FL scenario, we set the number of clients ($N$) to 100 and a fraction ratio ($r$) of 0.1 for each round of communication. Note that our investigation primarily addresses a balanced environment, wherein all clients possess datasets of identical size.[2] To create data heterogeneity, we adopt a sharding partition strategy, as used in prior studies [24, 27].This strategy divides the dataset $D$ into $N \times s$ non-overlapping shards. Each client is allocated $s$ shards, with each shard containing $\frac{|D|}{N \times s}$ samples from a single class. Consequently, each client can hold samples from up to $s$ distinct classes, implying that a smaller value of $s$ leads to greater data heterogeneity.

**Implementation Details** Details regarding the code implementation can be found in the Appendix B. All experiments involve 320 communication rounds. To optimize the initial learning rate ($\eta$) and the number of local epochs ($E$) on CIFAR-10 and CIFAR-100, we conduct grid searches, with results in Appendix C. $\eta$ is explored in the range of $\{0.01, 0.03, 0.05, 0.1\}$ for CIFAR-10 and $\{0.1, 0.3, 0.5, 1.0\}$ for CIFAR-100. We evaluate $E$ in the set $\{1, 5, 10, 15, 20\}$ for both datasets and find optimal values of 15 for CIFAR-10 and 5 for CIFAR-100. When specific values are not mentioned, default initial learning rates of 0.01 for CIFAR-10 and 0.1 for CIFAR-100 are used.

## 3 Heterogeneity in FedAVG: The Devil is in Feature Norm Discrepancy

### 3.1 4-Factor Analysis of FedAVG

Within the FedAVG, we explore the impact of data heterogeneity on both feature representations at the penultimate layer and classifier weights, denoted as $f(\cdot; \theta_{ext}) \in \mathbb{R}^d$ and $\theta_{cls} \in \mathbb{R}^{C \times d}$, respectively. Our investigation centers on four factors often used to assess model performance [11, 28]:

- **(Factor 1) Weight similarity** quantifies the cosine similarity among classes in $\theta_{cls}$ (i.e., rows of $\theta_{cls}$). Lower values are preferred in this context.
- **(Factor 2) Inter-class similarity** calculates the cosine similarity among feature prototypes. A feature prototype for a class $c$ is defined as $\frac{1}{|D_{test}(c)|} \sum_{(x,y) \in D_{test}(c)} f(x; \theta_{ext})$, with a preference for lower values. Here, $D_{test}(c)$ represents the test dataset containing only samples from class $c$, and $(x, y)$ denotes (input image, true class label of $x$).
- **(Factor 3) Intra-class similarity** quantifies the averaged cosine similarity between feature prototype and features for each class. Higher values are preferred here.
- **(Factor 4) Prototype-weight alignment** measures the cosine similarity between feature prototype and classifier weight for each class. Higher values are also preferred in this case.

[2] Appendix D reports the results under unbalanced and non-IID derived Dirichlet distributions.

These four factors encompass both feature and classifier-related aspects, enabling us to discern which aspects are more negatively impacted as data heterogeneity increases.

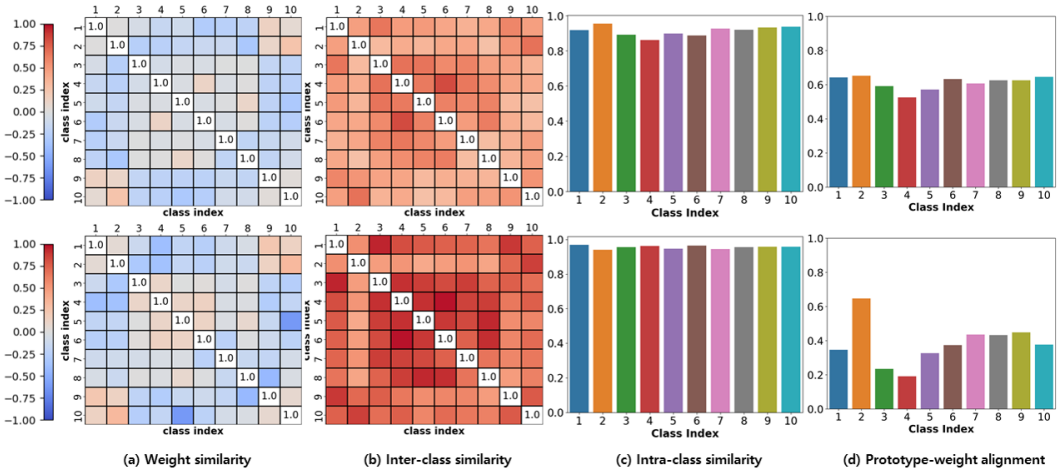
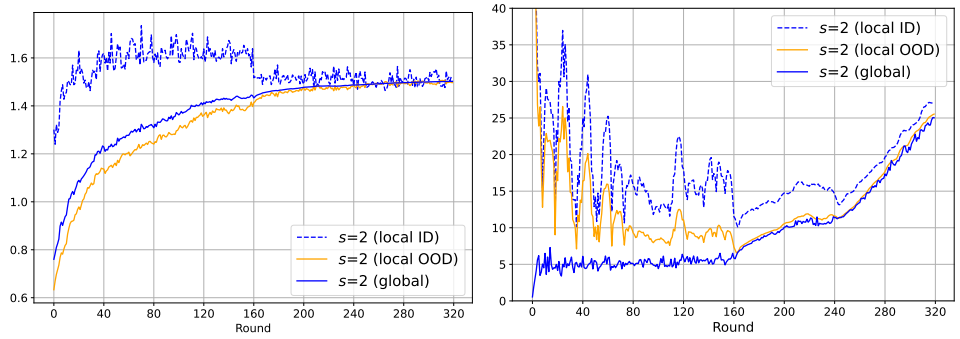



Figure 1: 4-Factor Analysis of FedAVG: Results for $s = 10$ (Top Row) and $s = 2$ (Bottom Row).

Figure 1 visualizes the four factors concerning data heterogeneity. The upper and lower rows present the results under smaller (i.e., $s$=10) and larger data heterogeneity (i.e., $s$=2), respectively. In both settings, weight similarity exhibits lower values, as indicated by the blue color. However, with increasing data heterogeneity, there is an increase in inter-class similarity within FedAVG, indicating a negative impact. Conversely, as data heterogeneity rises, intra-class similarity improves. With higher data heterogeneity, prototype-weight alignment deteriorates, likely influenced by the more pronounced decrease in inter-class similarity. In summary, as data heterogeneity increases, the factors most adversely affected are inter-class similarity and prototype-weight alignment, both of which are common feature-related factors.

### 3.2 Feature Norm Discrepancy Persists in Local and Global Models




(a) Classifier weight norm means (b) Feature norm means

Figure 2: Means of Classifier Weight and Feature Norms for Local and Global Models with $s$=2.

We take a closer look at feature norm bias alongside weight norm bias [11, 23, 27, 37, 40, 42], which tends to favor major classes with larger weight norms. Our investigation is conducted in a high data heterogeneity setting ($s$=2) using the CIFAR-10 dataset on VGG11. We are motivated by the extensive distribution of calssifier weight norms within classes observed across clients in FL, as discussed in [23]. Furthermore, our prior four factor analysis has emphasized the significant influence of feature-related aspects in response to varying data heterogeneity. This analysis strongly encourages us to look into feature norm bias. This is important because even though a lot of research has been done on weight norm bias, not much attention has been given to feature norm bias.

We compute weight norm means for two groups of classes: those seen (ID) and unseen (OOD) classes during their respective local training. Additionally, we calculate the total class weight norm mean from the global model. Furthermore, we explore feature norm means derived from local models using both the ID and OOD test datasets, in addition to the feature norm mean obtained

from the global model using the entire test dataset. Figure 2 illustrates the visual representation of the results.

During the initial stage of training, weight norms in local models exhibit a significant bias in favor of ID classes over OOD classes. Simultaneously, feature norm mean within local model also display a corresponding bias. However, as the learning rate gradually decreases, both feature norm bias and weight norm bias diminish. Weight norm bias eventually vanishes in local models, aligning with the weight norm mean of global model. In contrast, feature norm bias persists, consistently resulting in higher feature norm mean in local model from ID classes compared to the global model.

### 3.3 Feature Norm Discrepancy with Increasing Heterogeneity

We examine the discrepancy of feature norm means between local models on ID test dataset and global model on the entire test dataset under varying data heterogeneity. Specifically, we consider $s \in \{2, 5, 10\}$ and IID (Exactly class balanced data distribution across clients) on the CIFAR-10 dataset. As depicted in Figure 3, the discrepancy in the norms between the global model and local models increases, as data heterogeneity increases (i.e., IID $\rightarrow$ $s$=10 $\rightarrow$ $s$=5 $\rightarrow$ $s$=2).

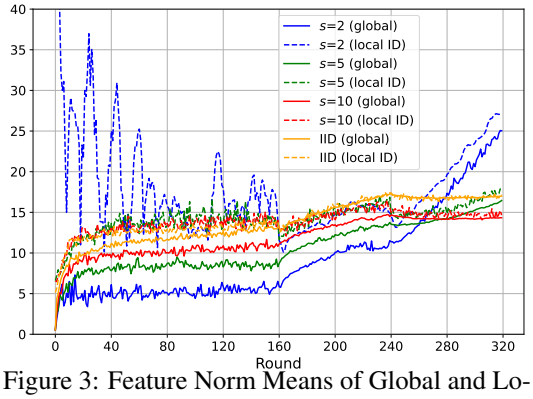
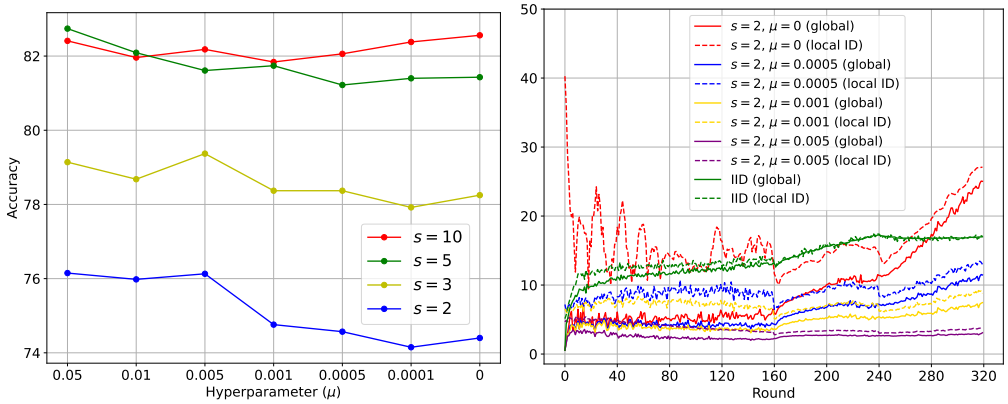



Figure 3: Feature Norm Means of Global and Local Models across Varying Data Heterogeneity.

### 3.4 Reducing Feature Norm Discrepancy for Improved Performance




(a) Accuracy on hyperparameter $\mu$

(b) Feature norm means of local and global models

Figure 4: Effects of Feature Norm Regularization.

To mitigate feature norm discrepancy between local and global models, especially in scenarios of high heterogeneity that lead to elevated local model feature norms, we introduce an L2 feature norm regularization term with a hyperparameter $\mu$ to the local model training loss. This term is added alongside the cross-entropy loss $\mathcal{L}_{CE}$, and the combined loss $\mathcal{L}_{\mu}$ is formulated as follows:

$$\mathcal{L}_{\mu}(x;\theta) = \mathcal{L}_{CE}(x;\theta) + \mu\,||f(x;\theta_{ext})||_2. \quad (1)$$

We apply $\mathcal{L}_{\mu}$ with various hyperparameters $\mu \in \{0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}$ across different $s \in \{2, 3, 5, 10\}$ settings, and the results are illustrated in Figure 4.

The optimal hyperparameter $\mu$ varies across different heterogeneity settings. For example, in Figure 4 (a) , we observe that for $s = 5$ and $s = 2$, the optimal $\mu$ values are 0.05 and 0.005, respectively. In Figure 4 (b), we illustrate the feature norm means of local and global models during training under high heterogeneity setting ($s$=2) for different $\mu$ values. Notably, at the optimal $\mu$ value of 0.005, we observe a significant reduction in these discrepancies, aligning more closely with the ideal IID setting. Moreover, each feature norm mean of local and global models are noticeably lower than those in the ideal IID setting. This difference can be attributed to the decrease in the feature norm mean of local model, which subsequently impacts the feature norm mean of the global model. In summary, our findings underscore the performance improvements can be achieved by reducing discrepancies in feature norm means between the global and local models during the training process.

# 4 FedFN: Federated Averaging with Feature Normalization Update

In this section, we present Federated Averaging with Feature Normalization (FedFN), which integrates feature normalization (FN) with FedAVG. We also apply the four-factor analysis conducted in Section 3 to FedFN. Furthermore, the gradual reduction in weight norm bias within FedFN during training is detailed in Appendix D.

## 4.1 FedFN Algorithm

We revisit *Federated Averaging with Feature Normalization Update* (FedFN), an extension of FedAVG enriched with Feature Normalization (FN) updates as discussed in [4][3]. FN eliminates feature norm bias within the local model by normalizing the feature vector, ensuring that the norm is consistently set to 1 for any input $x$. In FedFN, compared to FedAVG, the FN update modifies the logit vector $z$ of an input $x$, represented as $\hat{z}(x;\theta) = \theta_{cls}\frac{f(x;\theta_{ext})}{||f(x;\theta_{ext})||_2}$. Consequently, the gradient of $\theta_{\text{cls}}$ concerning the cross-entropy loss $\mathcal{L}_{CE}(\cdot)$ for FedFN is expressed as follows:

$$\nabla_{\theta_{cls}}\mathcal{L}_{CE}(x;\theta) = \nabla_{\hat{z}(x;\theta)}\mathcal{L}_{CE}(x;\theta)\frac{f(x;\theta_{ext})^\top}{||f(x;\theta_{ext})||_2} \in \mathbb{R}^{C\times d}.$$

Unlike FedAVG, FedFN scales the gradient of $\theta_{cls}$ by dividing it by the feature vector norm. This scaling significantly impacts the gradient of $\theta_{cls}$ and, consequently, the applied learning rate. As a result of this influence, we conduct a thorough fine-tuning for the learning rate for the FN update, leading FedFN to adopt a larger initial learning rate of 0.03, compared to the baseline rate of 0.01 in FedAVG. These learning rates undergo careful selection through an extensive grid search, with detailed findings available in Appendix C. Table 1 demonstrates significant accuracy improvement with FedFN compared to FedAVG.

Table 1: Accuracy of FedAVG and FedFN.

| | FedAVG | FedFN |
|---|---|---|
| $s$=10 | 81.97 | **83.80** |
| $s$=2 | 74.24 | **77.77** |

## 4.2 4-Factor Analysis of FedFN

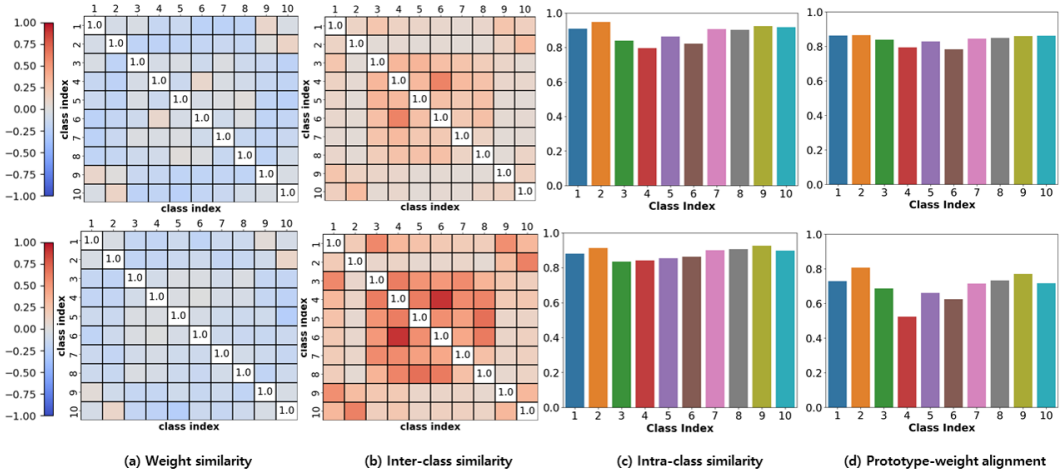



Figure 5: 4-Factor Analysis of FedFN; Results for $s$=10 (Top Row) and $s$=2 (Bottom Row).

We conduct four factor analysis on the improved global model compared to FedAVG, as reported in Table 1. The results are visualized in Figure 5. Similar to FedAVG, weight similarity remains robust even in high data heterogeneity. On the other hand, FedFN demonstrates significant improvement in inter-class similarity between prototypes compared to FedAVG, especially in the high data heterogeneity. In contrast to FedAVG, FedFN exhibits a slight decrease in intra-class similarity as data heterogeneity increases. This shift likely be attributed to the simultaneous improvement in inter-class similarity within FedFN, enabling finer class separation. While some degradation in

[3] SphereFed [4] proposes an approach to FL where the classifier is initialized in an orthonormalized manner and kept frozen. Meanwhile, feature normalization is applied to train the local model, utilizing MSE loss. The comparison between FedFN and SphereFed can be found in Appendix D.

prototype-weight alignment is observed in FedFN, FedFN consistently outperforms FedAVG, possibly due to its enhanced inter-class similarity. In summary, the superior inter-class similarity of FedFN contributes to its enhanced performance, resulting in improved discrimination and stability across various data heterogeneity.

# 5 Experiment Results

In this section, we present the experimental results of FedFN, demonstrating its compatibility with existing FL algorithms. Moreover, we investigate a comparative analysis with the pretrained model To assess the feasibility of implementing FedFN in the foundation models. Additionally, the application of FedFN for the personalized FL can be found in Appendix D.

## 5.1 Compatibility of FN with Existing FL Algorithms

We assess the compatibility of the FN update module with existing FL algorithms, including FedAVG [24], Scaffold [12], and FedEXP [10] (referred to as "Baseline" algorithms). Additionally, we compare the results with those obtained by applying the BABU module [27], which freezes the classifier part of the model, to the baseline algorithms. Table 2 summarizes the accuracy comparison for different FL algorithms, including baseline, +BABU, and +FN, on VGG11 for CIFAR-10 and MobileNet for CIFAR-100. For CIFAR-10 and CIFAR-100, we use FN updates with initial learning rates of 0.03 and 0.5, respectively. While Scaffold generally demonstrates superior performance, it faces training failures in scenarios with high data heterogeneity, even when applying the BABU module (e.g., $s$=3). In contrast, the FN update consistently outperforms all algorithms, including baselines and those with the BABU module, across all heterogeneity settings, demonstrating its superiority.

Table 2: FL Accuracy Comparison for Baseline, +BABU, and +FN.

| Algorithm | Module | VGG11 on CIFAR-10 | | | | MobileNet on CIFAR-100 | | |
|---|---|---|---|---|---|---|---|---|
| | | $s$=2 | $s$=3 | $s$=5 | $s$=10 | $s$=10 | $s$=50 | $s$=100 |
| FedAVG (2017) | Baseline | 73.62 | 77.70 | 81.62 | 82.13 | 37.25 | 42.90 | 43.36 |
| | + BABU | 73.11 | 76.57 | 81.22 | 81.89 | 43.20 | 39.70 | 39.59 |
| | **+ FN** | **76.47** | **78.66** | **82.07** | **83.09** | **44.67** | **48.17** | **49.67** |
| Scaffold (2020) | Baseline | 77.07 | (*Failed*) | **84.30** | 84.98 | 41.86 | 43.59 | 41.74 |
| | + BABU | 77.17 | (*Failed*) | 83.54 | 84.43 | 46.41 | 41.61 | 42.55 |
| | **+ FN** | **77.96** | **79.24** | **84.40** | **85.54** | **49.42** | **50.42** | **52.10** |
| FedEXP (2023) | Baseline | 73.49 | 77.90 | 81.64 | 82.42 | 36.35 | 41.06 | 42.38 |
| | +BABU | 72.58 | 77.59 | 81.07 | 81.96 | 43.38 | 40.73 | 39.04 |
| | **+ FN** | **76.33** | **78.20** | **82.41** | **83.26** | **45.90** | **49.10** | **49.11** |

## 5.2 Comparative Analysis with Pretrained Model

To evaluate the feasibility of applying FedFN within the foundation models, we conducted experiments using a pretrained ResNet18 model on the CIFAR-10 dataset [2, 26, 33, 41]. Specifically, we compare the performance of FedAVG, FedBABU, and FedFN when utilizing both pretrained and non-pretrained ResNet18 models. Table 3 presents the accuracy comparison for these algorithms. Across all experimental settings, FedFN consistently outperforms both FedBABU and FedAVG. Notably, FedAVG and FedBABU, particularly in scenarios with high data heterogeneity, exhibit significant performance declines when pretrained models are employed. In contrast, FedFN consistently improves with the application of pretrained models, utilizing FN updates with an initial learning rate of 0.1.

Table 3: Accuracy Comparison on CIFAR-10 on ResNet18.

| Algorithm | Pretrained=*False* | | | | Pretrained=*True* | | | |
|---|---|---|---|---|---|---|---|---|
| | $s$=2 | $s$=3 | $s$=5 | $s$=10 | $s$=2 | $s$=3 | $s$=5 | $s$=10 |
| FedAVG | 41.50 | 55.31 | 67.64 | 73.39 | 37.87 | 58.31 | 71.75 | 84.50 |
| FedBABU | 49.21 | 58.44 | 68.84 | 73.82 | 49.78 | 49.61 | 66.46 | 84.22 |
| FedFN | **55.17** | **60.47** | **77.12** | **81.26** | **56.84** | **76.84** | **80.02** | **84.99** |

## 6 Conclusion

In this study, we observe that increasing data heterogeneity leads to larger feature norm disparities between global and local models, which are influenced by feature norm bias within local models. We address this issue by introducing feature normalization techniques. Extensive experiments across various FL confirm the superior performance of feature normalization, emphasizing its role in enhancing the feature representations. Our experiments showcase the exceptional performance of FedFN, extending its effectiveness to pretrained ResNet18 models and confirming its applicability to foundational models.

## Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning, 90%] and [No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 10%]. We thank Minchan Jeong for the invaluable discussions and experimental support.

## References


[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[2] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han-Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. *URL https://arxiv.org/abs/2206.11488*, 2022.

[3] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[4] Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. Spherefed: Hyperspherical federated learning. In *European Conference on Computer Vision*, pages 165–184. Springer, 2022.

[5] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[6] Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. Deepvisage: Making face recognition simple yet with powerful generalization skills. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1682–1691, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] Chenxi Huang, Liang Xie, Yibo Yang, Wenxiao Wang, Binbin Lin, and Deng Cai. Neural collapse inspired federated learning with non-iid data, 2023.

[10] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. *arXiv preprint arXiv:2301.09604*, 2023.

[11] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[12] 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*, pages 5132–5143. PMLR, 2020.

[13] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009.

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

[16] 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.

[17] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[18] 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.

[19] Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 995–1005, 2021.

[20] Xin-Chun Li, Yi-Chu Xu, Shaoming Song, Bingshuai Li, Yinchuan Li, Yunfeng Shao, and De-Chuan Zhan. Federated learning with position-aware neurons. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10082–10091, 2022.

[21] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. *arXiv preprint arXiv:2303.10058*, 2023.

[22] 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.

[23] 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.

[24] 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*, pages 1273–1282. PMLR, 2017.

[25] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019.

[26] J Nguyen, J Wang, K Malik, M Sanjabi, and M Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2210.08090*, 2022.

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

[28] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[29] Marios Savvides, Dipan Kumar Pal, and Yutong Zheng. Convex feature normalization for face recognition, February 4 2021. US Patent App. 16/968,040.

[30] Yujun Shi, Jian Liang, Wenqing Zhang, Vincent YF Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. *arXiv preprint arXiv:2210.00226*, 2022.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8432–8440, 2022.

[33] Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in Neural Information Processing Systems*, 35:19332–19344, 2022.

[34] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[35] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[36] 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, 2020.

[37] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019.

[38] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023.

[39] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Fed2: Feature-aligned federated learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2066–2074, 2021.

[40] Haiyang Yu, Ningyu Zhang, Shumin Deng, Zonggang Yuan, Yantao Jia, and Huajun Chen. The devil is the classifier: Investigating long tail relation classification with decoupling analysis. *arXiv preprint arXiv:2009.07022*, 2020.

[41] Sixing Yu, J Pablo Muñoz, and Ali Jannesari. Federated foundation models: Privacy-preserving and collaborative learning for large models. *arXiv preprint arXiv:2305.11414*, 2023.

[42] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020.

[43] 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 –

# FedFN: Feature Normalization for Alleviating Data Heterogeneity in Federated Learning

## A Related Work

### A.1 Federated Learing

**Global Federated Learning** Global Federated Learning (GFL) aims to enhance the performance of a single global model across decentralized clients by addressing data heterogeneity arising from diverse user behaviors. Researchers have explored various methodologies within GFL to create robust models for diverse devices and data sources. These approaches include client drift mitigation [10, 12, 16], aggregation schemes to improve model fusion mechanisms at the server [35, 36], and data sharing techniques introducing public datasets or synthesized data to achieve a more balanced data distribution [22, 23, 43].

**Personalized Federated Learning** Personalized Federated Learning (PFL) focuses on training personalized models for individual clients, adapting to their specific data distributions and tasks. PFL methodologies include decoupling methods that separate the feature extractor and classifier during communication, enabling unique updates for the data distribution of each client [1, 3, 27], modifying local loss functions to improve task performance [5, 17], and utilizing prototype communication techniques [32, 38].

### A.2 Partial Model Updates in FL

**Debiasing classifier in FL** Efforts to address data heterogeneity in both GFL and PFL domains have explored differential updates within the model parameters, with a particular focus on the classifier part. For instance, Luo et al. [23] propose classifier post-calibration with virtual features to tackle a notable bias among classifiers of different local models. Li and Zhan [19] introduce the restricted softmax loss for local updates to prevent classifiers from becoming inaccurate when updating for missing classes. Additionally, certain studies [4, 9, 21, 27] suggest the use of fixed classifiers constructed from orthogonal basis vectors during training.

**Feature Enhancement in FL** Recent research [15, 20, 30, 39] in the context of GFL has focused on aligning feature representations among local models. For instance, Li et al. [15] introduce the contrastive loss during local iteration to improve feature alignment. Additionally, Shi et al. [30] highlights the issue of data heterogeneity leading to severe dimensional collapse in the global model, resulting in representations tending towards lower dimensions. To address this, they propose using a regularization term during local training to mitigate the issue and improve feature alignment. Moreover, some studies [32, 38] have explored communication strategies involving feature prototypes to further enhance feature alignment in the context of PFL.

### A.3 Feature Normalized Model in DL

Feature normalized model has been widely adapted in various fields of deep learning (DL), including face recognition [29, 34], regression [25], and federated learning [4], with the aim of enhancing the discriminative power of features. In the several studies [4, 34], both the feature vectors and classifier weights are normalized to enforce cosine-similarity element logits, restricting the values of each element in the logit vector.

# B Preliminaries

In the upcoming section, we elucidate the concept of FL, followed by an in-depth discussion of the experimental setup, encompassing dataset descriptions, model specifications, hyperparameter configurations during FL, and detailed description four factor analysis conducted in main paragraph. For clarity and convenience, we present a concise overview of key notations in Table 1, facilitating comprehension of the paper.

Table 1: Main Notations Throughout the Paper.

| | |
|---|---|
| **Indices** | |
| $c \in [C]$ | index for a class |
| $r \in [R]$ | index for FL round |
| $n \in [N]$ | index for a client |
| **Dataset** | |
| $(x, y)$ | (input image , true class label of $x$) |
| $D_{train}, D_{test}$ | total train and test dataset |
| $D^n_{train}, D^n_{test}$ | train and test dataset of client $n$ |
| $D(c)$ | collection of dataset $D$ with the label $c$ |
| **Parameters** | |
| $\theta := (\theta_{ext}, \theta_{cls})$ | model parameter |
| $\theta_{ext}$ | feature extractor part of $\theta$ |
| $\theta_{cls} \in \mathbb{R}^{C \times d}$ | classifier part of $\theta$ |
| $\theta_{cls,i}, i \in [C]$ | i-th row vector of $\theta_{cls}$ |
| **Model Forward** | |
| $p(x;\theta) \in \mathbb{R}^C$ | softmax probability of input $x$ |
| $p_i(x;\theta), i \in [C]$ | i-th element of $p(x;\theta)$ |
| $\mathcal{L}_{CE}(x;\theta) := -\log p_y(x;\theta)$ | cross entropy loss of input $x$ |
| $f(x;\theta_{ext}) \in \mathbb{R}^d$ | feature vector of input $x$ |
| $\hat{f}(x;\theta_{ext}) := f(x;\theta_{ext})/\|\|f(x;\theta_{ext})\|\|_2$ | normalized feature vector of input $x$ |
| $z(x;\theta) := \theta_{cls}\, f(x;\theta_{ext}) \in \mathbb{R}^C$ | logit vector of input $x$ |
| $z_i(x;\theta), i \in [C]$ | i-th element of $z(x;\theta)$ |
| **Prototype of label c** | |
| $f(D(c);\theta_{ext}) := \frac{1}{\|D(c)\|} \sum_{(x,y)\in D(c)} f(x;\theta_{ext})$ | prototype of $D(c)$ |
| $\hat{f}(D(c);\theta_{ext}) := \frac{1}{\|D(c)\|} \sum_{(x,y)\in D(c)} \hat{f}(x;\theta_{ext})$ | prototype from the normalized features |

## B.1 FL Procedure

In FL, we aim to train a robust image classification model on the central server while preserving the privacy and security of individual client data. The procedure involves communication over $R$ rounds. In each round $r \in [R]$, a random subset of clients $S_r \subset [N]$ is selected from the client pool. These selected clients receive the current global model parameters $\theta^{r-1}$ from the central server. Subsequently, each client $n \in S_r$ performs local updates on its local dataset $D^n_{train}$ for $E$ epochs using a batch size of $B$. The updated model parameters for client $n$ are denoted as $\theta^{r,n}$. Afterward, the central server updates the global model parameter $\theta^r$ as a result of convex combination based on the $\theta^{r,n}$ [18]. This collaborative approach iteratively refines the image classification model on the central server over the $R$ rounds, resulting in a robust model that performs well on the entire test dataset $D_{test}$ while preserving individual client data privacy.

## B.2 Experimental Setup

**Datasets and Models** To simulate a realistic federated learning scenario involving 100 clients, we conduct extensive studies on two widely-used datasets: CIFAR-10 and CIFAR-100 [14]. For CIFAR-10, we employ the VGG11 [31] model, while for CIFAR-100, the MobileNet [8] model is chosen. The training data is distributed among 100 clients using two distinct Non-IID partition strategies:

- **Sharding** [24, 27]: We meticulously organize the data by label and divide it into non-overlapping shards of equal size. Each shard encompasses $\frac{|D_{train}|}{100 \times s}$ samples of the same class, and $s$ denotes the number of shards per client. This results in each client having access to a maximum of $s$ different classes. As we decrease the number of shards per

user $s$, the level of data heterogeneity among clients increases. For CIFAR-10, we explore various $s$ values, such as $s \in \{2, 3, 5, 10\}$, while for CIFAR-100, we experiment with $s \in \{10, 50, 100\}$.

- **Latent Dirichlet Allocation (LDA)** [23, 35]: We utilize the LDA technique to sample a probability vector $p_c = (p_{c,1}, p_{c,2}, \cdots, p_{c,100}) \sim Dir(\alpha)$ and allocate a proportion $p_{c,k}$ of instances of class $c \in [C]$ to each client $k \in [100]$, where $Dir(\alpha)$ represents the Dirichlet distribution with the concentration parameter $\alpha$. The parameter $\alpha$ controls the strength of data heterogeneity, where smaller values lead to stronger heterogeneity among clients. For both CIFAR-10 and CIFAR-100, we conduct experiments with various $\alpha$ values, such as $\alpha \in \{0.1, 0.3, 0.5, 1.0\}$.

**Hyperparameter Search:** To optimize the hyperparameters for federated learning, we conduct grid searches for the initial learning rate on both CIFAR-10 and CIFAR-100. For CIFAR-10, we explore learning rates in the range of $\{0.01, 0.03, 0.05, 0.1\}$, while for CIFAR-100, we consider learning rates of $\{0.1, 0.3, 0.5, 1.0\}$. Additionally, we perform grid searches to determine the optimal number of local epochs, evaluating values in the set $\{1, 5, 10, 15, 20\}$ for both datasets. The optimal number of local epochs is found to be 15 for CIFAR-10 and 5 for CIFAR-100. In cases where specific values are not mentioned, we use default initial learning rates of 0.01 for CIFAR-10 and 0.1 for CIFAR-100.

**Implementation Details** All experiments are conducted for 320 rounds to thoroughly assess the performance and convergence behavior of the models. To ensure convergence during training, we decay the learning rate by 0.1 at half and three-quarters of the federated learning rounds. Additionally, we utilize random horizontal flipping as a data augmentation technique throughout the training process. Table 2 in Section 5 and Table 8 in Appendix D are constructed using the code structure from `https://github.com/Lee-Gihun/FedNTD`, while the rest of the implementations are based on `https://github.com/jhoon-oh/FedBABU`.

### B.3 4-Factor Analysis

We introduce four factors, extensively studied [11, 28] and aim to identify which of them have a high negative impact in the presence of data heterogeneity.

**(i) Weight similarity:** Measuring the similarity or dissimilarity among classifiers across classes, this factor computes the cosine similarity between their normalized weight vectors, resulting in a symmetric matrix. The detail form is :

$$N(\theta_{cls})^\top N(\theta_{cls}) \in [-1, 1]^{C \times C}, \text{where } N(\theta_{cls}) = \left[ \frac{\theta_{cls,1}^\top}{||\theta_{cls,1}||_2} | \cdots, \frac{\theta_{cls,C}^\top}{||\theta_{cls,C}||_2} \right] \in \mathbb{R}^{d \times C}.$$

Lower weight similarity is preferred, indicating distinct classifiers for each class and better discrimination.

**(ii) Inter-class similarity:** This factor delves into the relationships between feature prototypes representing different classes, represented by a symmetric matrix $f(D_{test}; \theta)^\top f(D_{test}; \theta) \in [-1, 1]^{C \times C}$. Here, $f(D_{test}; \theta)$ is represented as

$$[f(D_{test}(1); \theta)| \cdots | f(D_{test}(C); \theta)] \in \mathbb{R}^{d \times C}.$$

Lower inter-class similarity is desired, representing distinguishable feature vectors for different classes.

**(iii) Intra-class similarity:** This factor provides insights into the diversity or similarity of representations within individual classes. We achieve this by calculating the cosine similarity between feature vectors of test prototypes belonging to the same class. For each class $c \in [C]$, it is evaluated by:

$$\frac{1}{|D_{test}(c)|} \sum_{(x,y) \in D_{test}(c)} \frac{f(x; \theta_{ext})^\top}{||f(x; \theta_{ext})||_2} \frac{f(D_{test}(c); \theta_{ext})}{||f(D_{test}(c); \theta_{ext})||_2} \in \mathbb{R}.$$

Higher intra-class similarity is preferred, indicating that feature vectors belonging to the same class are closer to each other, thereby improving class representation and classification performance.

**(iv) Prototype-weight alignment:** Assessing this factor reveals the degree of alignment between the classifier and prototypes for each class, according to their internal product. High alignment signifies a strong match, while low alignment indicates a potential mismatch or poor fit. For each class $c \in [C]$, it is evaluated by inner product of $\theta_{cls,c}$ and $f(D_{test}(c); \theta)$. This factor is generally not a primary concern but may be correlated with weight similarity and inter-class similarity.

# C Grid Search Result

To optimize the hyperparameters for FL, we conduct grid searches for the initial learning rate on both CIFAR-10 and CIFAR-100. For CIFAR-10, we explore learning rate $\eta$ in the range of $\{0.01, 0.03, 0.05, 0.1\}$, while for CIFAR-100, we consider $\eta$ of $\{0.1, 0.3, 0.5, 1.0\}$. Additionally, we performe grid searches to determine the optimal number of local epochs $E$, evaluating values in the set $\{1, 5, 10, 15, 20\}$ for both datasets. The optimal number of $E$ is found to be 15 for CIFAR-10 and 5 for CIFAR-100. Unless otherwise specified, the values determined through grid search for the hyperparameter $\eta$ are as follows: default initial learning rates are 0.01 for CIFAR-10 and 0.1 for CIFAR-100.

## C.1 FedBABU vs FedAVG vs FedFN

We conduct grid searches for FedAVG [24], FedBABU [27], and FedFN. Table 2 to 5 present the grid search results for VGG on CIFAR-10, MobileNet on CIFAR-100, ResNet18 on CIFAR-10, and Pretrained ResNet18 on CIFAR-10, respectively. In these tables, we abbreviate FedAVG, FedBABU, and FedFN as AVG, BABU, and FN, respectively, and indicate cases where the final global model fails to converge during training with a "-" in the respective table cells. The determined optimal hyperparameter $\eta$ values for FedFN are as follows: 0.03, 0.5, 0.1, and 0.1 for Table 2 to 5, respectively.

Table 2: Grid Search Results for VGG11 on CIFAR-10.

| *η=0.01* | *E*=1 | | | *E*=5 | | | *E*=10 | | | *E*=15 | | | *E*=20 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| *s*=10 | 49.51 | 52.98 | 52.02 | 82.43 | 82.68 | 82.82 | 82.56 | 83.03 | 82.7 | **82.16** | **81.97** | 82.11 | 81.82 | 81.66 | 82.42 |
| *s*=5 | 42.47 | 46.28 | 50.26 | 79.39 | 80.42 | 79.76 | 81.13 | 81.36 | 82.09 | **81.04** | **81.08** | 81.74 | 80.82 | 81.35 | 81.3 |
| *s*=3 | 29.45 | 38.32 | 48.24 | 71.64 | 73.83 | 72.94 | 78.39 | 77.39 | 77.49 | **77.73** | **77.29** | 78.37 | 78.00 | 78.03 | 78.48 |
| *s*=2 | 24.21 | 30.61 | 46.24 | 56.37 | 63.78 | 62.5 | 73.31 | 73.24 | 73.79 | **75.05** | **74.24** | 75.34 | 75.25 | 74.98 | 76.33 |
| *η=0.03* | *E*=1 | | | *E*=5 | | | *E*=10 | | | *E*=15 | | | *E*=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| *s*=10 | 63.23 | 66.81 | 64.32 | 84.23 | 84.32 | 83.76 | 84.10 | 84.13 | 83.80 | 83.61 | 84.38 | **83.80** | 83.82 | 83.42 | 83.68 |
| *s*=5 | 54.86 | 59.21 | 57.94 | 82.14 | 82.56 | 81.43 | 82.19 | 82.39 | 82.34 | 83.17 | 82.92 | **82.43** | 82.49 | 82.66 | 82.79 |
| *s*=3 | 34.12 | 49.90 | 52.11 | 75.71 | - | 74.01 | - | - | 78.51 | - | - | **78.93** | - | - | 78.77 |
| *s*=2 | - | 34.80 | 46.54 | 50.80 | 62.80 | 66.60 | 74.27 | 75.08 | 76.26 | 74.57 | - | **77.77** | 77.36 | - | 77.67 |
| *η=0.05* | *E*=1 | | | *E*=5 | | | *E*=10 | | | *E*=15 | | | *E*=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| *s*=10 | 69.58 | 70.99 | 68.22 | 84.49 | 84.51 | 83.60 | 84.22 | 84.30 | 83.35 | 84.19 | 84.18 | 83.81 | 83.94 | - | 82.92 |
| *s*=5 | 56.65 | 62.67 | 59.31 | 81.74 | 81.62 | 81.87 | 82.77 | - | 82.57 | - | - | 82.63 | 82.90 | - | 82.52 |
| *s*=3 | 28.78 | 50.34 | 52.09 | - | - | 74.69 | - | - | 77.45 | - | - | 78.69 | - | - | 78.39 |
| *s*=2 | - | 31.77 | 40.34 | - | 61.66 | 62.52 | 71.52 | - | 71.6 | - | - | 76.06 | - | - | 76.77 |
| *η=0.1* | *E*=1 | | | *E*=5 | | | *E*=10 | | | *E*=15 | | | *E*=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| *s*=10 | 74.58 | 72.59 | 70.30 | 83.90 | - | 83.72 | 84.05 | - | 82.71 | - | - | 82.71 | - | - | 82.35 |
| *s*=5 | 60.35 | 61.88 | 59.01 | - | - | 80.08 | - | - | 81.79 | - | - | 81.54 | - | - | 81.77 |
| *s*=3 | 25.06 | 28.36 | 47.54 | - | - | 65.51 | - | - | 76.06 | - | - | 76.07 | - | - | 76.19 |
| *s*=2 | - | - | 41.27 | - | - | 58.43 | - | - | 70.86 | - | - | 73.10 | - | - | 74.90 |

Table 3: Grid Search Results for MobileNet on CIFAR-100.

| $\eta$=0.1 | $E$=1 | | | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=100 | 41.32 | 41.57 | 76.61 | **40.49** | **43.19** | 51.18 | 37.03 | 38.13 | 45.26 | 36.16 | 36.10 | 44.03 | 35.91 | 36.19 | 42.25 |
| $s$=50 | 40.86 | 38.35 | 7.51 | **40.76** | **40.63** | 51.11 | 36.79 | 37.77 | 46.74 | 36.10 | 37.54 | 44.47 | 34.93 | 36.05 | 43.61 |
| $s$=10 | 35.95 | 27.02 | 4.85 | **45.56** | **36.60** | 43.18 | 41.01 | 34.67 | 47.63 | 37.87 | 35.53 | 45.50 | 38.28 | 35.94 | 47.41 |
| $\eta$=0.3 | $E$=1 | | | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=100 | 46.65 | 41.76 | 19.52 | 40.57 | 46.67 | 50.80 | 37.79 | 41.28 | 44.34 | 36.16 | 38.29 | 41.18 | 35.91 | 38.22 | 39.38 |
| $s$=50 | 44.03 | 37.83 | 19.63 | 40.02 | 45.28 | 49.92 | 37.28 | 42.65 | 43.62 | 36.10 | 41.66 | 41.71 | 34.93 | 40.61 | 38.08 |
| $s$=10 | 37.37 | 26.66 | 16.66 | 47.07 | 33.78 | 48.89 | 44.57 | 36.42 | 46.77 | 37.87 | - | 47.95 | 38.28 | 35.12 | 48.39 |
| $\eta$=0.5 | $E$=1 | | | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=100 | 47.18 | 35.81 | 28.46 | 39.33 | - | **51.51** | 38.23 | - | 44.54 | 35.59 | - | 39.56 | 35.99 | - | 38.20 |
| $s$=50 | 47.53 | 36.28 | 25.71 | 40.42 | - | **50.6** | 37.71 | 44.76 | 42.47 | 37.66 | - | 40.96 | 38.38 | 43.15 | 38.42 |
| $s$=10 | 38.91 | 19.51 | 21.02 | 46.48 | 11.83 | **46.87** | 45.74 | 23.02 | 44.25 | 43.61 | 24.74 | 45.05 | 42.62 | 30.68 | 43.28 |
| $\eta$=1.0 | $E$=1 | | | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=100 | 46.92 | 31.15 | 36.07 | 39.61 | 34.66 | 50.95 | 37.12 | 42.06 | 42.17 | 37.83 | 46.53 | 39.64 | 37.64 | 33.12 | 39.97 |
| $s$=50 | 45.24 | 26.64 | 32.92 | 40.11 | 28.02 | 48.46 | 38.76 | 39.47 | 40.82 | 41.13 | 42.79 | 41.07 | 39.10 | 42.41 | 35.88 |
| $s$=10 | 38.59 | 8.39 | 26.63 | 45.14 | - | 39.64 | 44.03 | - | 38.64 | 40.51 | - | 36.05 | 39.67 | 12.26 | 37.41 |

Table 4: Grid Search Results for ResNet18 on CIFAR-10.

| $\eta$=0.01 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 75.32 | 76.07 | 78.00 | 74.53 | 74.17 | 77.72 | **73.55** | **73.85** | 77.80 | 73.16 | 72.32 | 77.18 |
| $s$=5 | 68.02 | 69.34 | 72.14 | 67.54 | 68.13 | 71.17 | **68.36** | **67.71** | 73.06 | 68.56 | 66.48 | 74.74 |
| $s$=3 | 58.60 | 58.93 | 64.94 | 57.87 | 58.45 | 66.45 | **57.07** | **57.38** | 64.23 | 60.93 | 55.13 | 66.58 |
| $s$=2 | 46.79 | 41.15 | 45.50 | 45.50 | 41.92 | 47.22 | **49.21** | **45.92** | 50.81 | 47.50 | 45.23 | 47.99 |
| $\eta$=0.03 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 78.64 | 78.77 | 80.63 | 77.76 | 77.73 | 79.86 | 76.83 | 76.57 | 79.34 | 76.02 | 76.87 | 78.78 |
| $s$=5 | 70.04 | 71.50 | 75.57 | 72.01 | 71.84 | 75.77 | 71.45 | 71.72 | 75.99 | 72.64 | 71.83 | 75.94 |
| $s$=3 | 57.53 | 61.54 | 67.55 | 56.60 | 59.58 | 66.88 | 58.06 | 60.53 | 68.63 | 60.80 | 62.07 | 68.65 |
| $s$=2 | 47.56 | 40.02 | 52.70 | 46.26 | 46.27 | 50.25 | 50.16 | 45.60 | 48.52 | 50.60 | 46.01 | 52.37 |
| $\eta$=0.05 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 80.17 | 79.41 | 81.21 | 78.22 | 77.93 | 80.67 | 77.87 | 78.50 | 80.98 | 77.16 | 77.77 | 79.59 |
| $s$=5 | 70.65 | 72.44 | 77.25 | 72.08 | 71.08 | 76.65 | 73.21 | 71.85 | 77.41 | 74.16 | 72.52 | 77.19 |
| $s$=3 | 56.26 | 62.22 | 66.91 | 56.22 | 61.38 | 67.01 | 59.22 | 60.85 | 64.09 | 61.79 | 57.00 | 64.51 |
| $s$=2 | 53.18 | 43.99 | 50.88 | 45.76 | 45.26 | 51.06 | 46.53 | 40.44 | 51.28 | 46.49 | 41.58 | 56.41 |
| $\eta$=0.1 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 80.83 | 80.13 | 82.37 | 80.04 | 79.87 | 82.01 | 79.50 | 78.57 | **81.19** | 78.64 | 78.21 | 80.37 |
| $s$=5 | 72.81 | 71.56 | 77.91 | 74.37 | 69.65 | 76.89 | 74.71 | 69.66 | **75.98** | 73.97 | 71.65 | 77.47 |
| $s$=3 | 57.14 | 59.65 | 63.75 | 57.33 | 52.71 | 61.12 | 60.34 | 58.13 | **65.09** | 58.46 | 54.72 | 67.08 |
| $s$=2 | 47.35 | 50.51 | 50.58 | 42.78 | 41.36 | 51.19 | 47.63 | 44.81 | **50.12** | 40.23 | 39.03 | 49.05 |

Table 5: Grid Search Results for Pretrained ResNet18 on CIFAR-10.

| $\eta$=0.01 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 83.85 | 83.80 | 84.65 | 84.06 | 84.29 | 84.20 | **84.02** | **83.96** | 84.43 | 84.43 | 84.33 | 84.81 |
| $s$=5 | 68.83 | 73.31 | 74.90 | 69.73 | 69.24 | 74.72 | **69.44** | **70.77** | 75.26 | 68.58 | 70.89 | 76.75 |
| $s$=3 | 54.21 | 58.52 | 63.87 | 51.85 | 48.52 | 61.76 | **48.96** | **57.85** | 60.72 | 47.84 | 51.11 | 62.06 |
| $s$=2 | 48.21 | 33.84 | 47.74 | 43.02 | 35.91 | 47.30 | **49.00** | **39.04** | 50.82 | 44.91 | 36.95 | 49.98 |
| $\eta$=0.03 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 84.86 | 82.73 | 85.28 | 85.02 | 82.00 | 85.07 | 84.85 | 82.71 | 85.10 | 84.82 | 83.04 | 85.27 |
| $s$=5 | 67.27 | 60.13 | 72.24 | 68.64 | 59.54 | 70.55 | 67.81 | 65.43 | 73.14 | 68.23 | 68.91 | 73.27 |
| $s$=3 | 53.48 | 47.72 | 54.61 | 53.74 | 50.32 | 56.27 | 54.18 | 53.78 | 59.92 | 56.35 | 53.82 | 60.28 |
| $s$=2 | 43.10 | 36.69 | 46.89 | 45.80 | 32.61 | 49.99 | 51.62 | 39.08 | 45.50 | 55.78 | 41.35 | 51.88 |
| $\eta$=0.05 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 86.06 | 71.37 | 85.84 | 85.71 | 75.94 | 85.52 | 85.31 | 76.50 | 84.73 | 85.41 | 77.77 | 85.11 |
| $s$=5 | 69.47 | 65.13 | 69.89 | 69.72 | 60.01 | 71.02 | 69.49 | 62.47 | 73.75 | 70.17 | 68.17 | 73.24 |
| $s$=3 | 57.91 | 52.70 | 53.10 | 55.63 | 49.75 | 54.86 | 62.72 | 54.51 | 63.03 | 65.22 | 52.00 | 65.71 |
| $s$=2 | 44.49 | 42.07 | 45.55 | 43.65 | 43.69 | 45.46 | 48.36 | 37.11 | 52.96 | 50.87 | 41.37 | 51.87 |
| $\eta$=0.1 | $E$=5 | | | $E$=10 | | | $E$=15 | | | $E$=20 | | |
| | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN | BABU | AVG | FN |
| $s$=10 | 85.89 | 72.78 | 86.02 | 85.67 | 74.95 | 85.65 | 84.98 | 78.00 | **85.45** | 85.25 | 78.73 | 85.20 |
| $s$=5 | 68.32 | 65.71 | 74.01 | 70.17 | 64.67 | 73.20 | 75.10 | 67.00 | **73.43** | 73.03 | 71.16 | 77.12 |
| $s$=3 | 62.57 | 46.52 | 66.26 | 64.15 | 59.18 | 74.36 | 63.56 | 56.59 | **73.85** | 66.52 | 59.25 | 73.86 |
| $s$=2 | 45.93 | 38.19 | 41.82 | 45.26 | 38.53 | 50.13 | 52.22 | 38.06 | **50.94** | 55.18 | 47.67 | 53.42 |

### C.2 SphereFed

We conduct grid searches for SphereFed (CE) and SphereFed (MSE) [4] for MobileNet on CIFAR-100. In Table 6, we conduct a $\eta$ search within the range $\{0.1, 0.3, 0.5, 1.0\}$ for both SphereFed (CE) and SphereFed (MSE). However, SphereFed (MSE) exhibit suboptimal performance within this range. Consequently, in Table 7, we extend the $\eta$ search exclusively for SphereFed (MSE) to include values $\{1.5, 3.0, 4.5, 5.0\}$. In summary, the determined optimal hyperparameter $\eta$ values are 0.03 for SphereFed (CE) and 4.5 for SphereFed (MSE).

Table 6: Grid Search Results for SphereFed with MobileNet on CIFAR-100.

| *η=0.1* | *E*=1 | | *E*=5 | | *E*=10 | |
|---|---|---|---|---|---|---|
| | CE | MSE | CE | MSE | CE | MSE |
| *s*=100 | 12.96 | 1.37 | 44.92 | 2.16 | 39.50 | 4.25 |
| *s*=50 | 12.21 | 1.30 | 43.38 | 1.73 | 38.65 | 3.84 |
| *s*=10 | 6.67 | 1.46 | 33.32 | 2.34 | 30.02 | 3.81 |
| *η=0.3* | *E*=1 | | *E*=5 | | *E*=10 | |
| | CE | MSE | CE | MSE | CE | MSE |
| *s*=100 | 24.62 | 1.39 | **44.96** | 5.25 | 40.47 | 19.41 |
| *s*=50 | 23.56 | 1.44 | **43.68** | 5.89 | 38.21 | 19.65 |
| *s*=10 | 14.91 | 1.40 | **40.39** | 4.46 | 33.53 | 15.69 |
| *η=0.5* | *E*=1 | | *E*=5 | | *E*=10 | |
| | CE | MSE | CE | MSE | CE | MSE |
| *s*=100 | 32.12 | 1.46 | 45.47 | 12.30 | 40.11 | 36.38 |
| *s*=50 | 28.45 | 1.43 | 44.97 | 11.76 | 39.49 | 34.07 |
| *s*=10 | 18.70 | 1.61 | 37.02 | 8.32 | 33.83 | 24.65 |
| *η=1.0* | *E*=1 | | *E*=5 | | *E*=10 | |
| | CE | MSE | CE | MSE | CE | MSE |
| *s*=100 | 38.98 | 2.33 | 46.45 | 29.94 | 41.58 | 40.02 |
| *s*=50 | 35.69 | 2.27 | 43.95 | 29.39 | 41.20 | 40.87 |
| *s*=10 | 26.40 | 1.70 | 36.72 | 23.38 | 35.73 | 32.64 |

Table 7: Additional Grid Search Results for SphereFed (MSE) with MobileNet on CIFAR-100.

| *η=1.5* | *E*=1 | *E*=5 | *E*=10 |
|---|---|---|---|
| | MSE | MSE | MSE |
| *s*=100 | 2.84 | 44.40 | 40.32 |
| *s*=50 | 2.81 | 41.91 | 42.18 |
| *s*=10 | 2.69 | 33.68 | 35.66 |
| *η=3.0* | *E*=1 | *E*=5 | *E*=10 |
| | MSE | MSE | MSE |
| *s*=100 | 6.37 | 47.44 | 42.15 |
| *s*=50 | 5.05 | 45.56 | 42.61 |
| *s*=10 | 4.87 | 40.84 | 34.68 |
| *η=4.5* | *E*=1 | *E*=5 | *E*=10 |
| | MSE | MSE | MSE |
| *s*=100 | 9.43 | **49.51** | 43.19 |
| *s*=50 | 8.19 | **48.05** | 42.42 |
| *s*=10 | 7.50 | **43.42** | 37.93 |
| *η=5.0* | *E*=1 | *E*=5 | *E*=10 |
| | MSE | MSE | MSE |
| *s*=100 | 10.58 | 49.29 | 36.07 |
| *s*=50 | 10.95 | 48.08 | 38.49 |
| *s*=10 | 8.26 | 43.26 | 21.73 |

# D Additional Experiments

## D.1 Weight Norm Does Not Matter on FedFN

We investigate the impact of increased data heterogeneity on weight norm disparity in the case of FedFN. Focusing on the $s$=2 setting, representing the most heterogeneous scenario, as shown in Figure 1, we explore the evolution of weight norms between global and local models. Initially, during early training stages, a noticeable weight norm bias emerges within local models, favoring seen (ID) classes over unseen (OOD) classes. However, this bias progressively diminishes as the learning rate decreases, eventually aligning local models with the weight norm mean of the global model.

Additionally, we analyze variations in weight norm means between local models on the ID classes and the global model on the total classes under varying data heterogeneity. Specifically, we consider $s \in \{2, 5, 10\}$ and IID (Exactly class balanced data distribution across clients) on the CIFAR-10 dataset. As illustrated in Figure 2, during initial training phases, the norm disparity between the global model and local models increases with data heterogeneity (i.e., IID $\rightarrow$ $s$=10 $\rightarrow$ $s$=5 $\rightarrow$ $s$=2). However, this disparity gradually diminishes across all settings as the learning rate decreases.

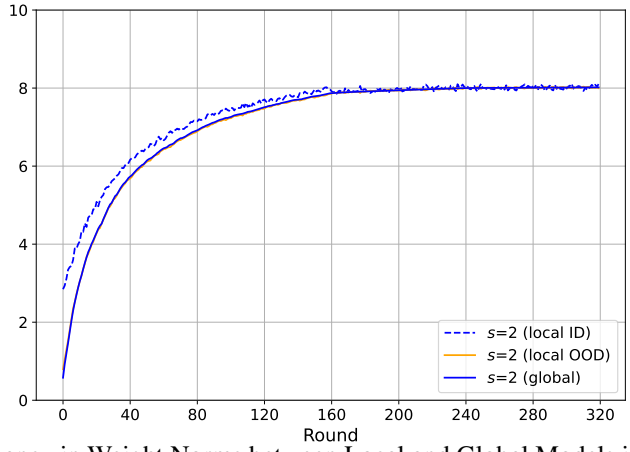



Figure 1: Discrepancy in Weight Norms between Local and Global Models in the $s$=2 Setting.

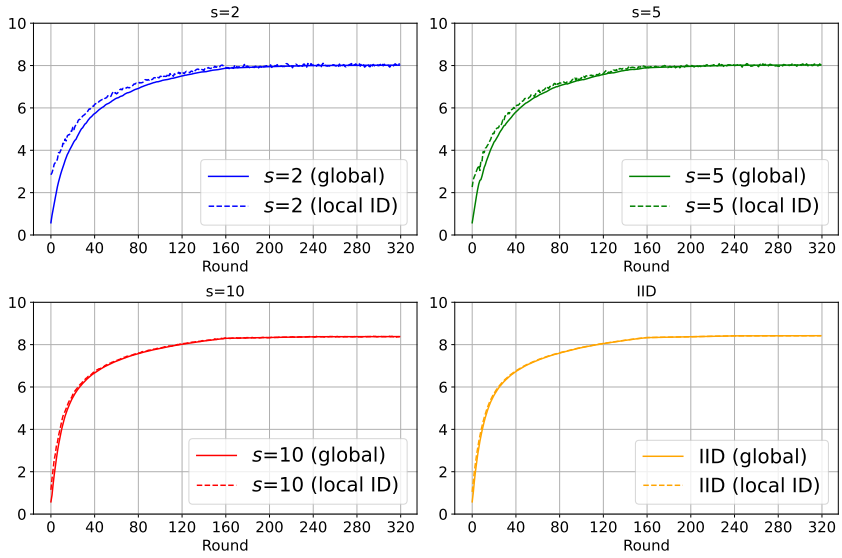



Figure 2: Discrepancy in Weight Norms between Local and Global Models in Various Heterogeneity Settings.

### D.2 Additional GFL Results

In the main paragraph of Section 5, we report results exclusively in a balanced environment, specifically in the sharding setting. In Table 8, results are presented, encompassing an unbalanced environment, specifically in the LDA setting. The reported outcomes stem from implementations with a consistent and identical seed, deviating from the context of the paragraph. In contrast to the main paragraph, we observe that in the MobileNet on CIFAR-100 environment with the $s = 10$ setting, all baseline algorithms exhibit notably lower performance for the seed used in Table 8. However, when applying the BABU or FN module, the performance difference is less pronounced, showing a more stable outcome compared to the main paragraph. Furthermore, consistently across all settings, applying the FN module to the baseline consistently demonstrates superior performance.

Table 8: FL Accuracy Comparison for Baseline, +BABU, and +FN.

| Algorithm | Module | VGG11 on CIFAR-10 | | | | | | | | MobileNet on CIFAR-100 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $s$=2 | $s$=3 | $s$=5 | $s$=10 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=1.0 | $s$=10 | $s$=50 | $s$=100 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=1.0 |
| FedAVG | Baseline | 73.83 | 78.99 | 81.40 | 82.79 | 73.16 | 80.77 | 81.63 | 82.45 | 26.64 | 40.18 | 41.50 | 42.66 | 42.68 | 43.82 | 41.57 |
| | + BABU | 74.31 | 78.88 | 81.17 | 82.37 | 72.42 | 80.59 | 80.94 | 82.93 | 43.96 | 39.94 | 40.37 | 42.99 | 39.61 | 39.53 | 39.58 |
| | **+ FN** | **77.23** | **81.46** | **82.80** | **84.16** | **75.58** | **81.85** | **83.06** | **83.65** | **45.16** | **48.83** | **49.54** | 47.87 | 47.66 | 47.86 | 47.80 |
| Scaffold | Baseline | 77.07 | (*Failed*) | **84.50** | 85.28 | (*Failed*) | 83.24 | 83.54 | 85.21 | 34.64 | 42.56 | 43.67 | **45.18** | 45.62 | 44.42 | 43.16 |
| | + BABU | 76.89 | 82.24 | 84.26 | 85.02 | (*Failed*) | 82.95 | 84.12 | 85.06 | 46.80 | 42.73 | 44.50 | 43.28 | 43.76 | 44.37 | 44.20 |
| | **+ FN** | **79.05** | **82.83** | **84.80** | **85.83** | **74.53** | **83.65** | **84.79** | **85.42** | **50.17** | **48.74** | **51.04** | 45.10 | **49.71** | **50.42** | **52.13** |
| FedEXP | Baseline | 73.92 | 78.86 | 81.47 | 82.61 | 73.39 | 80.94 | 81.16 | 82.64 | 26.57 | 39.93 | 42.11 | 42.71 | 42.77 | 44.54 | 41.55 |
| | +BABU | 74.48 | 78.91 | 81.16 | 82.17 | 71.61 | 80.50 | 81.24 | 82.72 | 45.44 | 41.49 | 40.24 | 42.73 | 40.09 | 39.54 | 40.66 |
| | **+ FN** | **77.17** | **80.92** | **82.79** | **83.57** | **75.03** | **82.29** | **83.12** | **83.36** | **45.37** | **48.45** | **49.80** | **47.73** | **47.16** | **48.32** | **48.01** |

### D.3 Personalized Federated Learning (PFL) Results

We present FedFN-FT, a fine-tuned algorithm for PFL, inspired by prior work [4, 27], utilizing local data. We compare FedFN-FT with existing PFL methods, including simple local models, 1-step approaches like FedPer [1], Per-FedAVG [5], and FedRep [3], as well as 2-step methods such as FedAVG-FT, FedBABU-FT [27], SphereFed-FT (CE, MSE)[4]. Table 9 provides detailed personalized accuracy results. The entries form of X±Y, representing the mean and standard deviation of personalized accuracies across all clients for PFL algorithms. Entries without standard deviation indicate performance on $D_{test}$, derived from the global model after the initial step of 2-step methods.

Regarding the SphereFed method, it constructs each element of the logit vector based on the cosine similarity between feature vectors and classifiers. Similar to FedFN, SphereFed necessitates rescaling the learning rate for classifier weights, leading to the need for separate learning rate tuning. To address this requirement, we conduct an extensive grid search to determine the appropriate initial learning rate, denoted as $\eta$. Further details about the grid search are provided in Section C.

For SphereFed (CE), SphereFed (MSE), and FedFN, we initialize the learning rates with $\eta$ values of 0.3, 4.5, and 0.5, respectively. The 2-step fine-tuning methods undergo a total of 5 local epochs, during which learning rate was carried out through grid search within the range of $\{\eta, 0.1 \times \eta, 0.01 \times \eta\}$. The comprehensive results of the grid search for SphereFed and FN, including learning rate adjustments, are confirmed to be available in Appendix D.3.1. The resulting tunned learning rates are documented in Table 9. In summary, 2-step methods consistently outperform 1-step methods in terms of PFL performance. Among the 2-step methods, FedFN-FT consistently exhibits superior performance.

Table 9: PFL Accuracy Comparison for MobileNet on CIFAR-100.

| Algorithm | $s$=10 | $s$=50 | $s$=100 |
|---|---|---|---|
| Local only | 58.64$\pm$7.21 | 25.38$\pm$4.12 | 18.52$\pm$3.15 |
| FedPer (2019) | 70.92$\pm$6.93 | 33.73$\pm$4.68 | 22.77$\pm$4.30 |
| Per-FedAVG (2020) | 32.57$\pm$11.02 | 43.09$\pm$7.36 | 45.00$\pm$7.05 |
| FedRep (2021) | 62.69$\pm$7.26 | 34.66$\pm$5.34 | 26.53$\pm$4.52 |
| FedAVG (2017) | 36.30 | 41.97 | 42.51 |
| FedAVG-FT (0.1$\times\eta$) | 77.39$\pm$6.40 | 50.57$\pm$5.31 | 46.95$\pm$4.90 |
| FedBABU (2022) | 45.73 | 39.57 | 40.70 |
| FedBABU-FT (0.01$\times\eta$) | 79.76$\pm$6.05 | 50.93$\pm$4.69 | 46.63$\pm$5.25 |
| SphereFed (CE) (2022) | 40.39 | 43.68 | 44.96 |
| SphereFed-FT (CE) (0.1$\times\eta$) | 77.24$\pm$6.38 | 55.08$\pm$5.33 | 50.17$\pm$5.04 |
| SphereFed (MSE) (2022) | 43.42 | 48.05 | 49.51 |
| SphereFed-FT (MSE) (0.01$\times\eta$) | **82.50**$\pm$6.04 | 56.45$\pm$5.56 | 53.10$\pm$4.90 |
| FedFN | 46.98 | 49.47 | 50.92 |
| **FedFN-FT** (0.1$\times\eta$) | **82.85**$\pm$5.83 | **60.78**$\pm$4.99 | **55.43**$\pm$5.24 |

### D.3.1 Fine Tuning Learning Rate Search

Table 10: Search for Finetuning Learning Rates in Two-Step Methods.

| Algorithm | $s$=10 | $s$=50 | $s$=100 |
|---|---|---|---|
| FedAVG (2017) | 36.30 | 41.97 | 42.51 |
| FedAVG-FT ($\eta$) | 67.23 $\pm$ 6.32 | 35.98 $\pm$ 5.56 | 38.25 $\pm$ 5.29 |
| **FedAVG-FT** (0.1$\times\eta$) | **77.39 $\pm$ 6.40** | **50.57 $\pm$ 5.31** | **46.95 $\pm$ 4.90** |
| FedAVG-FT (0.01$\times\eta$) | 70.96 $\pm$ 6.81 | 44.78 $\pm$ 4.88 | 44.00 $\pm$ 4.93 |
| FedBABU (2022) | 45.73 | 39.57 | 40.7 |
| FedBABU-FT ($\eta$) | 13.32 $\pm$ 4.51 | 4.72 $\pm$ 1.85 | 2.85 $\pm$ 1.27 |
| FedBABU-FT (0.1$\times\eta$) | 77.88 $\pm$ 6.59 | 51.75 $\pm$ 4.84 | 46.02 $\pm$ 5.24 |
| **FedBABU-FT** (0.01$\times\eta$) | **79.76 $\pm$ 6.05** | **50.93 $\pm$ 4.69** | **46.63 $\pm$ 5.25** |
| SphereFed (CE) (2022) | 40.39 | 43.68 | 44.96 |
| SphereFed-FT (CE) ($\eta$) | 53.79 $\pm$ 9.47 | 32.11 $\pm$ 6.89 | 27.81 $\pm$ 4.98 |
| **SphereFed-FT (CE)** (0.1$\times\eta$) | **77.24 $\pm$ 6.38** | **55.08 $\pm$ 5.33** | **50.17 $\pm$ 5.04** |
| SphereFed-FT (CE) (0.01$\times\eta$) | 77.18 $\pm$ 6.17 | 49.99 $\pm$ 4.79 | 47.45 $\pm$ 4.70 |
| SphereFed (MSE) (2022) | 43.42 | 48.05 | 49.51 |
| SphereFed-FT (MSE) ($\eta$) | 57.34 $\pm$ 7.53 | 35.43 $\pm$ 5.32 | 27.95 $\pm$ 5.03 |
| SphereFed-FT (MSE) (0.1$\times\eta$) | 79.73 $\pm$ 6.63 | 56.80 $\pm$ 5.29 | 51.06 $\pm$ 5.59 |
| **SphereFed-FT (MSE)** (0.01$\times\eta$) | **82.50 $\pm$ 6.04** | **56.45 $\pm$ 5.56** | **53.10 $\pm$ 4.90** |
| FedFN | 46.98 | 49.47 | 50.92 |
| FedFN-FT ($\eta$) | 71.88 $\pm$ 6.73 | 43.80 $\pm$ 4.54 | 41.23 $\pm$ 5.46 |
| **FedFN-FT** (0.1$\times\eta$) | **82.85 $\pm$ 5.83** | **60.78 $\pm$ 4.99** | **55.43 $\pm$ 5.24** |
| FedFN-FT (0.01$\times\eta$) | 82.15 $\pm$ 5.74 | 54.24 $\pm$ 5.06 | 51.92 $\pm$ 5.06 |

### D.4 Logit Should Be Non-Restricted

SphereFed [4] modifies i-th index of the logit vector of an input $x$, represented as $\tilde{z}_i(x;\theta) = \tilde{\theta}_{cls,i} \frac{f(x;\theta_{ext})}{||f(x;\theta_{ext})||_2}$. It maintains the classifier $\tilde{\theta}_{cls}$ in a frozen state, ensuring that the norms of $\tilde{\theta}_{cls,i}$ are orthonormal to each other. As a result, $\tilde{z}_i(x;\theta)$ becomes the cosine similarity between $f(x;\theta_{ext})$ and $\tilde{\theta}_{cls,i}$, yielding values restricted to the range [-1,1]. Following this modification, the logit margin is constrained to a maximum value of 2.

As seen in Table 9, despite utilizing feature normalization, SphereFed (CE) exhibits inferior performance compared to even FedBABU. To address this limitation, we propose a modification to the SphereFed (CE) logit vector. We transform it to $\tilde{z}_i^{\tau}(x;\theta) = \tau\, \tilde{\theta}_{cls,i} \frac{f(x;\theta_{ext})}{||f(x;\theta_{ext})||_2}$, which yields values in the range of [-$\tau$, $\tau$], providing less constrained outputs. We apply this approach with different values of $\tau$, specifically $\{10, 15, 20, 25, 30\}$, in the $s$=10 setting. We compared the results of this modified SphereFed (CE) with those of SphereFed (MSE), FedBABU, and FedFN, and the outcomes are presented in Table 11. Increasing $\tau$ up to 15 results in improvements in SphereFed(CE), although not as significant as compared to FedFN. However, it shows enhancements over FedBABU and SphereFed(MSE) at 15. Consequently, this indicates that creating the logit vector through feature normalization with relaxed constraints on the elements of the logit is recommended.

Table 11: Comparison of Modified SphereFed (CE) with SphereFed (MSE), FedBABU, and FedFN on $s$=10 Setting of CIFAR-100.

| Algorithm | Accuracy |
|---|---|
| SphereFed (CE), $\tau$=1 | 40.39 |
| SphereFed (CE), $\tau$=10 | 42.84 |
| SphereFed (CE), $\tau$=15 | **45.78** |
| SphereFed (CE), $\tau$=20 | 44.95 |
| SphereFed (CE), $\tau$=25 | 44.46 |
| SphereFed (CE), $\tau$=30 | 39.62 |
| SphereFed (MSE) | 43.42 |
| FedBABU | 45.73 |
| FedFN | **46.98** |

### D.5 Reproduced Result from SphereFed

We present the experimental results for SphereFed [4] trained with LDA settings ($\alpha \in \{0.1, 0.5\}$) on CIFAR-100. To reproduce the experiments presented in the original paper, we deviated from our previous experimental settings. Specifically, for the MobileNetV2 model architecture, we constructed the layers exactly as described in Table 7 of [4]. Regarding the training setup, each case is trained for 500 rounds using cosine annealing, following the guideline of original paper. We follow the instructions of original paper for all other hyperparameters as well. It should be noted that we did not employ the FFC algorithm in any of the experiments, including those using FedAVG and FedFN.

Table 12 presents the new results we obtain and compares them with the original outcomes of SphereFed, encompassing FedAVG, FedFN and centralized learning. If certain algorithms are not indicated at the original results, we represent them with a dash ("-"). For the algorithms implemented as described in the original paper, we provide specific details like the actual learning rate $\eta$. Furthermore, for each FL algorithm, we present reproduced results across a specified range of initial learning rates $\eta \in \{0.1, 0.3, 0.5, 1.0, 1.5, 3.0, 4.5, 5.0\}$. In the case of centralized learning, results are specifically provided for $\eta = 0.1$.

Following this, we conclude that the results of the original paper could not be reproduced. The reported performance of FL algorithms in the actual original paper (71.85, 68.78) appears surprisingly higher than the reproduced results in centralized learning (68.27). Moreover, implementing the algorithms with the exact settings reported in the original results consistently leads to lower performance (71.85 vs 18.01, 68.78 vs 37.76). Even when implemented in accordance with the specifications of the original paper ($\eta$=0.5), SphereFed (MSE) demonstrated significantly poor performance at 18.01. Subsequently, despite a thorough investigation through grid search, the best-performing configuration obtained is 52.69, still falling below the reported performance. Furthermore, when comparing the performance at the optimal learning rate for each FL algorithm, we consistently observe that FedFN outperforms the baselines.

Table 12: Reproduced Results for SphereFed, FedAVG, and FedFN under the Same Settings, Utilizing MobileNet (as Described in [4]) on CIFAR-100.

| *SphereFed (MSE)* | *$\eta$=0.1* | *$\eta$=0.3* | *$\eta$=0.5* | *$\eta$=1.0* | *$\eta$=1.5* | *$\eta$=3.0* | *$\eta$=4.5* | *$\eta$=5.0* | Original Result (*$\eta$=0.5*) |
|---|---|---|---|---|---|---|---|---|---|
| $\alpha$=0.5 | 2.77 | 9.73 | 18.01 | 43.26 | 52.19 | **52.69** | 48.20 | 40.87 | 71.85 |
| $\alpha$=0.1 | 2.88 | 9.65 | 20.19 | 41.48 | 45.41 | **46.34** | 43.62 | 40.56 | - |
| *SphereFed (CE)* | *$\eta$=0.1* | *$\eta$=0.3* | *$\eta$=0.5* | *$\eta$=1.0* | *$\eta$=1.5* | *$\eta$=3.0* | *$\eta$=4.5* | *$\eta$=5.0* | Original Result |
| $\alpha$=0.5 | 37.57 | 49.39 | 48.18 | 52.51 | **52.99** | 51.09 | 44.87 | 44.23 | - |
| $\alpha$=0.1 | 22.37 | 40.58 | 42.85 | 40.92 | **47.72** | 42.11 | 35.30 | 33.83 | - |
| *FedAVG* | *$\eta$=0.1* | *$\eta$=0.3* | *$\eta$=0.5* | *$\eta$=1.0* | *$\eta$=1.5* | *$\eta$=3.0* | *$\eta$=4.5* | *$\eta$=5.0* | Original Result (*$\eta$=0.1*) |
| $\alpha$=0.5 | 37.76 | **38.82** | 23.76 | 1.04 | 1.02 | 1.03 | 1.25 | 1.01 | 68.78 |
| $\alpha$=0.1 | 38.58 | **40.47** | 27.35 | 1.22 | 1.04 | 1.02 | 1.09 | 1.17 | - |
| *FedFN* | *$\eta$=0.1* | *$\eta$=0.3* | *$\eta$=0.5* | *$\eta$=1.0* | *$\eta$=1.5* | *$\eta$=3.0* | *$\eta$=4.5* | *$\eta$=5.0* | Original Result |
| $\alpha$=0.5 | 55.00 | **53.38** | 48.55 | 49.89 | 45.79 | 42.45 | 35.43 | 34.82 | - |
| $\alpha$=0.1 | 46.38 | **49.16** | 46.61 | 41.69 | 42.03 | 38.80 | 30.92 | 2.89 | - |
| *Centralized Learning* | **68.27** (*$\eta$=0.1*) | | | | | | | | - |

### D.6 FedFN vs FedFR

We compare the performance of FedFN with Federated Averaging with Feature Norm Regularization (FedFR) introduced in Eq. (3.4) in the main paragraph. Table 13 and Table 14 present the performance on CIFAR-10 and CIFAR-100 with $s = 10$ setting, respectively.

Table 13 reports results of FedFR on CIFAR-10, referring to the optimal hyperparameter $\mu = 0.005$ from Figure 4 in the main paragraph. FedFR exhibits slightly lower performance compared to FedFN but demonstrates superiority over FedAVG and FedBABU. In the $s = 10$ setting of CIFAR-100, FedFR shows comparable or superior performance to FedAVG. However, FedFR performs worse than both FedBABU and FedFN across all hyperparameter candidates. In contrast to FedFR, FedFN consistently demonstrates superior performance across all settings.

Table 13: Accuracy Comparison on CIFAR-10.

| Algorithm | VGG11 on CIFAR-10 | | | |
|---|---|---|---|---|
| | $s$=2 | $s$=3 | $s$=5 | $s$=10 |
| FedAVG | 74.24 | 77.29 | 81.08 | 81.97 |
| FedBABU | 75.05 | 77.73 | 81.04 | 82.16 |
| FedFR | 76.14 | 77.89 | 81.61 | 82.18 |
| FedFN | **77.77** | **78.93** | **82.43** | **83.80** |

Table 14: Accuracy Comparison on the $s = 10$ Setting of MobileNet on CIFAR-100.

| FedFR | *$\mu$=0.5* | *$\mu$=0.1* | *$\mu$=0.05* | *$\mu$=0.01* | *$\mu$=0.005* | *$\mu$=0.001* | *$\mu$=0.0005* | *$\mu$=0.0001* |
|---|---|---|---|---|---|---|---|---|
| | *(Failed)* | 37.74 | **39.50** | 36.72 | 36.51 | 36.77 | 37.04 | 37.30 |
| FedAVG | **36.30** (*$\mu$=0.0*) | | | | | | | |
| FedBABU | **45.73** | | | | | | | |
| FedFN | **46.98** | | | | | | | |

### D.7 FN in the Centralized Learning

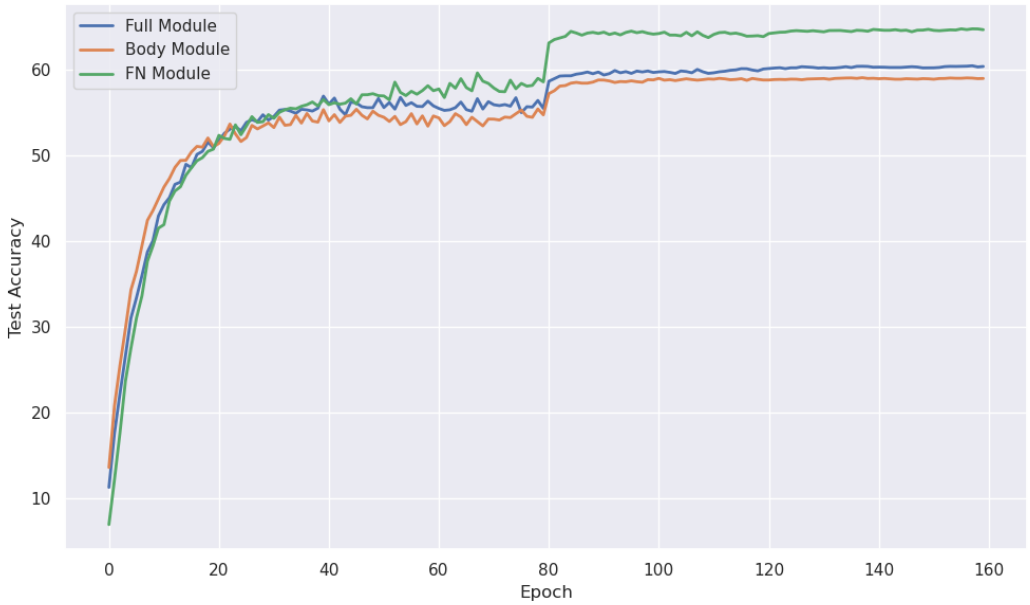



Figure 3: Module Comparison in Centralized Learning

In FedBABU [27], the authors comprehensively evaluate the performance of the Full and Body modules in a centralized learning setup. Expanding upon their analysis, we incorporate the FN module to assess its performance in comparison to that of the Full and Body modules. To ensure fair comparisons, we replicate the experimental settings outlined in [27]. The experiments are carried out on the CIFAR-100 dataset, utilizing the MobileNet architecture. The results of the conducted experiments are depicted in Figure 3.

As expected, our evaluation reveals a slight decline in performance within the Body module compared to the Full module, aligning with the findings reported in [27]. This performance difference can be attributed to the partial update constraint imposed on the model during the training of the Body module. In contrast, the FN module, despite incorporating the constraint of feature normalization, exhibits significant improvements over the Full module. This enhancement is attributed to the capacity of FN module to empower the model to acquire more effective and discriminative representations, thereby enhancing overall performance.