# Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning – Supplementary Material

Matias Mendieta[1], Taojiannan Yang[1], Pu Wang[2], Minwoo Lee[2], Zhengming Ding[3], Chen Chen[1]
[1]Center for Research in Computer Vision, University of Central Florida, USA
[2]Department of Computer Science, University of North Carolina at Charlotte, USA
[3]Department of Computer Science, Tulane University, USA
{mendieta,taoyang1122}@knights.ucf.edu; chen.chen@crcv.ucf.edu
{pu.wang,minwoo.lee}@uncc.edu; zding1@tulane.edu

## A. Overview

The supplementary material is organized into the following sections:

- Section B: Analysis of both communication and compute efficiency for all explored methods.
- Section C: Second-order analysis of FedAlign.
- Section D: Hyperparameter ablations for FedAlign.
- Section D: Details and visualization for the non-IID data partitioning scheme.
- Section F: Implementation details for transmitting matrices and FedAlign training.

## B. Communication and Compute Efficiency

Communication cost is another critical factor in FL systems, as participating client devices are often on slow or congested networks [4]. Therefore, total efficiency in FL systems includes both the ability to reduce the local computational burden, as well as the communication overhead. We evaluate all methods with such measures in Table 1. We maintain the CIFAR-100 setting described in Section 3.2 of the main paper, except that we do not limit the number of rounds, but rather allow all methods to reach a common accuracy of 60%. This allows us to analyze the total costs of each method consistently. FedAlign proves to be the most efficient in all respects, achieving the target accuracy in less number of rounds, less communication cost, and less local computation.

Table 1. Number of rounds (Rounds), local compute (MFLOPs), and communication cost (Comm Cost) required by each method to achieve 60% accuracy on CIFAR-100. Local computation is computed as the sum total over all nodes and samples for all completed rounds. Communication costs are calculated by the number of parameters of the model transferred as 32 bit weights with all completed rounds.

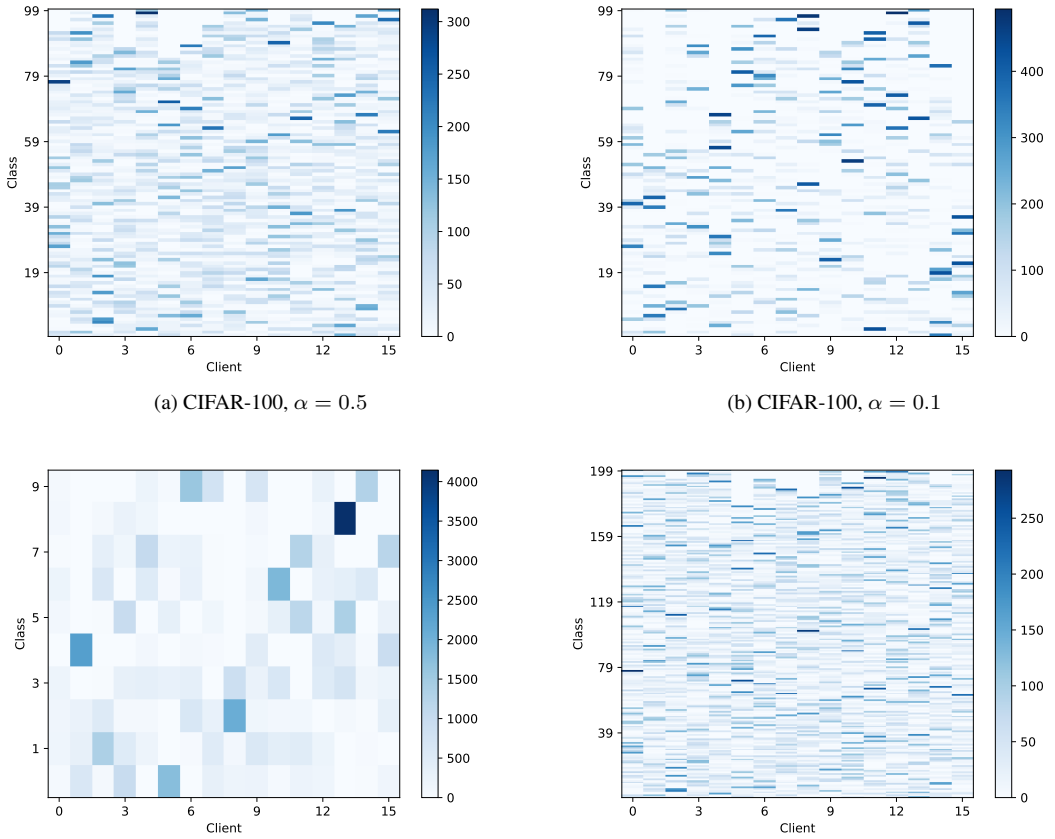
| Method | Rounds | MFLOPs | Comm Cost (Gb) |
|---|---|---|---|
| FedAvg | 84 | 7332 | 26.23 |
| FedProx | 78 | 6809 | 24.36 |
| MOON | 55 | 14421 | 17.18 |
| Mixup | 71 | 6198 | 22.17 |
| StochDepth | 46 | 3790 | 14.37 |
| GradAug ($n = 2$) | 41 | 6999 | 12.81 |
| GradAug ($n = 1$) | 44 | 5892 | 13.74 |
| **FedAlign** | **37** | **3297** | **11.56** |

## C. Second-order Analysis

In Table 2, we show the second-order analysis results for FedAlign along with the other methods. The Liptzshitz-focused distillation loss of FedAlign effectively reduces the Lipshitz constant $\lambda_{max}$ considerably as intended (FedAlign results in the lowest $\lambda_{max}$ across all methods), and therefore helps provide stronger generalization and performance. However, one limitation to FedAlign is that it does not directly translate to a strong reduction in $H_N$. Therefore, a promising direction for future work could extend FedAlign to also consider this aspect.

## D. Hyperparameter Ablations of FedAlign

The default hyperparameter setting used throughout the paper is $\omega_S = 0.25$ and $\mu = 0.45$. The performance of FedAlign with various hyperparameters on the CIFAR-100 basic setting (described in Section 3.2 of the main paper) is shown in Table 3. We vary $\mu$ and $\omega_S$ independently, meaning $\omega_S = 0.25$ for the $\mu$ ablations, and $\mu = 0.45$ when varying $\omega_S$. Table 3 shows that FedAlign is more sensitive to $\omega_s$ than $\mu$; nonetheless, we found $\omega = 0.25$ to be a versatile choice in practice. Furthermore, hyperparameters only




(a) CIFAR-100, $\alpha = 0.5$

(b) CIFAR-100, $\alpha = 0.1$

(c) CIFAR-10, $\alpha = 0.5$

(d) ImageNet-200, $\alpha = 0.5$

Figure 1. Data distribution visualization for $Dir(\alpha)$ and $C = 16$ across multiple datasets. Each column shows the number of samples per class allocated to a client.

need to be decided once, as they transfer well across a variety of other datasets and FL settings as shown in Section 4.1 of the main paper.

Table 2. Results for accuracy (%) on CIFAR-100 and second-order metrics indicating the smoothness of the loss space ($\lambda_{max}$, $H_T$) and cross-client consistency ($H_N$, $H_D$) for each method.

| Method | Acc. ↑ | $\lambda_{max}$↓ | $H_T$↓ | $H_N$↓ | $H_D$↑ |
|---|---|---|---|---|---|
| FedAvg | 52.9 | 297 | 6240 | 11360 | 0.98 |
| FedProx | 53.0 | 270 | 6132 | 6522 | 0.98 |
| MOON | 55.3 | 252 | 5520 | 5712 | 0.97 |
| Mixup | 54.0 | 216 | 5468 | 15434 | **0.99** |
| StochDepth | 55.5 | 215 | 3970 | 8267 | 0.97 |
| GradAug ($n = 2$) | **57.1** | 167 | **2597** | 2924 | 0.96 |
| GradAug ($n = 1$) | 56.8 | 179 | 3620 | **2607** | 0.97 |
| **FedAlign** | 56.7 | **143** | 4409 | 9655 | **0.99** |

Table 3. FedAlign hyperparameter ablations on with CIFAR-100

| Method | $\mu = 0.35$ | $\mu = 0.45$ | $\mu = 0.55$ | $\omega_S = 0.1$ | $\omega_S = 0.25$ | $\omega_S = 0.4$ |
|---|---|---|---|---|---|---|
| FedAlign | 56.0 | **56.7** | 56.1 | 54.9 | **56.7** | 55.2 |

## E. Data Partitioning

As is common in the literature [1–3], we partition the employed datasets into $K$ unbalanced subsets using a Dirichlet distribution $Dir(\alpha)$. The distribution for all three datasets at $\alpha = 0.5$ is visualized in Fig. 1 (a), (c), and (d). Additionally, (b) shows the distribution for CIFAR-100 with $\alpha = 0.1$ as studied in Section 3.5 of the main paper. Overall, we see that the number of samples for each class varies considerably across clients, and often times a client will not have any samples from multiple classes. This enhances the FL setting by making it more realistic and challenging. For implementation, we utilize the same data partitioning script as that in [2].

## F. Additional Implementation Details

When calculating $\mathbf{X}_F$ and $\mathbf{X}_S$, the input and output features involved will typically be of different spatial sizes in practice. Therefore, [5] utilizes an adaptive average pool operation in PyTorch to reduce the spatial size of the larger feature map to that of the smaller one. We likewise employ this operation.

Prior to performing backpropagation, we apply a relative scale to $\mathcal{L}_{Lip}$ along with the $\mu$ scaling parameter. In PyTorch-style pseudocode: $loss\_lip = \mu^* (loss\_ce.item()/loss\_lip.item())^* loss\_lip$. This is to ensure that $\mathcal{L}_{Lip}$ is on relatively the same scale with $\mathcal{L}_{CE}$. A gradient clip is also applied.

## References


[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. 2

[2] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020. 2

[3] 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. 2

[4] 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. 1

[5] Yuzhang Shang, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. Lipschitz continuity guided knowledge distillation, 2021. 3