FedDisco: Federated Learning with Discrepancy-Aware Collaboration

Rui Ye [1] **Mingkai Xu** [1] **Jianyu Wang** [2] **Chenxin Xu** [1] **Siheng Chen** [1 3] **Yanfeng Wang** [3 1]

Abstract

This work considers the category distribution heterogeneity in federated learning (FL). This issue is due to biased labeling preferences at multiple clients and is a typical setting of data heterogeneity. To alleviate this issue, most previous works consider either regularizing local models or fine-tuning the global model, while they ignore the adjustment of aggregation weights and simply assign weights based on the dataset size. However, based on our empirical observations and theoretical analysis, we find that the dataset size is not optimal and the discrepancy between local and global category distributions could be a beneficial and complementary indicator for determining aggregation weights. We thus propose a novel FL algorithm, Federated Learning with Discrepancy-Aware Collaboration (FedDisco), whose aggregation weights not only involve both the dataset size and the discrepancy value, but also contribute to a tighter theoretical upper bound of the optimization error. FedDisco can promote utility and modularity in a communication- and computation-efficient way. Extensive experiments show that our FedDisco outperforms several state-of-the-art methods and can be easily incorporated with many existing methods to further enhance the performance. Our code will be available at https://github.com/MediaBrain-SJTU/FedDisco.

1. Introduction

Federated learning (FL) is an emerging field that offers a privacy-preserving collaborative machine learning paradigm (Kairouz et al., 2019). Its main idea is to enable multiple clients to collaboratively train a shared global model by sharing information about model parameters. FL algorithms have been widely applied to many real-world scenarios, such as users' next-word prediction (Hard et al., 2018), recommendation systems (Ding et al., 2022), and smart healthcare (Kaissis et al., 2020).

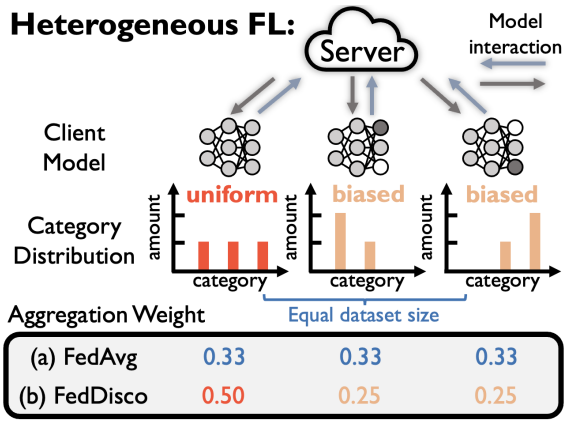


Figure 1. To mitigate the negative impact of category distribution heterogeneity, FedDisco considers discrepancy-aware aggregation weights, which involves both dataset size and discrepancy between local category distribution and uniform distribution. FedDisco is still privacy preserving, communication and computation efficient.

In spite of the promising trend, there are still multiple key challenges that impede the further development of FL, including data heterogeneity, communication cost and privacy concerns (Kairouz et al., 2019). In this work, we focus on one critical issue in data heterogeneity: *category distribution heterogeneity*; that is, clients have drastically different category distributions. For example, some clients have many samples in category 1 but very few in category 2; while other clients are opposite. This setting is common in practice, since clients collect local data privately and usually tend to have biased labeling preferences. Due to this heterogeneity, different local models are optimized towards different local objectives, causing divergent optimization directions. It is thus difficult to aggregate these divergent local models to obtain a robust global model. This unpleasant phenomenon has been theoretically and empirically verified in (Zhao et al., 2018; Li et al., 2019; Wang et al., 2021).

To mitigate the negative impact of category distribution heterogeneity, there are mainly two approaches: i) local model adjustment at the client side and ii) global model

[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China [2]Carnegie Mellon University, Pittsburgh, PA, USA [3]Shanghai AI Laboratory, Shanghai, China. Correspondence to: Siheng Chen <sihengc@sjtu.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

adjustment at the server side. Most previous works consider the first approach. For example, FedProx (Li et al., 2020a) and FedDyn (Acar et al., 2020) introduce regularization terms; and MOON (Li et al., 2021) aligns intermediate outputs of local and global models to reduce the variance among optimized local models. Along the line of the second approach, FedDF and FedFTG (Lin et al., 2020; Zhang et al., 2022) introduce an additional fine-tuning step to refine the global model. Despite all these diverse efforts, few works consider optimizing the aggregation weight at the server side[1]. In fact, most previous works simply aggregate the local models according to the dataset size at each local client. However, the dataset size does not reflect any categorical information and thus cannot provide sufficient information of a local client. It is still unclear whether the dataset size based aggregation is the best aggregation strategy.

In this paper, we first answer the above open question by empirically showing that, in many cases, aggregating local models based on the local dataset size is consistently far from optimal; meanwhile, incorporating the discrepancy between local and global category distributions into the aggregation weights could be beneficial. Intuitively, a lower discrepancy value reflects that the private data at a local client has a more similar category distribution with the hypothetically aggregated global data, and the corresponding client should contribute more to the global model than those with higher discrepancy values. To understand the effect of category distribution heterogeneity, we next provide a theoretical analysis for federated averaging with arbitrary aggregation weights, dataset sizes and local discrepancy levels. By minimizing the reformulated upper bound of the optimization error, we obtain an optimized aggregation weights for clients, which depend on not only the local dataset sizes but also the local discrepancy levels.

Inspired by the above empirical and theoretical observations, we propose a novel model aggregation method, federated learning with discrepancy-aware collaboration (FedDisco), to alleviate the category distribution heterogeneity issue. This new method leverages each client's dataset size and discrepancy in determining aggregation weight by assigning larger aggregation weights to those clients with larger dataset sizes and smaller discrepancy values. As a novel model aggregation scheme, FedDisco is still privacy-preserving and can be easily incorporated with many FL methods, such as FedProx (Li et al., 2020a) and MOON (Li et al., 2021). Compared to the vanilla aggregation based on local dataset size, FedDisco nearly introduces no additional computation and communication costs.

To validate the effectiveness of our proposed FedDisco, we conduct extensive experiments under diverse scenarios, such as various heterogeneous settings, globally balanced and imbalanced distributions, full and partial client participation, and across four datasets. We observe that FedDisco can significantly and consistently improve the performance over existing FL algorithms.

The main contributions of this paper is listed as follows:

1. We empirically show that the dataset size is not optimal to determine aggregation weights in FL and the discrepancy between local and global category distributions could be a beneficial complementary indicator;
2. We theoretically show that aggregation weight correlated with both dataset size and discrepancy could contribute to a tighter error bound;
3. We propose a novel aggregation method, called FedDisco (federated learning with discrepancy-aware collaboration), whose aggregation weight follows from the optimization of the theoretical error bound;
4. Extensive experiments show that FedDisco achieves state-of-the-art performances under diverse scenarios.

2. Background of Federated Learning

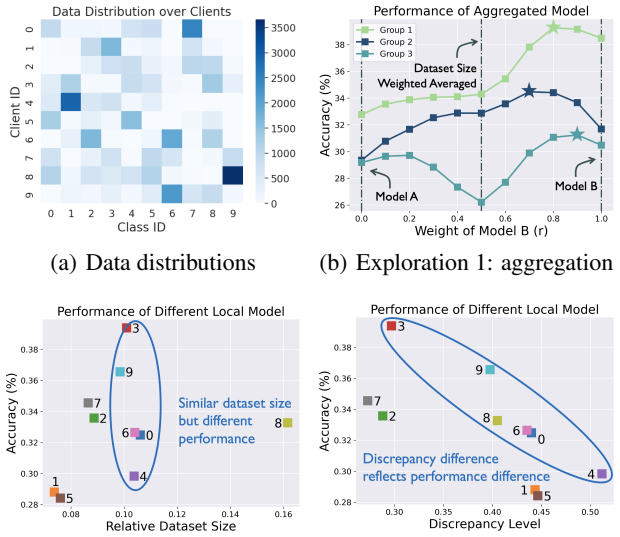
Suppose there are total K clients, where the k-th client holds a private dataset $\mathcal{B}_k = \{(\mathbf{x}_i, y_i)|i = 1, 2, ..., |\mathcal{B}_k|\}$ and a local model $\mathbf{w}_k^{(t,r)}$, where $\mathbf{x}_i$ and y_i denote the i-th input and label, t and r denote the round and iteration indices of training. The local category distribution of k-th client's dataset is defined as $\mathbf{D}_k \in \mathbb{R}^C$, where the c-th element $\mathbf{D}_{k,c} = \frac{|\{(\mathbf{x}_i, y_i)|y_i=c\}|}{|\mathcal{B}_k|}, c \in \{1, 2, \cdots, C\}$ is the data proportion of the c-th catregory. See detailed notation descriptions in Table 8.

Mathematically, the global objective of federated learning is $\min_{\mathbf{w}} F(\mathbf{w}) = \sum_{k=1}^{K} n_k F_k(\mathbf{w})$, where $n_k = \frac{|\mathcal{B}_k|}{\sum_{i=1}^{K} \mathcal{B}_i}$ and $F_k(\mathbf{w})$ are the dataset relative size and local objective of client k, respectively. In the basic FL, FedAvg (McMahan et al., 2017), each training round t proceeds as follows:

1. Sever broadcasts the global model $\mathbf{w}^{(t,0)}$ to clients;
2. Each client k performs local model training using τ SGD steps to obtain a trained model denoted by $\mathbf{w}_k^{(t,\tau)}$;
3. Clients upload the local models to the server;
4. Server updates the global model based on the aggregated local models: $\mathbf{w}^{(t+1,0)} = \sum_{k=1}^{K} p_k \mathbf{w}_k^{(t,\tau)}$, where p_k is the aggregation weight for the client k.

As mentioned in introduction, the category distribution heterogeneity issue could cause different local objectives in

[1]FedNova (Wang et al., 2020b) targets system heterogeneity.



(a) Data distributions (b) Exploration 1: aggregation

(c) Exploration 2: dataset size (d) Exploration 2: discrepancy

Figure 2. Experiments for empirical observations. Exploration 1 shows that dataset size could be not optimal indicator for aggregation weight. Exploration 2 shows that discrepancy could be a beneficial complementary indicator.

local training. To address this, many previous works focus on adjustment on training $\mathbf{w}_k^{(t,\tau)}$, while neglects the adjustment of aggregation weight p_k. In fact, most previous works conduct model aggregation simply based on the local dataset relative size; that is, $p_k = n_k$. However, we notice that dataset-size-based weighted aggregation could be not optimal in empirical investigation, which motivate us to search for a better aggregation strategy. The empirical observations are delivered in the next section.

3. Empirical Observations

This section presents a series of empirical observations, which motivate us to consider a new form of aggregation weight and propose our method. In the experiments, we divide the CIFAR-10 (Krizhevsky et al., 2009) dataset into 10 clients, whose local category distributions are heterogeneous and follow commonly used Dirichlet distribution (Wang et al., 2020a); see Figure 2 (a). Based on this common setting, we conduct the following experiments.

To explore the relationship between the aggregation weight and the performance of aggregated global model, we conduct three independent trials, where each trial considers federated training of two clients with similar dataset sizes. In each trial, each client trains a model (denoted as A and B) for 10 epochs and we aggregate these two models following: $\mathbf{w}_{\text{agg}} = (1-r)\mathbf{w}_{\text{A}} + r\mathbf{w}_{\text{B}}$. Figure 2 (b) shows three curves, which are the testing results of the aggregated model $\mathbf{w}_{\text{agg}}$ as a function of aggregation weight r in three trials, respectively. Note that $r = 0$ reflects Model A, $r = 1$ reflects Model B, and $r = 0.5$ represents dataset-size-based weighted aggregation. We observe that i) **it is not optimal to determine aggregation weights purely based on local dataset size.** The performances at $r = 0.5$ could be far from the optimal performances (stars in the plot); and ii) **best performance is achieved when assigning a relatively larger weight r to a better-performed local model.** In these trials, Model B outperforms Model A, and the best performance is achieved when Model B has a larger weight (r is around $0.7 \sim 0.9$). Similar phenomena can be seen in ensemble learning (Jiménez, 1998; Shen & Kong, 2004), which assigns larger weight to better model in ensemble.

To search for indicators that can reflect local model's performance and eventually appropriate aggregation weight, we explore the effects of two informative properties of client in category heterogeneity scenario: dataset size and discrepancy between local and global category distribution. Here we use ℓ_2 distance to measure the discrepancy. We plot all 10 clients in Figure 2 (c) & (d), where y-axis denote the local model's testing accuracy and x-axis in (c) and (d) denotes client's dataset relative size and discrepancy level, respectively. We highlight four clients in a circle, which have similar dataset sizes. Comparing these two plots, we clearly see that **the discrepancy level is a better indicator to reflect the local model's performance than the dataset size.** In plot (c), we see that these clients have similar dataset sizes but largely different performances; while in plot (d), the discrepancy level clearly reflects the performance difference among these four clients, that is, the client with a smaller discrepancy performs better.

Based on these observations, we hypothesize that, in FL, when a client has a smaller discrepancy value, it might have a better-performed model and thus need a larger weight in aggregation. Motivated by this, we propose to leverage discrepancy in determining the aggregation weight of each client, which is theoretically analyzed in the following.

4. Theoretical Analysis

In this section, we firstly obtain a convergence error bound for FedAvg (McMahan et al., 2017), which highlights the effect of aggregation weight p_k, dataset size n_k and discrepancy level d_k. Then, a concise analytical expression of the optimized aggregation weight is derived by minimizing the error bound, which indicates that aggregation weight should depend on both dataset size and local discrepancy level.

Optimization error upper bound. Our analysis is based on the following four standard assumptions in FL.

Assumption 4.1 (Smoothness)**.** Function $F_k(\mathbf{w})$ is Lipschitz-smooth: $||\nabla F_k(\mathbf{x}) - \nabla F_k(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$

for some L.

Assumption 4.2 (Bounded Scalar)**.** The global objective function $F(\mathbf{w})$ is bounded below by F_{inf}.

Assumption 4.3 (Unbiased Gradient and Bounded Variance)**.** For each client, the stochastic gradient is unbiased: $\mathbb{E}_\xi[g_k(\mathbf{w}|\xi)] = \nabla F_k(\mathbf{w})$, and has bounded variance: $\mathbb{E}_\xi[||g_k(\mathbf{w}|\xi) - \nabla F_k(\mathbf{w})||^2] \leq \sigma^2$.

Assumption 4.4 (Bounded Dissimilarity)**.** For each loss function $F_k(\mathbf{w})$, there exists constant $B > 0$ such that $||\nabla F_k(\mathbf{w})||^2 \leq ||\nabla F(\mathbf{w})||^2 + Bd_k$.

All assumptions are commonly used in federated learning literature (Wang et al., 2020b; Li et al., 2020a; 2019; Reddi et al., 2021). As no previous literature has considered discrepancy in their theory, a new assumption is required to include discrepancy. However, rather than introducing an additional assumption, we slightly adjust standard dissimilarity assumption (Wang et al., 2020b) and apply Assumption 4.4 to include the discrepancy level, this modification correlates the gradient dissimilarity and distribution discrepancy, and enables us to explore the relationships among aggregation weight p_k, dataset relative size n_k and discrepancy d_k.

We present the optimization error bound in Theorem 4.5.

Theorem 4.5 (Optimization bound of the global objective function)**.** *Let $F(\mathbf{w}) = \sum_{k=1}^{K} n_k F_k(\mathbf{w})$ be the global objective. Under these Assumptions, if we set $\eta L \leq \frac{1}{2\tau}$, the optimization error will be bounded as follows:*

$$\min_t \mathbb{E}||\nabla F(\mathbf{w}^{(t,0)})||^2 \leq \underbrace{\frac{1}{1-3A-W_D(1-A)}}_{T_0}$$

$$\Bigg(\underbrace{\frac{2(1-A)[F(\mathbf{w}^{(0,0)})-F_{inf}]}{\tau\eta T}}_{T_1} + \underbrace{\frac{(1-A)W_D B}{K}\sum_{k=1}^{K} d_k}_{T_2}$$

$$+ \underbrace{2(1-A)L\eta\sigma^2\sum_{k=1}^{K} p_k^2}_{T_3} + \underbrace{2(\tau-1)\sigma^2 L^2\eta^2}_{T_4} + \underbrace{2AB\sum_{k=1}^{K} p_k d_k}_{T_5}\Bigg)$$

where $W_D = 2\sum_{k=1}^{K}(n_k - p_k)^2$, $A = 2\tau(\tau-1)\eta^2 L^2$, η is local learning rate, B is a constant in Assumption 4.4.

Proof. See details in Appendix C.2. □

Generally, a tighter bound corresponds to a better optimization result. Thus, we explore the effects of p_k on upper bound. In Theorem 4.5, there are four parts related to p_k. First, we see that larger difference between p_k and n_k contributes to larger W_D and thus smaller denominator in T_0 and larger value in T_2, which tends to loose the bound. As for T_5, by setting p_k negatively correlated to d_k, when clients have different level of discrepancy, i.e. different d_k, T_5 will be further reduced, which tends to tight the bound. Therefore, there could be an optimal set of $\{p_k|k \in [K]\}$ that contributes to the tightest bound, where an optimal p_k should be correlated to both n_k and d_k. This theoretically show that dataset size could be not the optimal indicator and that discrepancy level can be a reasonable complementary indicator for determining aggregation weight.

Upper bound minimization. In the following, we derive the analytical expression of aggregation weight (p_k) by minimizing the upper bound in Theorem 4.5. To minimize this upper bound, directly solving the minimization results in a complicated expression, which involves too many unknown hyper-parameters in practice. To simplify the expression, we convert the original objective from minimizing $(T_1 + T_2 + T_3 + T_4 + T_5)/T_0$ to minimizing $T_1+T_2+T_3+T_4+T_5-\lambda T_0$, where λ is a hyper-parameter. The converted objective still promotes maximization of T_0 and minimization of $T_1 + T_2 + T_3 + T_4 + T_5$, and still contributes to tighten the bound $(T_1 + T_2 + T_3 + T_4 + T_5)/T_0$ (also see analysis in Section 7.3). Then, our discrepancy-aware aggregation weight is obtained through solving the following optimization problem:

$$\begin{aligned}
\min_{\{p_k\}} & \frac{2(1-A)[F(\mathbf{w}^{(0,0)})-F_{inf}]}{\tau\eta T} + \frac{(1-A)W_D B}{K}\sum_{k=1}^{K} d_k \\
& + 2(1-A)L\eta\sigma^2\sum_{k=1}^{K} p_k^2 + 2(\tau-1)\sigma^2 L^2\eta^2 \\
& + 2AB\sum_{k=1}^{K} p_k d_k - \lambda\left(1-3A-W_D(1-A)\right), \\
\text{s.t.} & \sum_k p_k = 1,\ p_k \geq 0,
\end{aligned} \tag{1}$$

from which we derive the concise expression of an optimized aggregation weight (see details of derivation in Appendix C.5):

$$p_k \propto n_k - a \cdot d_k + b, \tag{2}$$

where a, b are two constants. This suggests that for a tighter upper bound, the aggregation weight p_k should be correlated with both dataset size n_k and local discrepancy level d_k. This expression of Disco aggregation weight can mitigate the limitation of standard dataset size weighted aggregation by assigning larger aggregation weight to client with larger dataset size and smaller discrepancy level.

5. FL with Discrepancy-Aware Collaboration

Motivated by these empirical and theoretical observations, we propose federated learning with discrepancy-aware collaboration (FedDisco).

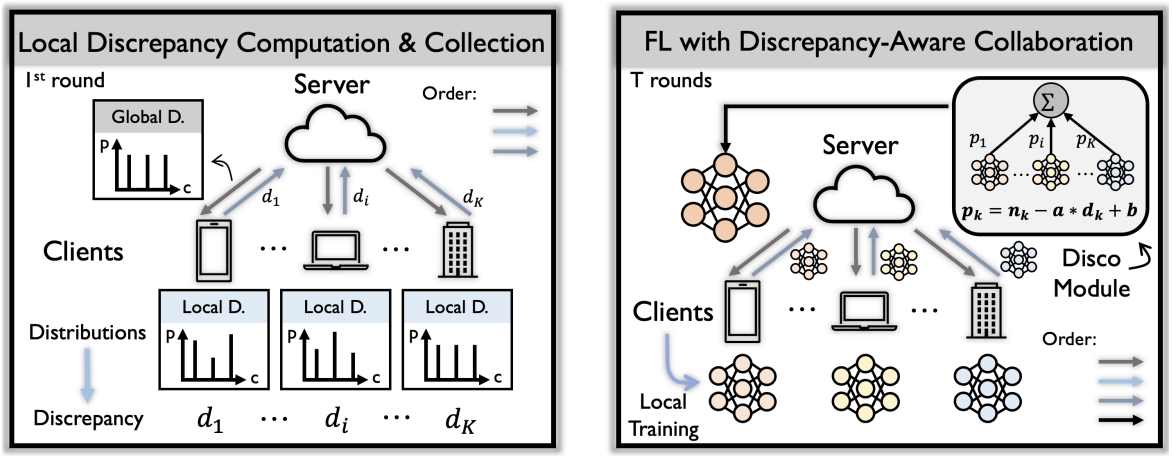


Figure 3. The overview of federated learning with discrepancy-aware collaboration. The left shows the acquisition of clients' discrepancy by comparing global and local category distribution, which is only required at the first round. The right shows the federated learning process with discrepancy-aware collaboration by adjusting each client's aggregation weight based on dataset size and discrepancy level.

5.1. Framework

FedDisco involves four key steps: local model training, local discrepancy computation, discrepancy-aware aggregation weight determination and global model aggregation. Steps 1 and 4 follows from the standard federated learning; meanwhile, Steps 2 and 3 integrates the discrepancy between local and global category distributions into aggregation weights. The overview is shown on the right of Figure 3. We also provide an algorithm flow in Algorithm 1.

Local model training. Each client k performs τ steps of local training based on dataset $\mathcal{B}_k$ and initial model $\mathbf{w}^{(t,0)}$ to obtain the trained model, denoted as $\mathbf{w}_k^{(t,\tau)} = LocalTrain(\mathcal{B}_k, \mathbf{w}^{(t,0)})$.

Local discrepancy computation. Each client needs to compute the discrepancy between its local category distribution and the hypothetically aggregated global category distribution. Here we consider that the global category distribution is uniform, since it naturally promotes the fairness across all categories and the generalization capability of the global model. Moreover, each local client can compute its discrepancy without additional data sharing, preventing information leakage of category distribution. Concretely, $\mathbf{D}_k$ and $\mathbf{T}$ denote local and global category distribution. Thus, we set all elements $\mathbf{T}_c = 1/C$ to treat all categories equally. Given the global distribution $\mathbf{T}$, each client k can calculate its local discrepancy level $d_k \in \mathbb{R}$ by evaluating the difference between these two distributions: $d_k = e(\mathbf{D}_k, \mathbf{T})$, where $e(\cdot)$ is a pre-defined metric function (e.g. L2 difference or KL-Divergence). Applying L2 difference corresponds to $d_k = \sqrt{\sum_{c=1}^{C}(\mathbf{D}_{k,c} - \mathbf{T}_c)^2}$. Finally, the server collects clients' discrepancy levels, see the left of Figure 3. Note that though we are more interested in uniform target distribution for the sake of category-level fairness, our method is also applicable to scenarios where target distribution is imbalanced; see experiments in Table 5.

Discrepancy-aware aggregation weight determination. Motivated by the previous empirical and theoretical observations, we propose to determine more distinguishing aggregation weights for each client k by leveraging dataset relative size n_k and local discrepancy level d_k (derived from (2)):

$$p_k = \frac{\mathrm{ReLU}(n_k - a \cdot d_k + b)}{\sum_{m=1}^{K} \mathrm{ReLU}(n_m - a \cdot d_m + b)}, \tag{3}$$

where $\mathrm{ReLU}(\cdot)$ is the relu function to take care of negative values (Fukushima, 1975), a is a hyper-parameter to balance n_k and d_k, b is another hyper-parameter to adjust the weight. This Disco aggregation can determine more distinguishing aggregation weights for clients by assigning larger weight for clients with larger dataset sizes and smaller local discrepancy levels.

Global model aggregation. The server conducts model aggregation based on the above discrepancy-aware aggregation weights. The global model is $\mathbf{w}^{(t+1)} = \sum_{k=1}^{K} p_k \mathbf{w}_k^{(t)}$, where $\mathbf{w}_k^{(t)}$ is the kth local model.

5.2. Discussions

Privacy. Our proposed FedDisco does not leak client's exact distribution (Figure 3) since the discrepancy is calculated at the client side. The server collects discrepancy level of clients, from which the exact category distribution can not be inversely inferred. This indicates that this process is more privacy-preserving compared with several existing works, such as FedFTG and CCVR (Zhang et al., 2022; Luo et al., 2021), which transmitting exact category distribution.

Communication & computation efficiency. Discrepancy communication is only required at the *first communication*

round. Besides, the discrepancy is only a numerical value, which is negligible compared with the model communication. The discrepancy calculation is only a simple operation between two vectors, which is negligible compared with model training. Moreover, FedDisco requires much less computation overhead and no computation burden to the server; while previous methods, such as FedDF (Lin et al., 2020) and FedFTG (Zhang et al., 2022), require additional model fine-tuning by simultaneously running K local models at the server side for *every communication round*.

Modularity. The modularity of FedDisco indicates its broad range of application. Typically, federated learning involves four steps: (1) global model downloading, (2) local updating, (3) local model uploading and (4) model aggregation. Our FedDisco focuses on step 4, which suggests that it can be easily combined with existing works conducting correction in step 2 and compression in step 1 and 3. Beyond this, it could also be incorporated with methods in step 4 by adjusting aggregation weight to be negatively correlated with discrepancy, such as conducting Disco re-weighting before normalization in FedNova (Wang et al., 2020b).

6. Related Works

Federated learning (FL) has been an emerging topic (McMahan et al., 2017; Hao et al., 2020). However, data distribution heterogeneity may significantly degrades FL's performance (Zhao et al., 2018; Kairouz et al., 2019; Li et al., 2019). Works that focus on this can be mainly divided into two directions: local and global model adjustment.

6.1. Local Model Adjustment

Methods in this direction conduct the adjustment on local model training at the client side, aiming at producing local models with smaller difference (Li et al., 2020a; Karimireddy et al., 2020). FedProx (Li et al., 2020a) regularizes the ℓ_2-distance between local and global model. FedDyn (Acar et al., 2020) proposes a dynamic regularizer to align the local and global solutions. SCAFFOLD (Karimireddy et al., 2020) and FedDC (Gao et al., 2022) introduce control variates to correct each client's gradient but require double communication cost. MOON (Li et al., 2021) aligns the features of local and global model; FedFM (Ye et al., 2022) aligns category-wise feature space across clients, contributing to more consistent feature spaces.

All of these methods simply assign aggregation weight based on local dataset size, which is not sufficient to distinguish each client's contribution. While our proposed FedDisco leverages both dataset size and discrepancy between local and global category distributions to determine more distinguishing aggregation weights, which has strong ability of plug-and-play that can be easily incorporated with these methods to further enhance the overall performance.

6.2. Global Model Adjustment

Methods in this direction conducts the adjustment on global model at the server side, aiming at produce a global model with better performance (Lin et al., 2020; Reddi et al., 2021; Li et al., 2023; Zhang et al., 2023; Fan et al., 2022). FedNova (Wang et al., 2020b) conducts re-weighting to target system heterogeneity while we conduct re-weighting based on local discrepancy level to target data heterogeneity. CCVR (Luo et al., 2021) conducts post-calibration, FedAvgM (Hsu et al., 2019) and FedOPT (Reddi et al., 2021) introduce global model momentum to stabilize FL training; while these methods are orthogonal to FedDisco and can be easily combined. FedDF (Lin et al., 2020) and FedFTG (Zhang et al., 2022) introduce an additional fine-tuning step to refine the global model; FedGen (Zhu et al., 2021) learns a feature generator for assisting local training. However, these methods require great computation capability of the server with much more computation cost (e.g. FedFTG (Zhang et al., 2022) requires twice computation cost compared with FedAvg (McMahan et al., 2017)). As a contrast, our FedDisco brings no computation burden to the server and works with negligible computation overhead.

7. Experiments

We show key experimental setups and results in this section. More details and results are in Appendix B.

7.1. Experimental Setup

Datasets. We consider five image classification datasets to cover medical, natural and artificial scenarios, including HAM10000 (Tschandl et al., 2018), CIFAR-10 & CIFAR-100 (Krizhevsky et al., 2009), CINIC-10 (Darlow et al., 2018) and Fashion-MNIST (Xiao et al., 2017); and AG News (Zhang et al., 2015), a text classification dataset.

Federated scenarios. We consider two data distribution heterogeneous settings, termed NIID-1 and NIID-2. NIID-1 follows Dirichlet distribution (Wang et al., 2020a; Li et al., 2021) Dir_β, where β (default 0.5) is an argument correlated with heterogeneity level. We consider 10 clients for NIID-1. NIID-2 is a more heterogeneous setting consists of 5 biased clients (each has data from $C/5$ categories (McMahan et al., 2017; Li et al., 2020a)) and 1 unbiased client (has all C categories), where C is the total category number.

Implementation details. The number of local epochs and batch size are 10 and 64, respectively. We run federated learning for 100 rounds. We use ResNet18 (He et al., 2016) for HAM10000, a simple CNN network for other image datasets and TextCNN (Zhang & Wallace, 2015) for AG News. We use SGD optimizer with a 0.01 learning rate.

Table 1. Accuracy comparisons (mean ± std on 5 trials, %) on several heterogeneous settings and datasets. Experiments show that FedDisco consistently outperforms these state-of-the-art methods.

Method	HAM10000	CIFAR-10		CINIC-10		Fashion-MNIST	
	NIID-1	NIID-1	NIID-2	NIID-1	NIID-2	NIID-1	NIID-2
FedAvg	42.54±0.59	68.47±0.20	65.60±0.16	54.24±0.18	50.35±0.43	89.26±0.09	86.46±0.03
FedAvgM	42.54±0.45	68.59±0.32	66.01±0.25	54.00±0.44	50.23±0.64	89.31±0.08	87.06±0.11
FedProx	44.76±1.35	69.33±0.49	65.61±0.31	56.38±0.35	50.30±0.44	89.28±0.12	87.24±0.21
SCAFFOLD	55.08±0.23	71.09±0.24	66.93±0.17	57.47±0.35	53.50±0.43	89.65±0.07	86.87±0.24
FedDyn	54.44±0.98	70.14±0.31	68.49±0.62	56.41±0.41	52.69±0.52	89.12±0.13	86.38±0.43
FedNova	44.92±0.59	68.57±0.19	65.61±0.39	54.27±0.20	50.37±0.73	89.05±0.08	86.47±0.20
MOON	45.87±0.90	67.42±0.14	66.25±0.35	52.08±0.20	50.42±0.56	89.29±0.04	86.44±0.04
FedDC	54.48±0.77	70.93±0.15	67.89±0.44	57.26±0.26	52.43±0.60	89.19±0.02	86.57±0.06
FedDisco	**59.05**±0.67	**72.05**±0.24	**69.85**±0.18	**58.07**±0.15	**53.84**±0.08	**89.74**±0.07	**87.85**±0.20

Table 2. Modularity. Each entry shows accuracy of baseline with Disco (accuracy **difference** compared with baseline without Disco in Table 1). Experiments show consistent improvements across settings, indicating the modularity of FedDisco.

+ Disco	HAM10000	CIFAR-10		CINIC-10		Fashion-MNIST	
	NIID-1	NIID-1	NIID-2	NIID-1	NIID-2	NIID-1	NIID-2
FedAvg	50.95 (**+8.41**)	70.05 (**+1.58**)	68.30 (**+2.70**)	54.81 (**+0.57**)	52.46 (**+2.11**)	89.56 (**+0.30**)	87.56 (**+1.10**)
FedAvgM	50.00 (**+7.46**)	70.07 (**+1.48**)	67.73 (**+1.72**)	54.69 (**+0.69**)	51.91 (**+1.68**)	89.36 (**+0.05**)	87.46 (**+0.40**)
FedProx	50.48 (**+5.72**)	70.68 (**+1.35**)	68.33 (**+2.72**)	56.93 (**+0.55**)	52.62 (**+2.32**)	89.58 (**+0.27**)	87.72 (**+0.48**)
SCAFFOLD	57.62 (**+2.54**)	71.70 (**+0.61**)	69.10 (**+2.17**)	58.05 (**+0.58**)	53.82 (**+0.32**)	89.74 (**+0.09**)	87.85 (**+0.98**)
FedDyn	59.05 (**+4.61**)	72.05 (**+1.91**)	69.85 (**+1.36**)	58.07 (**+1.66**)	53.84 (**+1.15**)	89.31 (**+0.19**)	87.18 (**+0.80**)
FedNova	51.43 (**+6.51**)	70.04 (**+1.47**)	67.83 (**+2.22**)	55.04 (**+0.77**)	52.23 (**+1.86**)	89.28 (**+0.23**)	87.52 (**+1.05**)
MOON	52.86 (**+6.99**)	68.79 (**+1.37**)	68.35 (**+2.10**)	53.26 (**+1.18**)	51.97 (**+1.55**)	89.50 (**+0.21**)	87.59 (**+1.15**)
FedDC	58.25 (**+3.77**)	71.96 (**+1.03**)	68.94 (**+1.05**)	57.70 (**+0.44**)	53.25 (**+0.82**)	89.51 (**+0.32**)	87.11 (**+0.54**)

We evaluate the accuracy on the global testing set. We use KL-Divergence to measure the discrepancy.

Baselines. We compare FedDisco with eight representative baselines. Among these, 1) FedAvg (McMahan et al., 2017) is the pioneering FL method; 2) FedProx (Li et al., 2020a), SCAFFOLD (Karimireddy et al., 2020), FedDyn (Acar et al., 2020), MOON (Li et al., 2021), and FedDC (Gao et al., 2022) focus on local model adjustment; 3) FedAvgM (Hsu et al., 2019) and FedNova (Wang et al., 2020b) focus on global model adjustment. The tuned hyper-parameters are shown in Appendix B.1.5.

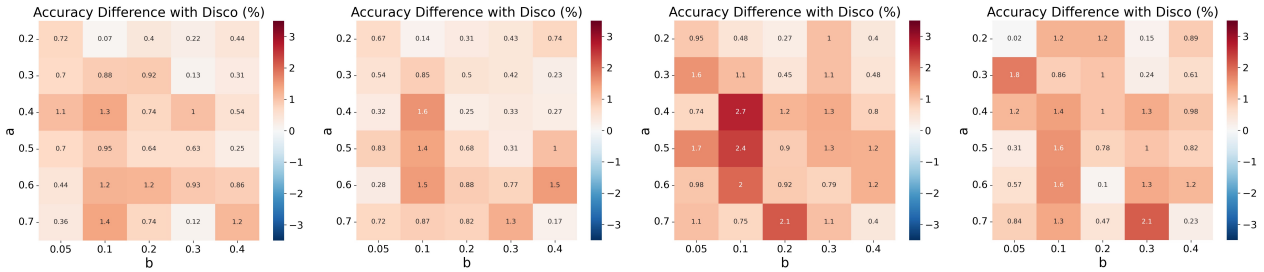
7.2. Main Results

On four standard datasets and two types of heterogeneity, we compare FedDisco with state-of-the-art algorithms, show the modularity of FedDisco, and explore FedDisco's broader scope of applications.

Performance: state-of-the-art accuracy. We compare the accuracy of several state-of-the-art methods and our proposed FedDisco on multiple heterogeneous settings and datasets in Table 1. Note that HAM10000 is an imbalanced dataset and thus is not applicable for NIID-2. The local training protocol used in FedDisco is FedDyn except SCAFFOLD for Fashion-MNIST. Experiments show that i) our proposed FedDisco consistently outperforms others across different settings, indicating the effectiveness of our proposed method; ii) FedDisco achieves significantly better on NIID-1 setting of HAM10000 ($\beta = 5$), which is a more difficult task for its severe heterogeneity and imbalance.

Modularity: improvements over baselines. One key advantage of our proposed FedDisco is its modularity, that is, it can be a plug-and-play module in many existing FL methods to further improve their performance. Following the experiments in Table 1, we report the accuracy of baselines combined with our Disco module and the accuracy difference (in parentheses) compared with baselines without Disco in Table 2 (see CIFAR-100 in Table 11). Experiments show that i) Disco consistently enhances the performance of state-of-the-art methods under different datasets and heterogeneity types; ii) for the most difficult task (i.e., HAM10000), FedDisco brings the largest performance improvement. Specifically, it achieves 19.8% relative accuracy improvement over FedAvg (McMahan et al., 2017).

Applicability to partial client participation scenarios. Partial client participation is a common scenario in cross-device FL applications where only a subset of clients are available in a specific FL round. To verify that our proposed FedDisco is applicable to this scenario, we sample 10 out of 60 clients for each round under NIID-2 on CIFAR-10, which consists of 50 biased clients and 10 unbiased clients. We report the averaged accuracy of last 10 rounds and the number



(a) MOON, NIID-1, CIFAR-10 (b) FedAvg, NIID-1, CIFAR-10 (c) FedAvg, NIID-2, CIFAR-10 (d) FedAvg, NIID-2, CINIC10

Figure 4. Ease of hyper-parameters tuning. Experiments show that our Disco works for a wide range of hyper-parameters.

Table 3. Performance under partial client participation scenario without (w.o.) and with our proposed Disco module. FedDisco not only brings accuracy improvement, but also speeds up training.

Method	Acc (%)		Rounds → 55%	
	W.o.	With Disco	W.o.	With Disco
FedAvg	54.27	61.59 (↑ **7.32**)	58	20 (↓ **65.52%**)
FedAvgM	57.75	60.65 (↑ **2.90**)	36	20 (↓ **44.44%**)
FedProx	55.50	60.19 (↑ **4.69**)	44	20 (↓ **54.55%**)
SCAFFOLD	60.43	64.10 (↑ **3.67**)	34	20 (↓ **41.18%**)
FedDyn	59.44	62.53 (↑ **3.09**)	33	27 (↓ **18.18%**)
FedNova	54.11	58.13 (↑ **4.02**)	61	24 (↓ **60.66%**)
MOON	54.30	58.79 (↑ **4.49**)	56	24 (↓ **57.14%**)
FedDC	61.18	62.90 (↑ **1.72**)	27	20 (↓ **25.93%**)

Table 4. Results on text classification dataset AG News (Zhang et al., 2015) under full and partial participation. FedDisco consistently brings accuracy improvement over baselines.

Disco?	Full		Partial	
	FedAvg	FedProx	FedAvg	FedProx
×	82.03	79.89	50.88	58.22
√	**84.09**	**80.81**	**57.71**	**62.86**

of rounds to reach target accuracy (55%) in Table 3. We see that Disco module significantly i) brings performance gain to baselines (up to 7.32%); ii) reduces the communication cost to reach a target accuracy (up to 65.52%). These results indicate that FedDisco not only improves the accuracy but also speeds up the training process under this scenario.

Applicability to text-modality scenarios. To verify that our proposed FedDisco can also be applied to text-modality, we explore on a text classification dataset, AG News (Tschandl et al., 2018) under full and partial participation scenarios. Results in Table 4 show that Disco still consistently improves the baselines on text modality and brings significant performance gain under partial client participation scenario.

Applicability to globally imbalanced scenarios. Here, we verify that FedDisco is also capable of globally imbalanced category distribution scenario, that is the hypothetically aggregated global data is imbalanced. We firstly allocate CIFAR-10 to each category c following an exponential distribution (Cui et al., 2019): $n_c = n_1 \rho^{-\frac{c-1}{C-1}}$, where ρ denotes the imbalance ratio and $C = 10$ is the total category number, $n_1 = 5000$. The dataset is then distributed to 10 clients as NIID-1. Note that $\rho = 1$ denotes globally balanced, $\rho = 20$ is the most imbalanced, where category 1 has 5000 samples while category 10 only has 250 samples. We consider two scenarios 1) the global category distribution is non-uniform while the distribution of test dataset is uniform; 2) the global category distribution and the distribution of test dataset are similar, and both non-uniform.

1) We conduct experiments on two typical baselines, FedAvg and FedDyn in Figure 5(a). Experiments show that our FedDisco consistently improves the baseline regardless of the globally imbalance level; see more results in Appendix B.2.

2) We conduct experiments under two scenarios ($\rho = 10$ and $\rho = 50$) and report the results in Table 5. Experiments show that FedDisco still performs the best when both global and test category distribution are non-uniform; see how to obtain global category distribution in a privacy-preserving way in Appendix A.3.

7.3. Numerical Simulation of Reformulation

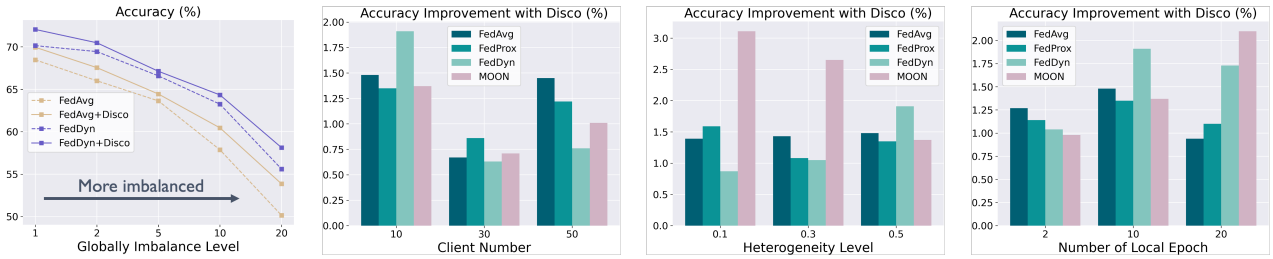
In Section 4, we minimize the reformulated the upper bound in Theorem 4.5 for obtaining a concise expression, which benefits the practical utility as the tuning efforts are mitigated. Here, we conduct numerical simulation to examine the optimization process of the original form of upper bound in Theorem 4.5 and the reformulated form in Equation (1). We run 400 steps of gradient descent on $\{p_k\}$ by minimizing the reformulated form, and record the resulting values of both reformulated and original form in Table 6. Results show that minimizing the reformulated form benefit minimizing the original form, verifying the rationality of such reformulation during derivation.

7.4. Ablation Study

Ease of hyper-parameter tuning. We tune a and b in (3) under four settings to illustrate the ease of hyper-parameter tuning for our proposed FedDisco. For more comprehensive understanding, we consider multiple scenarios, includ-

Table 5. Performance on scenario where global category distribution and test distribution are similar but both non-uniform. Experiments show that FedDisco is also applicable to scenarios where the global and test distributions are non-uniform.

METHOD	FEDAVG	FEDAVGM	FEDPROX	SCAFFOLD	FEDDYN	FEDNOVA	MOON	FEDDC	**FEDDISCO**
$\rho = 10$	69.78	69.25	69.27	71.48	69.29	68.44	68.29	70.00	**71.77**
$\rho = 50$	74.03	73.77	74.13	74.03	74.78	74.53	74.03	74.53	**76.06**



(a) Globally Imbalance Level (b) Client Number (c) Heterogeneity Level (d) Number of Local Epoch

Figure 5. Effects of four key FL arguments. Experiments show our Disco consistently brings performance improvement.

Table 6. Numerical simulation. Minimizing the reformulated form promotes minimizing the original form.

STEP	0	100	200	300	400
REFORMULATED	0.0684	0.0670	0.0663	0.0660	0.0658
ORIGINAL	0.0524	0.0513	0.0507	0.0504	0.0502

Table 7. Effects of discrepancy metric. FedDisco is robust to various choices of discrepancy metrics.

METRIC	L1	L2	COSINE	KL-DIVERGENCE
ACC	69.43	69.99	69.90	70.05

ing different baseline methods (FedAvg (McMahan et al., 2017) and MOON (Li et al., 2021)), heterogeneity types (NIID-1 and NIID-2) and datasets (CIFAR-10 and CINIC-10). We plot the accuracy difference brought by Disco for each group of hyper-parameters in Figure 4. By comparing three pairs: (a)&(b), (b)&(c) and (c)&(d), we see that i) for a wide range ($a \in [0.2, 0.7], b \in [0.05, 0.4]$), our Disco brings performance improvement for most cases regardless of the baseline method, heterogeneity type and dataset; ii) generally, $a = 0.4 \sim 0.6, b = 0.1$ is a safe choice, which leads to stable and great performance.

Effects of client number, heterogeneity level and local epoch. We tune three key arguments in FL, including client number ($K \in \{10, 30, 50\}$), heterogeneity level (smaller value corresponds to more severe heterogeneity, $\beta \in \{0.1, 0.3, 0.5\}$) and number of local epoch $E \in \{2, 10, 20\}$, and show the accuracy improvement brought by Disco in Figure 5(b), 5(c), and 5(d), respectively. Experiments show that our proposed Disco consistently brings performance improvement across different FL arguments.

Effects of different discrepancy metrics. Unless specified, we use the KL-Divergence throughout the paper. Though, Table 7 compares four discrepancy metrics under NIID-1 on CIFAR-10, including L1&L2 norm, cosine similarity and KL-Divergence. Experiments show that FedDisco with these metrics achieve similar performance, indicating its robustness to different discrepancy metrics.

8. Conclusions

This paper focuses on data heterogeneity issue in FL. Through empirical and theoretical explorations, we find that conventional dataset-size-based aggregation manner could be far from optimal. Addressing these, we introduce a discrepancy value as a complementary indicator and propose FedDisco, a novel FL algorithm that assigns larger aggregation weight to client with larger dataset size and smaller discrepancy. FedDisco introduces negligible computation and communication cost, and can be easily incorporated with many methods. Experiments show that FedDisco consistently achieves state-of-the-art performances.

Though this work mainly explores category-level heterogeneity, we may extend the idea of discrepancy-aware collaboration to other types of heterogeneity, such as feature-level heterogeneity for classification task and mask-level heterogeneity for segmentation task.

Acknowledgements

This research is supported by the National Key R&D Program of China under Grant 2021ZD0112801, NSFC under Grant 62171276 and the Science and Technology Commission of Shanghai Municipal under Grant 21511100900 and 22DZ2229005.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Darlow, L. N., Crowley, E. J., Antoniou, A., and Storkey, A. J. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

Ding, Y., Niu, C., Wu, F., Tang, S., Lyu, C., Chen, G., et al. Federated submodel optimization for hot and cold data features. *Advances in Neural Information Processing Systems*, 35:1–13, 2022.

Fan, Z., Wang, Y., Yao, J., Lyu, L., Zhang, Y., and Tian, Q. Fedskip: Combatting statistical heterogeneity with federated skip aggregation. *arXiv preprint arXiv:2212.07224*, 2022.

Fukushima, K. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20(3):121–136, 1975.

Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., and Xu, C.-Z. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10121, June 2022.

Hao, M., Li, H., Luo, X., Xu, G., Yang, H., and Liu, S. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 2020. doi: 10.1109/TII.2019.2945367.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

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

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

Jiménez, D. Dynamically weighted ensemble neural networks for classification. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 1, pp. 753–756. IEEE, 1998.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Kaissis, G. A., Makowski, M. R., Rückert, D., and Braren, R. F. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

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

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

Li, Q., He, B., and Song, D. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020a.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020b.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

Li, Z., Lin, T., Shang, X., and Wu, C. Revisiting weighted aggregation in federated learning with neural networks. *arXiv preprint arXiv:2302.10911*, 2023.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363, 2020.

Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., and Feng, J. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., and Fetaya, E. Multi-task learning as a bargaining game. In *International Conference on Machine Learning*, pp. 16428–16446. PMLR, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Shen, Z.-Q. and Kong, F.-S. Dynamically weighted ensemble neural networks for regression problems. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, volume 6, pp. 3492–3496. IEEE, 2004.

Singh, K., Sandhu, R. K., and Kumar, D. Comment volume prediction using neural networks and decision trees. In *IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015 (UKSim2015)*, 2015.

Tschandl, P., Rosendahl, C., and Kittler, H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=BkluqlSFDS.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

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

Ye, R., Ni, Z., Xu, C., Wang, J., Chen, S., and Eldar, Y. C. Fedfm: Anchor-based feature matching for data heterogeneity in federated learning. *arXiv preprint arXiv:2210.07615*, 2022.

Zhang, L., Shen, L., Ding, L., Tao, D., and Duan, L.-Y. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10174–10183, 2022.

Zhang, R., Xu, Q., Yao, J., Zhang, Y., Tian, Q., and Wang, Y. Federated domain generalization with generalization adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3954–3963, 2023.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhang, Y. and Wallace, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

Zhang, Y., Kang, B., Hooi, B., Yan, S., and Feng, J. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhu, Z., Hong, J., and Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.

Table 8. Notation descriptions.

Notation	Description
K	The total client number in FL system
τ	The number of SGD steps during local model training for each round
$\mathcal{B}_k$	Local private dataset of client k
$\mathbf{w}_k^{(t,r)}$	Local model of client k at round t and iteration r
$\mathbf{w}^{(t,0)}$	Global model at round t
$\mathbf{D}_k$	Local category distribution of client k
$\mathbf{D}_{k,c}$	The c-th element of $\mathbf{D}_k$
$\mathbf{T}$	Global category distribution
n_k	The relative dataset size of client k
p_k	The aggregation weight of client k
d_k	The discrepancy value of client k
$F_k(\mathbf{w})$	Local objective of client k
$F(\mathbf{w})$	Global objective of FL

Algorithm 1 FedDisco: Federated Learning with Discrepancy-Aware Collaboration

Initialization: Global model $\mathbf{w}^{(0,0)}$
for $k = 0$ **to** $K-1$ **do**
  Client sends discrepancy value d_k to server
end for
Server computes the aggregation weight p_k according to Equation (3)
for $t = 0$ **to** $T-1$ **do**
  Server sends global model $\mathbf{w}^{(t,0)}$ to each client
  for $k = 0$ **to** $K-1$ **do**
    $\mathbf{w}_k^{(t,\tau)} \leftarrow$ local model training for τ steps of SGD
    Client sends local model $\mathbf{w}_k^{(t,\tau)}$ to server
  end for
  Server aggregates local models $\mathbf{w}^{(t+1,0)} = \sum_{k=1}^{K} p_k \mathbf{w}_k^{(t,\tau)}$
end for

A. Methodology

A.1. Complementary Description

Notation table. For convenience, we provide a detailed notation descriptions in Table 8.

Algorithm table. We provide the overall algorithm in Algorithm 1.

A.2. Discussions

Connection with multi-task learning method, Nash-MTL (Navon et al., 2022). Nash-MTL focuses on aggregation weights of multiple tasks and we focus on aggregation weights of multiple clients. We will cite this paper in the revision. However, there are two major differences between Nash-MTL and our work. 1) Nash-MTL focuses on multi-task learning in a centralized manner while we focus on federated learning in a distributed manner. 2) The aggregation weights in Nash-MTL is learned through minimizing a pre-defined problem to search for Nash bargaining solution, which focuses on pair-wise gradient relationships among multiple tasks. In comparison, our aggregation weights is directly computed based on dataset size and discrepancy level, whose design is guided by our empirical and theoretical observation.

A.3. Obtaining Global Category Distribution in A Privacy-Preserving Manner

In Section 5, we regard the target distribution $\mathbf{T}$ as uniform since we want to emphasize category-level fairness. However, our method is also applicable in scenarios where the global category distribution and test distribution are both non-uniform; see results in Table 5. Here, we show that we can obtain the global category distribution in a privacy-preserving way.

Specifically, we can send each client's category distribution $\mathbf{D}_k$ (a vector) to the server using Secure Aggregation (Bonawitz et al., 2017) technique such that the server knows the actual global category distribution without knowing each client's actual category distribution. Then, this actual category distribution can be sent to each client and the discrepancy can be calculated (this process is efficient as it only requires once). As a simple example for the secure aggregation process, the distribution vectors of Client 1, 2 are $\mathbf{D}_1$ and $\mathbf{D}_2$. Client 1 adds an arbitrary noise vector $\mathbf{a}$ to $\mathbf{D}_1$ to obtain $\mathbf{D}_1 + \mathbf{a}$; while Client 2 substitutes $\mathbf{a}$ from $\mathbf{D}_2$ to obtain $\mathbf{D}_2 - \mathbf{a}$. These two transformed vectors are then sent to the server, where the server knows the sum of global category distribution without knowing the exact distribution of each client: $\mathbf{D}_1 + \mathbf{a} + \mathbf{D}_2 - \mathbf{a} = \mathbf{D}_1 + \mathbf{D}_2$.

B. Experiments

B.1. Implementation details

B.1.1. Environments

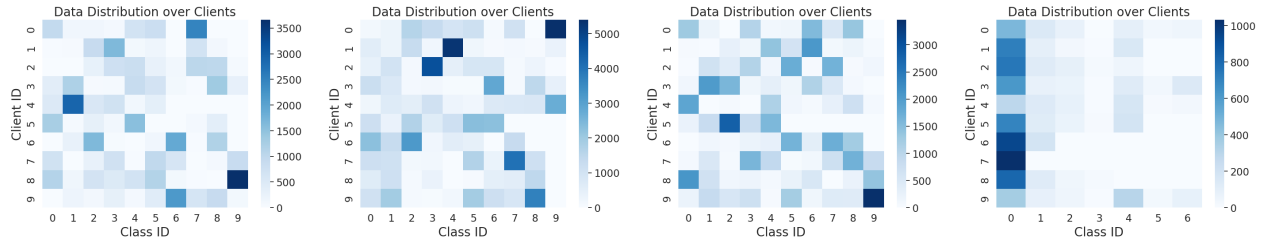
We run all methods by using Pytorch framework (Paszke et al., 2019) on a single NVIDIA GTX 3090 GPU. The memory occupation ranges from 2065MB to 4413MB for diverse datasets and methods.

B.1.2. Datasets

We use five image classification datasets, which cover medical, natural and artificial images. For medical image classification dataset, we consider HAM10000 (Codella et al., 2019; Tschandl et al., 2018), a 7 category classification task for pigmented skin lesions. For natual image classification dataset, we consider CIFAR-10, CIFAR-100 and CINIC-10 (Krizhevsky et al., 2009; Darlow et al., 2018), all of which are classification tasks for natural objects with 10, 100 and 10 categories, respectively. For artificial image classification dataset, we consider Fashion-MNIST (Xiao et al., 2017), a 10 category classification task for clothing. All these four public datasets can be downloaded online. Note that for HAM10000, we hold out a uniform testing set by allocating 30 samples for each category. We use one text classification dataset, AG News (Zhang et al., 2015), which is a 4 classification task.

B.1.3. Heterogeneity Level

Here, we illustrate the data distribution over categories and clients under NIID-1 setting of four datasets. As mentioned in the main paper, NIID-1 denotes the setting of Dirichlet distribution Dir_β. We set $\beta = 0.5$ for all datasets except HAM10000 and CIFAR-100. For HAM10000, all methods fail at NIID-1 scenario when β is too small since HAM10000 is an severely imbalanced dataset. Thus, we choose a moderated $\beta = 5.0$. We also choose $\beta = 5.0$ for CIFAR-100. We plot the NIID-1 data distribution over categories and clients on four datasets in Figure (6). Note that HAM10000 is globally imbalanced and it has the largest number of samples in class 0. For AG News, we partition the dataset to 5 and 50 clients for full and partial participation scenarios, where 80% biased clients have data samples from 2 categories and the other 20% uniform clients have data samples from 4 categories.



(a) CIFAR-10 (b) CINIC-10 (c) Fashion-MNIST (d) HAM10000

Figure 6. Data distribution over categories and clients under NIID-1 setting.

B.1.4. Models

For the CIFAR-10, CINIC-10 and Fashion-MNIST datasets, we use a simple CNN network as (Li et al., 2021; Luo et al., 2021). The network is sequentially consists of: 5×5 convolution layer, max-pooling layer, 5×5 convolution layer, three fully-connected layer with hidden size of 120, 84 and 10 respectively. For the HAM10000 and CIFAR-100 dataset, we use

Table 9. Classification accuracy (%) comparisons under globally imbalanced dataset scenario ($\rho = 10$). We highlight the **best** performance and ***second-best*** performance. SCAFFOLD and FedDyn performs the best while SCAFFOLD requires twice communication costs. Experiments show our proposed FedDisco consistently improves the performances of baselines, indicating FedDisco's applicability to this scenario.

METHOD	FEDAVG	FEDAVGM	FEDPROX	SCAFFOLD	FEDDYN	FEDNOVA	MOON	FEDDC
WITHOUT DISCO	57.86	57.53	57.78	63.78	63.24	59.64	57.74	61.74
WITH DISCO	60.45	60.05	60.33	**65.13**	***64.33***	60.77	59.46	63.13

ResNet18 (He et al., 2016) in Pytorch library. We replace the first 7×7 convolution layer with a 3×3 convolution layer and eliminate the first pooling layer. We also replace the batch normalization layer with group normalization layer as (Acar et al., 2020). For AG News (Zhang et al., 2015), we use TextCNN model (Zhang & Wallace, 2015) with a 32 hidden dimension.

B.1.5. HYPER-PARAMETERS

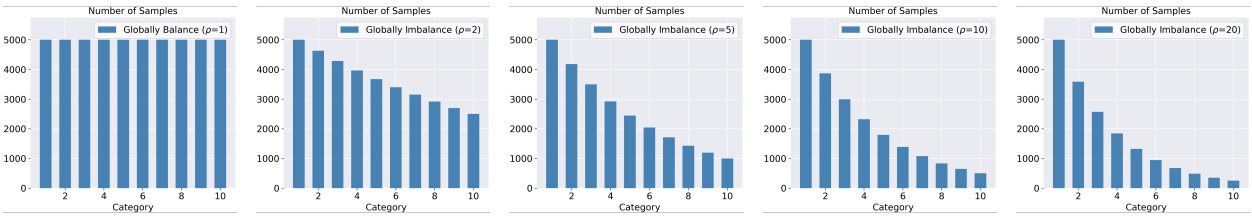
For all methods, we tune the hyper-parameter in a reasonable range and report the highest accuracy in the paper. For FedProx (Li et al., 2020a), we tune the hyper-parameter μ from $\{0.001, 0.01, 0.1, 1.0\}$. For FedAvgM (Hsu et al., 2019), we tune the hyper-parameter β from $\{0.3, 0.5, 0.7, 0.9\}$. For FedDyn (Acar et al., 2020), we tune the hyper-parameter α from $\{0.001, 0.01, 0.1, 1.0\}$. For MOON (Li et al., 2021), we tune the hyper-parameter μ from $\{0.01, 0.1, 0.5, 1.0, 5.0\}$. For FedDC (Gao et al., 2022), we tune the hyper-parameter α from $\{0.001, 0.01, 0.1, 1.0\}$.

The tuned best hyper-parameter for these methods are: $\mu = 0.01$ in FedProx (Li et al., 2020a), $\beta = 0.5$ in FedAvgM (Hsu et al., 2019), $\alpha = 0.01$ in FedDyn (Acar et al., 2020), $\mu = 0.1$ in MOON (Li et al., 2021), $\alpha = 0.01$ in FedDC (Gao et al., 2022).

B.2. Globally Imbalanced Scenario: Accuracy and Fairness

We show that our proposed FedDisco is also capable of globally imbalanced category distribution scenario, that is the hypothetically aggregated global data is imbalanced. In the main paper, we show the performance of FedAvg (McMahan et al., 2017) and FedDyn (Acar et al., 2020) with and without Disco for different globally imbalance level ($\rho = 1, 2, 5, 10, 20$) in Figure 5 (a).

We firstly visualize the hypothetically aggregated global data distribution in Figure 7, where x-axis denotes the category and y-axis denotes the number of data samples of the corresponding category. We plot the data distribution over categories for different globally imbalance levels (ρ). $\rho = 1$ denotes balanced situation and $\rho = 20$ denotes the most severe imbalanced situation. Our experiments in the Figure 5 (a) of the main paper show that our FedDisco works for all these globally imbalance levels.



(a) Balanced: $\rho = 1$ (b) Imbalanced: $\rho = 2$ (c) Imbalanced: $\rho = 5$ (d) Imbalanced: $\rho = 10$ (e) Imbalanced: $\rho = 20$

Figure 7. Hypothetically aggregated global data distribution over categories for different globally imbalance level (ρ). The globally imbalance level increases from the left to the right, where $\rho = 1$ denotes balanced situation and $\rho = 20$ denotes the most severe imbalanced situation.

Next, we provide the performance of more baselines when $\rho = 10$ in Table 9. Experiments show that our proposed FedDisco consistently improves over baselines under this globally imbalanced scenario, indicating its applicability to this scenario.

We also explore the performance of global model on each individual category in detail in Figure 8. Here, we use FedAvg (McMahan et al., 2017) as an example. Following (Zhang et al., 2021), we define and evaluate on three subsets:

Table 10. Performance under partial client participation scenario. Mean ± std is evaluated over last 10 rounds. Rounds: rounds to reach target accuracy (55%). Higher mean, lower std and smaller rounds represent better performance. We highlight the difference of mean accuracy brought by Disco in parentheses. We also highlight the reduced number of rounds together with the proportion of reduction brought by Disco in parentheses. Methods with Disco achieve higher accuracy, more stable performance and faster training. Specifically, FedAvg with Disco achieves 7.32% accuracy improvement and requires 38 less rounds to achieve the target accuracy, which saves 65.52% training time and communication cost.

Method	Mean ± std (%)		Rounds	
	Without Disco	With Disco	Without Disco	With Disco
FedAvg	54.27 ±3.44	61.59 ±1.84 (↑ **7.32**)	58	20 (↓ **38**, ↓ **65.52%**)
FedAvgM	57.75 ±1.92	60.65 ±1.32 (↑ **2.90**)	36	20 (↓ **16**, ↓ **44.44%**)
FedProx	55.50 ±3.02	60.19 ±2.47 (↑ **4.69**)	44	20 (↓ **24**, ↓ **54.55%**)
SCAFFOLD	60.43 ±1.96	64.10 ±0.76 (↑ **3.67**)	34	20 (↓ **14**, ↓ **41.18%**)
FedDyn	59.44 ±2.59	62.53 ±2.34 (↑ **3.09**)	33	27 (↓ **06**, ↓ **18.18%**)
FedNova	54.11 ±3.83	58.13 ±2.72 (↑ **4.02**)	61	24 (↓ **37**, ↓ **60.66%**)
MOON	54.30 ±3.97	58.79 ±3.07 (↑ **4.49**)	56	24 (↓ **32**, ↓ **57.14%**)

Head (category 1 - 4), Middle (category 5 - 7) and Tail (category 8 - 10). Additionally, we report the averaged accuracy over all categories and standard deviation across categories. We see that i) FedDisco achieves comparable performance (difference within 0.75%) on Head and Middle classes and significantly higher performance (nearly 8% improvement) on Tail classes. This suggests that FedAvg is severally biased to Head classes while FedDisco can mitigate this bias. ii) FedDisco achieves higher averaged accuracy with smaller standard deviation, which means FedDisco can simultaneous enhance overall performance and promote fairness across categories.

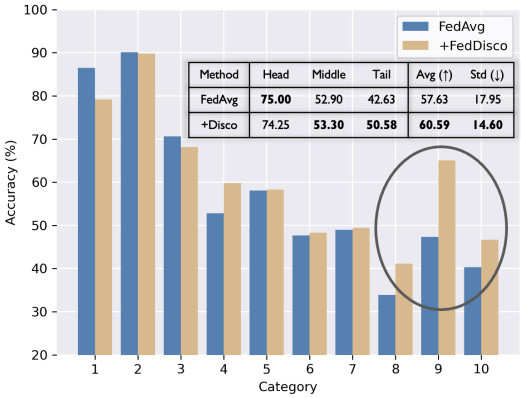


Figure 8. Accuracy comparison of each category on CIFAR-10. FedDisco significantly enhance the performance of Tail classes without severely affecting the Head classes. FedDisco simultaneously enhance overall performance (Higher Avg) and fairness (Lower Std) over FedAvg.

B.3. Partial Client Participation Scenario: Accuracy, Stability and Training Speed

In practice, clients may participate only when they are in reliable power and Wi-Fi condition, indicating that only a subset of clients participate in each round (partial client participation). We conduct experiments on CIFAR-10 (Krizhevsky et al., 2009) where there are 40 biased clients and 10 unbiased clients. In each round, we randomly sample 10 clients to participate.

We show results from two aspects in Table 10, including the mean and standard deviation (std) of accuracy of last 10 rounds and rounds required to reach target accuracy (55%). Experiments show that methods with our Disco achieve i) consistently higher mean and smaller std of accuracy, indicating better and more stable performance over rounds, ii) fewer rounds to achieve target accuracy, indicating faster training speed, which is friendly to save computation and communication cost. *Specifically, for the basic method FedAvg, FedAvg with Disco achieves* 7.32% *accuracy improvement and requires* 38 *less rounds to achieve the target accuracy, which saves* 65.52% *training time and communication cost.*

Table 11. Classification accuracy (%) comparisons on CIFAR-100 dataset under NIID-1 scenario. FedDyn with Disco is highlighted for **best** performance. Experiments show our proposed FedDisco consistently improves the performances of baselines on CIFAR-100.

METHOD	FEDAVG	FEDAVGM	FEDPROX	SCAFFOLD	FEDDYN	FEDNOVA	MOON	FEDDC
WITHOUT DISCO	57.28	56.27	58.99	61.39	62.29	57.59	58.01	62.20
WITH DISCO	58.28	56.73	59.29	61.90	**62.83**	58.02	58.16	62.61

Table 12. Comparison of accuracy variance across clients at different round. Lower denotes more fair.

ROUND	1	10	20	30	40
FEDAVG	178.05	226.38	**73.65**	94.49	89.97
FEDDISCO	**167.40**	**110.63**	74.70	**84.72**	**73.45**

B.4. Experimental results on CIFAR-100

We provide the results of several baselines on CIFAR-100 (Krizhevsky et al., 2009) under NIID-1 setting in Table 11. Experiments show that our proposed FedDisco consistently brings gains on CIFAR-100 dataset, verifying its modularity performance.

B.5. Effects on Client-Level Fairness

In this paper, we focus on the generalization ability and promote category-level fairness, where the corresponding global distribution is uniform. Client-level fairness is another interesting and important research topic in FL (Li et al., 2020b). Here, we explore from this aspect for more comprehensive understanding.

Note that FedDisco does not hurt client-level fairness for two reasons. First, when a client's aggregation weight is high, this client has a uniform training distribution, which can naturally benefit all the clients more or less. If we do the opposite thing and assign a high aggregation weight to a client whose training category distribution is highly skewed, this only benefits this single client, hurts the other clients and violates the client-level fairness. Second, each client's test data distribution could be close to uniform distribution, even though its training data distribution is far from uniform. One important aim of federated learning is to enable clients with biased and limited training data to be aware of global distribution.

To validate that FedDisco does not hurt client-level fairness, we run 40 rounds of FL on CIFAR-10 and record the variance of test accuracy across clients. This accuracy variance is often used for evaluation of fairness (Li et al., 2020b). Small variance denotes that the test accuracies of clients are similar, indicating high fairness. Thus, a lower variance denotes higher fairness. Table 12 compares the accuracy variance of FedAvg and the accuracy variance after applying FedDisco to FedAvg. From the table, we see that the variance of FedDisco is comparable or lower than FedAvg, indicating that FedDisco can potentially enhance the client-level fairness.

B.6. Comparison with Equal Aggregation Weights

Following the experiments in Table 1, Table 13 compares FedDisco with FedAvg with equal aggregation weights (i.e., $p_k = 1/K$). From the table, we see that FedDisco performs significantly better across different settings.

B.7. Experiments on Device Heterogeneity

For device heterogeneity where there are stragglers, we conduct experiments with the NIID-2 setting on CIFAR-10 following the setting in (Wang et al., 2020b), where we uniformly sample iteration numbers for each client in the range of 50 to 500 for each round. Results show that FedAvg achieves 60.21% while **FedDisco achieves** 63.20%**, indicating that FedDisco can still bring performance improvement under setting of stragglers.**

Device heterogeneity is an orthogonal issue to distribution heterogeneity. As these two issues can be concurrent in practice, FedDisco can still play a role in enhancing overall performance. More importantly, device heterogeneity can even exacerbate the effect of distribution heterogeneity. For example, if Client A has a smaller discrepancy level and Client B has a larger discrepancy level. When Client A is a straggler that performs fewer iterations while Client B performs much more iterations, it is more critical to enhance the aggregation weight of Client A, otherwise the FL system will be dominated by Client B and

Table 13. Comparison between FedDisco and FedAvg with equal aggregation weights $p_k = 1/K$. FedDisco performs significantly better across different datasets and heterogeneity types.

METHOD	HAM10000	CIFAR-10		CINIC-10		FASHION-MNIST	
	NIID-1	NIID-1	NIID-2	NIID-1	NIID-2	NIID-1	NIID-2
FEDAVG ($p_k = 1/K$)	44.76	68.11	65.60	54.27	50.35	89.35	86.46
FEDAVG+DISCO	**50.95**	**70.05**	**68.30**	**54.81**	**52.46**	**89.56**	**87.56**

the global model will be biased. Further, we can combine algorithms (e.g., FedNova (Wang et al., 2020b)) that specifically focuses on device heterogeneity with ours to further enhance performance.

B.8. Other Experiments

1) The idea of discrepancy-aware collaboration can be extended to otehr tasks such as regression. Here, we conduct experiments on regression dataset Prediction of Facebook Comment (Singh et al., 2015) (predicting the number of comments given a post). We pre-define several categories for this regression task, where each category covers regression labels in some specific range. For example, we group those samples with regression labels ranging from 1 to 10 as category 1. With this strategy, we can obtain a distribution vector as we do for classification tasks. Note that such operation is only conducted for obtaining a distribution vector, we still use the regression labels for model training. It turns out that the test loss of FedAvg (McMahan et al., 2017) is 0.432 and FedDisco achieves 0.407, indicating our FedDisco can also handle continuous label distributions.

2) Here, we consider feature-level heterogeneity. We conduct experiments on FEMNIST from LEAF benchmark (Caldas et al., 2018) following (Li et al., 2020a), where both feature-level and category-level heterogeneity exist. We see that FedAvg achieves 76.40% accuracy while FedDisco achieves 78.36% accuracy.

C. Theoretical Analysis

C.1. Preliminaries

The global objective function is $F(\mathbf{w}) = \sum_{k=1}^{N} p_k F_k(\mathbf{w})$, where $\sum_{k=1}^{K} p_k = 1$. For ease of writing, we use $g_k(\mathbf{w})$ to denote mini-batch gradient $g_k(\mathbf{w}|\xi)$ and $\nabla F_k(\mathbf{w})$ to denote full-batch gradient, where ξ is a mini-batch sampled from dataset. We further define the following two notions:

$$\text{Averaged Mini-batch Gradient:} \quad \mathbf{d}_k = \frac{1}{\tau} \sum_{r=0}^{\tau-1} g_k(\mathbf{w}_k^{(t,r)}), \tag{4}$$

$$\text{Averaged Full-batch Gradient:} \quad \mathbf{h}_k = \frac{1}{\tau} \sum_{r=0}^{\tau-1} \nabla F_k(\mathbf{w}_k^{(t,r)}). \tag{5}$$

Then, the update of the global model between two rounds is as follows:

$$\mathbf{w}^{(t+1,0)} - \mathbf{w}^{(t,0)} = -\tau\eta \sum_{k=1}^{K} p_k \mathbf{d}_k. \tag{6}$$

Here, we presents a key lemma and defer its proof to section C.3.

Lemma C.1. *Suppose $\{A_t\}_{t=1}^{T}$ is a sequence of random matrices and follows $\mathbb{E}[A_t|A_{t-1}, A_{t-2}, ..., A_1] = \mathbf{0}$, then*

$$\mathbb{E}\left[\left\|\sum_{t=1}^{T} A_t\right\|_F^2\right] = \sum_{t=1}^{T} \mathbb{E}\left[\|A_t\|_F^2\right]$$

C.2. Proof of Theorem 4.5

According to the Lipschitz-smooth assumption in Assumption 4.1, we have its equivalent form (Bottou et al., 2018)

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{w}^{(t+1,0)})\right] - F(\mathbf{w}^{(t,0)}) \\
&\leq \mathbb{E}\left[\left\langle \nabla F(\mathbf{w}^{(t,0)}), \mathbf{w}^{(t+1,0)} - \mathbf{w}^{(t,0)} \right\rangle\right] - \frac{L}{2}\mathbb{E}\left[\left\|\mathbf{w}^{(t+1,0)} - \mathbf{w}^{(t,0)}\right\|^2\right] && (7)\\
&= -\tau\eta \underbrace{\mathbb{E}\left[\left\langle \nabla F(\mathbf{w}^{(t,0)}), \sum_{k=1}^{K} p_k \mathbf{d}_k \right\rangle\right]}_{N_1} + \frac{L\tau^2\eta^2}{2} \underbrace{\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{d}_k\right\|^2\right]}_{N_2}, && (8)
\end{aligned}$$

where the expectation is taken over mini-batches $\xi_k^{(t,r)}, \forall k \in 1, 2, ..., K, r \in 0, 1, ..., \tau - 1$.

C.2.1. Bounding N_1 in (8)

$$\begin{aligned}
N_1 &= \mathbb{E}\left[\left\langle \nabla F(\mathbf{w}^{(t,0)}), \sum_{k=1}^{K} p_k(\mathbf{d}_k - \mathbf{h}_k) \right\rangle\right] + \mathbb{E}\left[\left\langle \nabla F(\mathbf{w}^{(t,0)}), \sum_{k=1}^{K} p_k \mathbf{h}_k \right\rangle\right] && (9)\\
&= \mathbb{E}\left[\left\langle \nabla F(\mathbf{w}^{(t,0)}), \sum_{k=1}^{K} p_k \mathbf{h}_k \right\rangle\right] && (10)\\
&= \frac{1}{2}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + \frac{1}{2}\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla F(\mathbf{w}^{(t,0)}) - \sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right], && (11)
\end{aligned}$$

where (10) uses the unbiased gradient assumption in Assumption 4.3, such that $\mathbb{E}[\mathbf{d}_k - \mathbf{h}_k] = \mathbf{h}_k - \mathbf{h}_k = \mathbf{0}$. (11) uses the fact that $2\langle a, b\rangle = \|a\|^2 + \|b\|^2 - \|a - b\|^2$.

C.2.2. Bounding N_2 in (8)

$$\begin{aligned}
N_2 &= \mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k(\mathbf{d}_k - \mathbf{h}_k) + \sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (12)\\
&\leq 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k(\mathbf{d}_k - \mathbf{h}_k)\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (13)\\
&= 2\sum_{k=1}^{K} p_k^2 \mathbb{E}\left[\|\mathbf{d}_k - \mathbf{h}_k\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (14)\\
&= \frac{2}{\tau^2}\sum_{k=1}^{K} p_k^2 \mathbb{E}\left[\left\|\sum_{r=0}^{\tau-1}(g_k(\mathbf{w}_k^{(t,r)}) - \nabla F_k(\mathbf{w}_k^{(t,r)}))\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (15)\\
&= \frac{2}{\tau^2}\sum_{k=1}^{K} p_k^2 \sum_{r=0}^{\tau-1} \mathbb{E}\left[\left\|g_k(\mathbf{w}_k^{(t,r)}) - \nabla F_k(\mathbf{w}_k^{(t,r)})\right\|^2\right] + 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (16)\\
&\leq \frac{2\sigma^2}{\tau}\sum_{k=1}^{K} p_k^2 + 2\mathbb{E}\left[\left\|\sum_{k=1}^{K} p_k \mathbf{h}_k\right\|^2\right] && (17)
\end{aligned}$$

where (13) follows $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, (14) uses the fact that clients are independent to each other so that $\mathbb{E}\langle \mathbf{d}_k - \mathbf{h}_k, \mathbf{d}_n - \mathbf{h}_n \rangle = 0, \forall k \neq n$. (16) uses Lemma C.1 and (17) uses bounded variance assumption in Assumption 4.3.

Plug (11) and (17) back into (8), we have

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{w}^{(t+1,0)})\right] - F(\mathbf{w}^{(t,0)}) \\
\leq& -\frac{\tau\eta}{2}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 - \frac{\tau\eta}{2}(1-2\tau\eta L)\mathbb{E}\left[\left\|\sum_{k=1}^K p_k \mathbf{h}_k\right\|^2\right] + L\tau\eta^2\sigma^2\sum_{k=1}^K p_k^2 + \frac{\tau\eta}{2}\underbrace{\mathbb{E}\left[\left\|\nabla F(\mathbf{w}^{(t,0)}) - \sum_{k=1}^K p_k \mathbf{h}_k\right\|^2\right]}_{N_3}.
\end{aligned} \tag{18}$$

C.2.3. Bounding N_3 in (18)

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F(\mathbf{w}^{(t,0)}) - \sum_{k=1}^K p_k \mathbf{h}_k\right\|^2\right] \\
=&\mathbb{E}\left[\left\|\sum_{k=1}^K (n_k - p_k)\nabla F_k(\mathbf{w}^{(t,0)}) + \sum_{k=1}^K p_k\left(\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right)\right\|^2\right] && (19)\\
\leq& 2\left\|\sum_{k=1}^K (n_k - p_k)\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2 + 2\left\|\sum_{k=1}^K p_k\left(\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right)\right\|^2 && (20)\\
\leq& 2\left[\sum_{k=1}^K (n_k - p_k)^2\right]\left[\sum_{k=1}^K \left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right] + 2\left\|\sum_{k=1}^K p_k\left(\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right)\right\|^2 && (21)\\
\leq& 2\left[\sum_{k=1}^K (n_k - p_k)^2\right]\left[K\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + B\sum_{k=1}^K d_k\right] + 2\left\|\sum_{k=1}^K p_k\left(\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right)\right\|^2, && (22)
\end{aligned}$$

where (20) follows $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, (21) follows Cauchy–Schwarz inequality, (22) uses the bounded similarity assumption in Assumption 4.4.

We use W_D to denote $2K\left[\sum_{k=1}^K (n_k - p_k)^2\right]$. When $1 - 2\tau\eta L \geq 0$, we have

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{w}^{(t+1,0)})\right] - F(\mathbf{w}^{(t,0)}) \\
\leq& -\frac{\tau\eta(1-W_D)}{2}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + L\tau\eta^2\sigma^2\sum_{k=1}^K p_k^2 + \frac{\tau\eta W_D B}{2K}\sum_{k=1}^K d_k + \tau\eta\mathbb{E}\left[\left\|\sum_{k=1}^K p_k\left(\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right)\right\|^2\right] && (23)\\
\leq& -\frac{\tau\eta(1-W_D)}{2}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + L\tau\eta^2\sigma^2\sum_{k=1}^K p_k^2 + \frac{\tau\eta W_D B}{2K}\sum_{k=1}^K d_k + \tau\eta\sum_{k=1}^K p_k\underbrace{\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right\|^2\right]}_{N_4}, && (24)
\end{aligned}$$

where (24) uses Jensen's Inequality $\left\|\sum_{k=1}^K p_k x_k\right\|^2 \leq \sum_{k=1}^K p_k \|x_k\|^2$.

C.2.4. BOUNDING N_4 IN (24)

$$\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right\|^2\right] = \mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \frac{1}{\tau}\sum_{r=0}^{\tau-1}\nabla F_k(\mathbf{w}_k^{(t,r)})\right\|^2\right] \tag{25}$$

$$= \mathbb{E}\left[\left\|\frac{1}{\tau}\sum_{r=0}^{\tau-1}(\nabla F_k(\mathbf{w}^{(t,0)}) - \nabla F_k(\mathbf{w}_k^{(t,r)}))\right\|^2\right] \tag{26}$$

$$\leq \frac{1}{\tau}\sum_{r=0}^{\tau-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \nabla F_k(\mathbf{w}_k^{(t,r)})\right\|^2\right] \tag{27}$$

$$\leq \frac{L^2}{\tau}\sum_{r=0}^{\tau-1}\underbrace{\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,r)}\right\|^2\right]}_{N_5}, \tag{28}$$

where (27) uses Jensen's Inequality and (28) follows Lipschitz-smooth property.

C.2.5. BOUNDING N_5 IN (34)

$$\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,r)}\right\|^2\right] = \eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{r-1} g_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{29}$$

$$\leq 2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{r-1}\left(g_k(\mathbf{w}_k^{(t,s)}) - \nabla F_k(\mathbf{w}_k^{(t,s)})\right)\right\|^2\right] + 2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{r-1}\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{30}$$

$$= 2\eta^2\sum_{s=0}^{r-1}\mathbb{E}\left[\left\|g_k(\mathbf{w}_k^{(t,s)}) - \nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] + 2\eta^2\mathbb{E}\left[\left\|\sum_{s=0}^{r-1}\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{31}$$

$$\leq 2r\eta^2\sigma^2 + 2\eta^2\mathbb{E}\left[\left\|r\sum_{s=0}^{r-1}\frac{1}{r}\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{32}$$

$$\leq 2r\eta^2\sigma^2 + 2r\eta^2\sum_{s=0}^{r-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{33}$$

$$\leq 2r\eta^2\sigma^2 + 2r\eta^2\sum_{s=0}^{\tau-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \tag{34}$$

where (30) uses $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, (31) uses Lemma C.1, (32) uses the bounded variance assumption in Assumption 4.3, (33) uses Jensen's Inequality.

Plug (34) back into (28) and use this equation $\sum_{r=0}^{\tau-1} r = \frac{\tau(\tau-1)}{2}$, we have

$$\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right\|^2\right] \leq \frac{L^2}{\tau}\sum_{r=0}^{\tau-1}\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,r)}\right\|^2\right] \tag{35}$$

$$\leq (\tau-1)L^2\eta^2\sigma^2 + (\tau-1)L^2\eta^2\sum_{s=0}^{\tau-1}\underbrace{\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right]}_{N_6}, \tag{36}$$

where N_6 in (36) can be further bounded.

C.2.6. BOUNDING N_6 IN (36)

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}_k^{(t,s)})\right\|^2\right] \\
\leq & 2\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}_k^{(t,s)}) - \nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right] + 2\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right] && (37) \\
\leq & 2L^2\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,s)}\right\|^2\right] + 2\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right], && (38)
\end{aligned}
$$

where (37) uses $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, (38) uses Lipschitz-smooth property. Plug (38) back to (36), we have

$$
\begin{aligned}
&\frac{L^2}{\tau}\sum_{r=0}^{\tau-1}\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,r)}\right\|^2\right] \\
\leq & (\tau-1)L^2\eta^2\sigma^2 + 2(\tau-1)\eta^2L^4\sum_{s=0}^{\tau-1}\mathbb{E}\left[\left\|\mathbf{w}_k^{(t,0)} - \mathbf{w}^{(t,s)}\right\|^2\right] \\
& + 2(\tau-1)\eta^2L^2\sum_{s=0}^{\tau-1}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right] && (39)
\end{aligned}
$$

After rearranging, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right\|^2\right] \\
\leq & \frac{L^2}{\tau}\sum_{r=0}^{\tau-1}\mathbb{E}\left[\left\|\mathbf{w}^{(t,0)} - \mathbf{w}_k^{(t,r)}\right\|^2\right] && (40) \\
\leq & \frac{(\tau-1)\eta^2\sigma^2L^2}{1-2\tau(\tau-1)\eta^2L^2} + \frac{2\tau(\tau-1)\eta^2L^2}{1-2\tau(\tau-1)\eta^2L^2}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right] && (41) \\
= & \frac{(\tau-1)\eta^2\sigma^2L^2}{1-A} + \frac{A}{1-A}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right], && (42)
\end{aligned}
$$

where we define $A = 2\tau(\tau-1)\eta^2L^2 < 1$. Then, the last term in (24) is bounded by

$$
\begin{aligned}
&\tau\eta\sum_{k=1}^{K}p_k\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)}) - \mathbf{h}_k\right\|^2\right] \\
\leq & \tau\eta\sum_{k=1}^{K}\left\{p_k\left[\frac{(\tau-1)\eta^2\sigma^2L^2}{1-A} + \frac{A}{1-A}\mathbb{E}\left[\left\|\nabla F_k(\mathbf{w}^{(t,0)})\right\|^2\right]\right]\right\} && (43) \\
\leq & \frac{\tau(\tau-1)\sigma^2L^2\eta^3}{1-A} + \frac{\tau\eta A}{1-A}\mathbb{E}\left[\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2\right] + \frac{\tau\eta AB}{1-A}\sum_{k=1}^{K}p_kd_k, && (44)
\end{aligned}
$$

where (44) follows bounded dissimilarity assumption in Assumption 4.4. Plug (44) back to (24), we have

$$\begin{aligned}
&\mathbb{E}\left[F(\mathbf{w}^{(t+1,0)})\right] - F(\mathbf{w}^{(t,0)}) \\
\leq &-\frac{\tau\eta(1-W_D)}{2}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + L\tau\eta^2\sigma^2\sum_{k=1}^{K}p_k^2 + \frac{\tau\eta W_D B}{2K}\sum_{k=1}^{K}d_k \\
&+\frac{\tau(\tau-1)\sigma^2L^2\eta^3}{1-A} + \frac{\tau\eta A}{1-A}\mathbb{E}\left[\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2\right] + \frac{\tau\eta AB}{1-A}\sum_{k=1}^{K}p_k d_k \qquad (45)\\
= &-\frac{\tau\eta}{2}(1-W_D-\frac{2A}{1-A})\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 + L\tau\eta^2\sigma^2\sum_{k=1}^{K}p_k^2 + \frac{\tau\eta W_D B}{2K}\sum_{k=1}^{K}d_k + \frac{\tau(\tau-1)\sigma^2L^2\eta^3}{1-A} + \frac{\tau\eta AB}{1-A}\sum_{k=1}^{K}p_k d_k \qquad (46)
\end{aligned}$$

Finally, by taking the average expectation across all rounds, we finish the proof of Theorem 4.5

$$\min_t \mathbb{E}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla F(\mathbf{w}^{(t,0)})\right\|^2 \qquad (47)$$

$$\begin{aligned}
\leq &\frac{2(1-A)\left(F(\mathbf{w}^{(0,0)})-F_{inf}\right)}{\tau\eta T\left[1-3A-W_D(1-A)\right]} + \frac{(1-A)W_D B\sum_{k=1}^{K}d_k}{\left[1-3A-W_D(1-A)\right]K} \\
&+\frac{2(1-A)L\eta\sigma^2\sum_{k=1}^{K}p_k^2}{\left[1-3A-W_D(1-A)\right]} + \frac{2(\tau-1)\sigma^2L^2\eta^2}{\left[1-3A-W_D(1-A)\right]} + \frac{2AB\sum_{k=1}^{K}p_k d_k}{\left[1-3A-W_D(1-A)\right]} \qquad (48)\\
= &\frac{1}{1-3A-W_D(1-A)}\Bigg(\underbrace{\frac{2(1-A)\left(F(\mathbf{w}^{(0,0)})-F_{inf}\right)}{\tau\eta T}}_{T_1} + \underbrace{\frac{(1-A)W_D B}{K}\sum_{k=1}^{K}d_k}_{T_2} \\
&+\underbrace{2(1-A)L\eta\sigma^2\sum_{k=1}^{K}p_k^2}_{T_3} + \underbrace{2(\tau-1)\sigma^2L^2\eta^2}_{T_4} + \underbrace{2AB\sum_{k=1}^{K}p_k d_k}_{T_5}\Bigg), \qquad (49)
\end{aligned}$$

where $W_D = 2K\left[\sum_{k=1}^{K}(n_k-p_k)^2\right]$, p_k is the aggregation weight, n_k is the dataset relative size, d_k is the discrepancy level, $A = 2\tau(\tau-1)\eta^2L^2 < 1$, τ is the number of steps in local model training, η is learning rate, T is the total communication round in FL, K is the total client number, F_{inf}, B, L, σ are the constants in assumptions.

C.3. Proof of Lemma C.1

Suppose $\{A_t\}_{t=1}^T$ is a sequence of random matrices and follows $\mathbb{E}[A_t|A_{t-1}, A_{t-2}, ..., A_1] = \mathbf{0}$, then

$$\mathbb{E}\left[\left\|\sum_{t=1}^{T}A_t\right\|_F^2\right] = \sum_{t=1}^{T}\mathbb{E}\left[\|A_t\|_F^2\right]$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{t=1}^{T}A_t\right\|_F^2\right] &= \sum_{t=1}^{T}\mathbb{E}\left[\|A_t\|_F^2\right] + \sum_{i=1}^{T}\sum_{j=1,j\neq i}^{T}\mathbb{E}\left[Tr\left\{A_i^\top A_j^\top\right\}\right] \qquad (50)\\
&= \sum_{t=1}^{T}\mathbb{E}\left[\|A_t\|_F^2\right] + \sum_{i=1}^{T}\sum_{j=1,j\neq i}^{T}Tr\left\{\mathbb{E}\left[A_i^\top A_j^\top\right]\right\} \qquad (51)\\
&= \sum_{t=1}^{T}\mathbb{E}\left[\|A_t\|_F^2\right], \qquad (52)
\end{aligned}$$

where (52) comes from assuming $i < j$ and using the law of total expectation $\mathbb{E}\left[A_i^\top A_j\right] = \mathbb{E}\left[A_i^\top\mathbb{E}[A_j|A_i, ..., A_1]\right] = \mathbf{0}$.

C.4. Further Analysis with Proper Learning Rate

As a conventional setting in theoretical literature, we can set the learning rate $\eta = \frac{1}{\sqrt{\tau T}}$ (Wang et al., 2020b). Here, note that the learning rate η is strongly correlated with the number of communication round T. Then, there are two typical cases:

T is finite and relatively small. According to $\eta = \frac{1}{\sqrt{\tau T}}$, the learning rate η will be relatively large. Then, $A = 2\tau(\tau-1)\eta^2 L^2$ will be relatively large and $(1-A)$ will be relatively small. As a result, T_2 will be relatively small and T_5 will be relatively large, making T_5 a dominant term in the optimization bound. Thus, in this case, tuning p_k to make T_5 smaller is a better solution such that the a and b in Equation (2) will be non-zero and the overall upper bound could be tighter.

T is quite large or infinite. the learning rate $\eta = \frac{1}{\sqrt{\tau T}}$ will be relatively small. Then, $A = 2\tau(\tau-1)\eta^2 L^2$ will be relatively small and $(1-A)$ will be relatively large. As a result, T_2 will be relatively large and T_5 will be relatively small, making T_2 a dominant term in the optimization bound. Thus, in this case, tuning p_k to make T_2 smaller is a better solution. To achieve this, W_D should be reduced to zero and thus a and b in Equation (2) will zero to make the upper bound tight. Here, we can have the following corollary:

Corollary C.2. *By substituting $\eta = \frac{1}{\sqrt{\tau T}}$ into Theorem 4.5, we will have the following bound:*

$$\min_t \mathbb{E}||\nabla F(\mathbf{w}^{(t,0)})||^2 \leq \mathcal{O}(\frac{1}{\sqrt{\tau T}}) + \mathcal{O}(\frac{1}{T}) + \mathcal{O}(\frac{\tau}{T}). \tag{53}$$

Corollary C.2 indicates that as $T \to \infty$, the optimization upper bound approaches 0 (reaches a stationary point) and matches with previous convergence results (Wang et al., 2020b).

As the ultimate goal of this paper is to achieve more pleasant performance in practice, we are more interested in the previous part that the number of communication round T is finite (T should not be too large due to issues such as communication burden). For this reason, we explore better aggregation weights to achieve a tighter upper bound in this paper. However, our algorithm can actually converge (reach a stationary point) based on our Theorem 4.5 and Corollary C.2 if the number of communication round goes to infinite.

C.5. Reformulated Upper Bound Minimization

Generally, a tighter bound corresponds to a better optimization result. Thus, we explore the effects of p_k on the upper bound in (49). In Theorem 4.5, there there are four parts related to p_k. First, we see that larger difference between p_k and n_k contributes to larger W_D and thus smaller denominator in T_0 and larger value in T_2, which tends to loose the bound. As for T_5, by setting p_k negatively correlated to d_k, when clients have different level of discrepancy, i.e. different d_k, T_5 will be further reduced, which tends to tighten the bound. Therefore, there could be an optimal set of $\{p_k | k \in [K]\}$ that contributes to the tightest bound, where an optimal p_k should be correlated to both n_k and d_k.

To minimize this upper bound, directly solving the minimization results in a complicated expression, which involves too many unknown hyper-parameters in practice. To simplify the expression, we convert the original objective from minimizing $(T_1+T_2+T_3+T_4+T_5)/T_0$ to minimizing $T_1+T_2+T_3+T_4+T_5-\lambda T_0$, where λ is a hyper-parameter. The converted objective still promotes maximization of T_0 and minimization of $T_1+T_2+T_3+T_4+T_5$, and still contributes to tighten the bound $(T_1+T_2+T_3+T_4+T_5)/T_0$. Then, our discrepancy-aware aggregation weight is obtained through solving the following optimization problem:

$$\begin{aligned}
\min_{\{p_k\}} & \frac{2(1-A)[F(\mathbf{w}^{(0,0)}) - F_{inf}]}{\tau\eta T} + \frac{(1-A)W_D B}{K}\sum_{m=1}^{K} d_m + 2(1-A)L\eta\sigma^2 \sum_{m=1}^{K} p_m^2 \\
& + 2(\tau-1)\sigma^2 L^2\eta^2 + 2AB\sum_{m=1}^{K} p_m d_m - \lambda\left(1 - 3A - W_D(1-A)\right), \\
\text{s.t.} & \sum_m p_m = 1,\ p_m \geq 0,
\end{aligned}$$

where $W_D = 2K\left[\sum_{m=1}^{K}(n_m - p_m)^2\right]$.

To solve this constrained optimization, one condition of the optimal solution is that the derivative of the following function equals to zero:

$$\begin{aligned} Q(p_k) =& \frac{2(1-A)[F(\mathbf{w}^{(0,0)}) - F_{inf}]}{\tau\eta T} + \frac{(1-A)W_D B}{K}\sum_{m=1}^{K} d_m + 2(1-A)L\eta\sigma^2\sum_{m=1}^{K} p_m^2 \\ &+ 2(\tau-1)\sigma^2 L^2\eta^2 + 2AB\sum_{m=1}^{K} p_m d_m - \lambda\left(1 - 3A - W_D(1-A)\right) + \mu(\sum_{m=1}^{K} p_m - 1) - \sum_{m=1}^{K}\nu_m p_m \end{aligned}$$

Then, we have the following equation:

$$4(1-A)B\sum_{m=1}^{K} d_m(p_k - n_k) + 4(1-A)L\eta\sigma^2 p_k + 2ABd_k + 4K\lambda(1-A)(p_k - n_k) + \mu - \nu_m = 0,$$

from which we can rearrange and obtain the expression of p_k:

$$p_k = \frac{\left[4B(1-A)\sum_{m=1}^{K} d_m + 4K\lambda(1-A)\right] n_k - 2ABd_k - \mu + \nu_k}{4B(1-A)\sum_{m=1}^{K} d_m + 4(1-A)L\eta\sigma^2 + 4K\lambda(1-A)}. \tag{54}$$

Finally, we can derive the following concise expression of an optimized aggregation weight p_k:

$$p_k \propto n_k - a \cdot d_k + b, \tag{55}$$

where a, b are two constants.

This theoretically show that simply dataset-size-based aggregation could be not optimal since the above analysis suggests that for a tighter upper bound, the aggregation weight p_k should be correlated with both dataset size n_k and local discrepancy level d_k. This expression of aggregation weight can mitigate the limitation of standard dataset size weighted aggregation by assigning larger aggregation weight to client with larger dataset size and smaller discrepancy level.