Federated Learning with Label Distribution Skew via Logits Calibration

Jie Zhang [*1] **Zhiqi Li** [1] **Bo Li** [2] **Jianghe Xu** [2] **Shuang Wu** [2] **Shouhong Ding** [2] **Chao Wu** [1]

Abstract

Traditional federated optimization methods perform poorly with heterogeneous data (i.e. , accuracy reduction), especially for highly skewed data. In this paper, we investigate the label distribution skew in FL, where the distribution of labels varies across clients. First, we investigate the label distribution skew from a statistical view. We demonstrate both theoretically and empirically that previous methods based on softmax cross-entropy are not suitable, which can result in local models heavily overfitting to minority classes and missing classes. Additionally, we theoretically introduce a deviation bound to measure the deviation of the gradient after local update. At last, we propose FedLC (**Fed**erated learning via **L**ogits **C**alibration), which calibrates the logits before softmax cross-entropy according to the probability of occurrence of each class. FedLC applies a fine-grained calibrated cross-entropy loss into local update by adding a pairwise label margin. Extensive experiments on federated datasets and real-world datasets demonstrate that FedLC leads to a more accurate global model and much improved performance. Furthermore, integrating other FL methods into our approach can further enhance the performance of the global model.

1. Introduction

Recently, machine learning techniques enriched by massive data have been used as a fundamental technology with applications in both established and emerging fields (Chen et al., 2022; Li et al., 2019a;b;c;d;e; 2021a; Tang & Li, 2020; Tang et al., 2021; 2022; Zhang et al., 2022; Zhong et al., 2021; 2022). A large volume of data is generated on various edge devices, raising concerns about privacy and security. Federated learning (FL) (McMahan et al., 2017) emerges as a new distributed learning paradigm, which is concerned with privacy data that distributed in a non-IID (not Independent and Identically Distributed) manner.

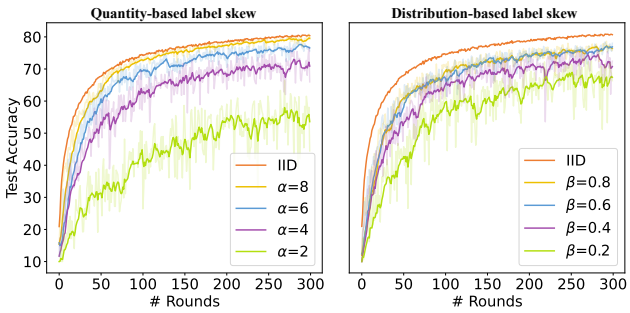

Figure 1. Test accuracy of FedAvg under various label skew settings on CIFAR10. The lower the α and β, the more skewed the distribution (See details in Section 5.1). In comparison with IID settings, the accuracy is significantly decreased by 26.07 % and 13.97 % for $\alpha = 2$ and $\beta = 0.2$, respectively.

As mentioned in earlier studies (Kairouz & McMahan, 2021; Li et al., 2018; Zhao et al., 2018), heterogeneous data can degrade the effectiveness of FL. Recent studies have proposed many methods to solve the issue of accuracy degradation in non-IID settings, such as FedProx (Li et al., 2018), Scaffold (Karimireddy et al., 2020), FedNova (Wang et al., 2020b) and FedOpt (Reddi et al., 2021). However, previous studies have very rigid data partitioning strategies among clients, which are hardly representative and thorough. To better explore the effect of non-IID data, (Li et al., 2021b) develops a benchmark with more comprehensive non-IID settings (e.g. , label distribution skew, feature distribution skew). As demonstrated in (Li et al., 2021b), none of these existing state-of-the-art FL algorithms outperforms others in all non-IID settings. It inspires researchers to develop specialized algorithms for a specific non-IID setting to further improve the performance of the global model. For example, FedBN (Li et al., 2021d) aims to address the feature distribution skew in FL.

In this paper, we primarily investigate the **label distribution skew**[1] in FL. Label distribution skew is one of the most

Work is completed during Jie Zhang's internship at Tencent Youtu Lab. [1]Zhejiang University, China [2]Youtu Lab, Tencent, China. Correspondence to: Bo Li <njumagiclibo@gmail.com>, Chao Wu <chao.wu@zju.edu.cn>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]To simulate label distribution skew, we conduct comprehensive experiments with different degrees of non-iidness.

challenging non-IID settings, where the distribution of labels varies across clients (Kairouz & McMahan, 2021; Li et al., 2021b). In fact, label distribution skew always exists in real-world applications (Al-Shedivat et al., 2021; Wang et al., 2021a). For example, pandas are only found in China and zoos, and a person's face may only appear in a few places worldwide. Following the settings in previous works (Kairouz & McMahan, 2021; Li et al., 2021b), we simulate two different label skew scenarios that are commonly used in practice: quantity-based label skew and distribution-based label skew (see detailed description in Section 5.1). As shown in Figure 1, in comparison with IID setting, the test accuracy is significantly decreased by 26.07 % and 13.97 % for highly skewed data [2]. We argue that this is reasonable. As demonstrated in previous studies (Wang et al., 2020b), heterogeneous data can result in inconsistent objective functions among clients, which leads the global model to converge to a stationary point that is far from global optima (Wang et al., 2020b). Furthermore, skewed data on the local client results in a biased model overfitting to minority classes and missing classes, which aggravates the objective inconsistency between clients (see the discussion in Section 3). Therefore, aggregating these severely biased models can result in the global model being further away from the optimal solution.

Previous studies have attempted to solve the inter-client objective inconsistency by regularizing the local objectives (Acar et al., 2021; Li et al., 2018; 2020c; Zhang et al., 2020), while ignoring the impact of local skewed data. Instead, our intuitive solution is to address the negative effects caused by intra-client label skew, which aims to reduce the bias in local update and in turn benefits the global model. This is because the performance of the global model is highly dependent on local models. Thus, resolving the intra-client label skew will produce higher quality of local models, and then a greater performance for the global model.

Our contributions are summarized as follows: 1) we first investigate the label distribution skew from a statistical perspective, and demonstrate that previous methods based on softmax cross-entropy are not suitable, which can result in biased local models. 2) Then we theoretically introduce a deviation bound to measure the deviation of the gradient after local update. 3) At last, we propose FedLC (**Fed**erated learning via **L**ogits **C**alibration), which calibrates the logit of each class before softmax cross-entropy according to the probability of occurrence. In detail, FedLC applies a fine-grained calibrated cross-entropy loss into local update by adding a pairwise label margin. By forcing the training to focus on the margins of missing classes and minority classes to reach the optimal threshold, our method encourages these underrepresented classes to have larger margins. 4) We show both theoretically and empirically that FedLC leads to a more accurate global model and much improved performance. Furthermore, integrating other methods that address inter-client objective inconsistency with our approach can further improve the performance of the server model.

[2]The more skewed the data distribution, the more difficult it is to improve the performance of the global model.

2. Related Works

Federated Learning with Non-IID Data In FL, the non-IID property across heterogeneous clients makes the local update diverge a lot, posing a fundamental challenge to aggregation. The performance of federated learning suffers from the heterogeneous data located over multiple clients (Acar et al., 2021; Li et al., 2020c). (Zhao et al., 2018) demonstrates that the accuracy of federated learning reduces significantly when models are trained with highly skewed data, which is explained by diverging weights. In a similar way to FedAvg, FedProx (Li et al., 2018) utilizes partial information aggregation and proximal term to deal with heterogeneity. FedNova (Wang et al., 2020b) puts insight on the number of epochs in local updates and proposes a normalized averaging scheme to eliminate objective inconsistency. FedOpt (Reddi et al., 2021) proposes to uses adaptive server optimization in FL and Scaffold (Karimireddy et al., 2020) uses control variates (variance reduction) to correct for the client-drift in its local updates. However, these previous works treat the non-IID problem as a general problem. How to design an effective FL algorithm to mitigate the significant accuracy reduction for label distribution skew still remains largely unexplored.

Learn from Imbalanced data In recent years, many studies have been focusing on analyzing imbalanced data (He & Garcia, 2009; Liu et al., 2019; Zhang et al., 2021b). Real-world data usually exhibits a imbalanced distribution, and the effectiveness of machine learning is severely affected by highly skewed data (Cao et al., 2019; Jamal et al., 2020). Re-sampling (Chawla et al., 2002) and re-weighting (Cui et al., 2019; Jamal et al., 2020; Kim & Kim, 2020) are traditional methods for addressing imbalanced data. Recent works use re-weighting methods to enable networks to pay more attention to minority categories by assigning a variable weight to each class. Besides, over-sampling minority classes and under-sampling frequent classes are two re-sampling methods that have been extensively discussed in previous studies. New perspectives like decoupled training (Kang et al., 2019) and deferred re-balancing (Cao et al., 2019) schedule are also proved to be effective. The majority of previous works on imbalanced data focus on long-tailed distributions. However, in federated learning settings, data can be imbalanced in many ways (Kairouz & McMahan, 2021; Li et al., 2021b), such as quantity-based label imbalance and distribution-based label imbalance, as discussed

in this paper. Besides, the label distribution skew includes long-tail scenarios, but long-tailed methods cannot handle the issue of missing classes, which is extremely common in FL.

Federated Learning with Label Distribution Skew To alleviate the negative effect of label distribution skew, (Wang et al., 2021b) proposes a monitoring scheme to detect class imbalance in FL. Nevertheless, this method relies heavily on auxiliary data, which is not practical in real-world FL and poses potential privacy issues. Besides, this method also needs an additional monitor in the central server, which requires more computation. Note that FedRS (Li & Zhan, 2021) works in a similar manner, which also attempts to alleviate the negative effect caused by local training on label skewed data. However, it only resolves the issue of missing classes during local updates. Generally, in real-world applications, the local data contains both majority classes and minority classes, as well as missing classes. By contrast with previous methods, our approach systematically analyzes the problem of federated learning with label distribution skew from a statistical perspective. Our approach considers majority classes, minority classes, and missing classes at the same time, which is more practical.

3. FL with Label Distribution Skew

Definition 1 (**Label Distribution Skew**). *Suppose that client i can draw an example $(x, y) \sim P_i(x, y)$ from the local data, and the data distribution $P_i(x, y)$ can be rewritten as $P_i(x \mid y)P_i(y)$. For label distribution skew, the marginal distributions $P_i(y)$ varies across clients, while $P_i(y \mid x) = P_j(y \mid x)$ for all clients i and j.*

Definition 2 (**Majority, Minority and Missing Classes**). *Generally, for a label skewed dataset, the label set K is split into majority class $j \in \mathcal{J}$, minority class $r \in \mathcal{R}$ and missing class $s \in \mathcal{S}$ respectively. We have $\mathcal{J} \cap \mathcal{R} = \emptyset, \mathcal{R} \cap \mathcal{S} = \emptyset, \mathcal{S} \cap \mathcal{J} = \emptyset$ and $\mathcal{J} \cup \mathcal{R} \cup \mathcal{S} = K$, where $\emptyset$ is an empty set. The number of training samples in each class satisfies $n_j \gg n_r > 0, n_s = 0$.*

In federated learning, a total of m clients aim to jointly minimize the following optimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x}) := \sum_{i=1}^{m} p_i F_i(\boldsymbol{x}), F_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{\xi \in \mathcal{P}_i} f_i(\boldsymbol{x}; \xi),$$

where $F_i(\boldsymbol{x})$ is the objective function of i-th client, and p_i denotes the relative sample size and $\sum_{i=1}^{m} p_i = 1$. The local data distribution P_i varies among clients, posing the problem of data heterogeneity. In FL, the clients selected in each communication round perform multiple local updates, and then these models are aggregated to update a global model. However, for label skew settings, after local updates, these models will learn highly biased decision boundaries, resulting in poor performance when aggregated.

Affected by the heterogeneous data, the local objectives at clients are generally not identical and may not share same minimizers (Wang et al., 2021a). Thus, when updating local models from the same global model, clients will drift towards the minima of local objectives. This phenomenon is often referred to as client drift (Charles & Konečný, 2021; Karimireddy et al., 2020). As demonstrated in previous studies (Wang et al., 2020b; Zhao et al., 2018), standard averaging of these models with client drift leads to convergence to a stationary point which is not that of the original objective function, but of an inconsistent objective function. That is, federated learning cannot achieve optimal weights when training data distribution is skewed. Generally, the more skewed the local data, the harder it is to aggregate a well-performed global model.

Besides, we empirically verify the impact of label skew in FL on five clients after 400 communication rounds and test the accuracy for each class before and after local update. All local models on the test set have same test accuracy before local update (currently, these local models are equivalent to the global model). As illustrated in Figure 2, after training on local data, the test accuracy of majority classes is even higher than that of the global model, and the test accuracy of minority classes is much lower, and the accuracy of missing classes is almost zero. It indicates that label skew can lead to a biased model, which severely overfits to the minority classes and missing classes. As a result, label skew exacerbates the negative effects of objective inconsistency and leads to a more inaccurate global model.

Based on above discussions, we show an illustration in Figure 3 of why FL with label distribution skew performs poorly. In the following section, we provide a deeper understanding of label skewed data from a statistic view.

Deviation of Standard FL algorithms To better analyse the negative effect of label distribution skew in FL, we theoretically show the problems of local update when faced with label distribution skew. We focus on multi-classification task for each client with label set K. Let $f(x) = \{w_y^T h\}_{y \in K}$ be the score function for a given input x, w is the classification weights in the last layer and h is the extracted feature as the input of the last layer. Let $p = \sigma(f(x))$ denote the probability vector after softmax function $\sigma(\cdot)$. To evaluate the degree of deviation of the update $\{\Delta w_y\}_{y \in K}$ from the expected direction during the local training, we introduce the deviation bound as follows:

Definition 3 (**Deviation Bound**). *For majority class $j \in \mathcal{J}$ and minority class $r \in \mathcal{R}$, let O_j and O_r denote the set of training samples belong to class j and class r, respectively.*

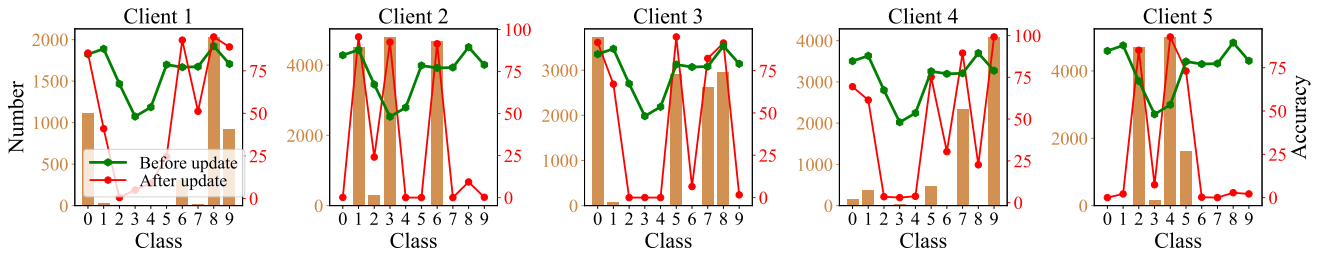

Figure 2. For skewed CIFAR10 dataset, the accuracy decreases heavily on minority classes, achieving an overall accuracy of zero for missing classes. The histogram displays the number of samples for each class, while the red line represents the accuracy of each class.

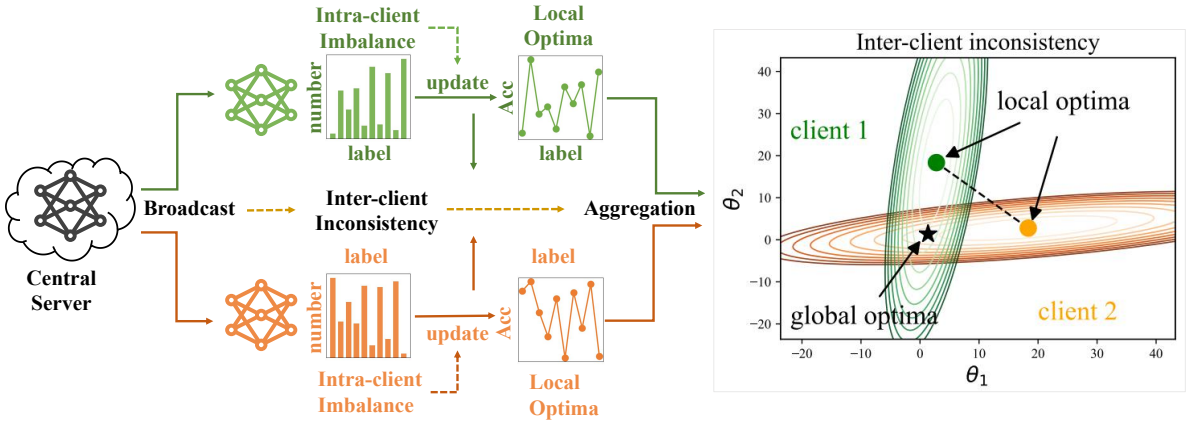

Figure 3. A toy example with 2 clients. The intra-client skew results in a biased model (converge to local optima), which means that after local update the accuracy drops heavily on minority classes. Moreover, averaging such biased local models can lead to a poor model that stray from the corresponding global optima (Al-Shedivat et al., 2021).

The deviation bound is :

$$D_{jr} = \frac{(1-\overline{p_r^{(r)}})||\overline{h^{(r)}}||_2^2}{\overline{p_r^{(j)}}\overline{h^{(r)}} \cdot \overline{h^{(j)}}} = \sum_{y\in K, y\neq r} \frac{\overline{p_y^{(r)}}||\overline{h^{(r)}}||_2^2}{\overline{p_r^{(j)}}\overline{h^{(r)}} \cdot \overline{h^{(j)}}},$$

where $\overline{p_r^{(j)}} = \frac{1}{n_j}\sum_{i\in O_j} p_r(x_i)$, $\overline{h^{(j)}} = \frac{1}{n_j}\sum_{i\in O_j} h(x_i)$ *are averaged over all samples in majority class* j, $\overline{p_r^{(r)}} = \frac{1}{n_r}\sum_{i\in O_r} p_r(x_i)$, $\overline{p_y^{(r)}} = \frac{1}{n_r}\sum_{i\in O_r} p_y(x_i)$ *and* $\overline{h^{(r)}} = \frac{1}{n_r}\sum_{i\in O_r} h(x_i)$ *are averaged over all samples in minority class* r. *The proof can be found in* ***Appendix C****.*

Theorem 1. *For majority class* $j \in \mathcal{J}$ *and minority class* $r \in \mathcal{R}$. *When* $n_j/n_r \gg D_{jr} > 0$, *the update* $\{\Delta w_y\}_{y\in K}$ *is much more likely to deviate from expected direction, where* $\Delta w_r \overline{h^{(r)}} < 0$ *and* $\Delta w_k \overline{h^{(r)}} > 0$. *The proof can be found in* ***Appendix C****.*

Here, the relation between n_j/n_r and D_{jr} describes how the scores for majority classes overwhelm those for minority classes during local training, and it generalise previous work (Li & Zhan, 2021), which only considers missing classes. Missing class can be viewed as a special case of minority class, where $n_r \approx 0$, and $n_j/n_r \gg D_{jr}$ almost always holds.

Based on above discussions, the deviation bound gives us a quantitative perspective to reveal the drawbacks of previous FL methods based on standard FL (softmax cross-entropy) during the local update, where the update $\{\Delta w_y\}_{y\in K}$ deviates from expected direction. More detailed analysis about the deviation in local update can be seen in **Appendix C**.

4. Federated Learning via Logits Calibration

As mentioned before, the local update is biased with label distribution skew. In this section, we demonstrate that standard softmax cross-entropy is not suitable for local update with highly skewed data. To overcome this, we propose a fine-grained calibrated cross-entropy loss to reduce the bias in local update.

Learning Objective Suppose the data distribution at i-th client is $P_i(x, y) = P_i(x \mid y)P_i(y)$. Given a data x, the predicted label is $\hat{y} = \arg\max_y f_y(x)$. For balanced label distribution, the goal of standard machine learning

is to minimize the misclassification error from a statistical perspective:

$$P_{x,y}(y \neq \hat{y}), \text{and } P(y \mid x) \propto P(x \mid y)P(y). \quad (1)$$

Since softmax cross-entropy is usually chosen as the surrogate loss function, the probability $P_y(x) \propto e^{f_y(x)}$ is regarded as the estimates of $P(y \mid x)$.

However, in this paper, we focus on label distribution skew in FL, which means $P(y)$ is skewed. That is, minority classes have a much lower probability of occurrence compared with majority classes, which means that minimizing the misclassification error $P(x \mid y)P(y)$ is no longer suitable (Menon et al., 2021). To handle this, we average each of the per-class error rate (Menon et al., 2021), and attempt to minimize the test error as follows:

$$\text{Calibrated error} = \min \frac{1}{k} \sum_{y \in K} \mathcal{P}_{x|y}(y \neq \hat{y}). \quad (2)$$

In this manner, the result is the estimate of $P(x \mid y)$, thus varying $P(y)$ arbitrarily will not affect the optimal results. In other words, when label distribution is skewed, we aim to minimize the calibrated error P^{Cal} as follows:

$$\begin{aligned} \arg\max_{y \in K} P^{Cal}(y \mid x) &= \arg\max_{y \in K} P(x \mid y) \\ &= \arg\max_{y \in K} \{P(y \mid x)/P(y)\}. \end{aligned} \quad (3)$$

Since softmax cross-entropy loss indicates that $P(y \mid x) \propto e^{f_y(x)}$, then Equation 3 can be rewritten as:

$$\arg\max_{y \in K} P^{Cal}(y \mid x) = \arg\max_{y \in K} \{f_y(x) - \log \gamma_y\}, \quad (4)$$

where γ_y is the estimate of the class prior $P(y)$. This formulation inspires us to calibrate the logits [3] before softmax cross-entropy according to the probability of occurrence of each class. In other words, we should encourage the logits of minority classes to minus a relative larger value. Inspired by Equation 4, we calibrate the logits for each class before softmax cross-entropy, then the modified cross-entropy loss can be formulated as:

$$\mathcal{L}_{\text{Cal}}(y; f(x)) = -\log \frac{1}{\sum_{i \neq y} e^{-f_y(x)+f_i(x)+\Delta_{(y,i)}}}, \quad (5)$$

where $\Delta_{(y,i)} = \log(\frac{\gamma_i}{\gamma_y})$. For more insight into $\Delta_{(y,i)}$, it can be viewed as a pairwise label margin, which represents the desired gap between scores for y and i. With this optimization objective, we aim to find the optimal pairwise label margin $\Delta_{(y,i)}$ and train the local model with our calibrated loss as usual even with label distribution skew.

[3] Logits denotes the output of the last classification layer and the input to softmax.

Fine-grained Calibrated Cross-Entropy Compared to the standard softmax cross-entropy, Equation 5 applies a pairwise label margin $\Delta_{(y,i)}$ to each logit. Adjusting the value of $\Delta_{(y,i)}$ for each class is the key factor in our modified loss function. For label skewed data, motivated by the interesting idea in (Cao et al., 2019), we provide the following optimal pairwise label margins to minimize the test error:

Theorem 2. *For any given input (x, y), the margin of label y is $d_y = f_y(x) - \max_{i \neq y} f_i(x)$, which denotes the minimum distance of the data in class y to the decision boundary. We show that the test error for label skewed data is bounded by $\frac{1}{d_y\sqrt{n_y}} + \frac{1}{d_i\sqrt{n_i}}$. The optimal pairwise label margin is:*

$$\Delta_{(y,i)} = \tau \cdot (n_y^{-1/4} - n_i^{-1/4}), \quad (6)$$

*where n_y and n_i are the sample size of class y and i, respectively. And τ is a hyper-parameter. The proof can be found in **Appendix D**.*

Based on above analysis, we propose a fine-grained loss function for local training that calibrates the logits based on the enforced pairwise label margins to reduce the bias in local update:

$$\mathcal{L}_{\text{Cal}}(y; f(x)) = -\log \frac{e^{f_y(x)-\tau \cdot n_y^{-1/4}}}{\sum_{i \neq y} e^{f_i(x)-\tau \cdot n_i^{-1/4}}}. \quad (7)$$

This loss function simultaneously minimizes the classification errors and forces the learning to focus on margins of minority classes to reach the optimal results. During local training, $\mathcal{L}_{\text{Cal}}$ should give an optimal trade-off between the margins of classes.

Deviation Bound after Calibration In this paragraph, we demonstrate that FedLC can mitigate the deviation of local gradient as follows:

Theorem 3. *For majority class $j \in \mathcal{J}$ and minority class $r \in \mathcal{R}$, after adding the pairwise label margin $\Delta_{(y,i)} = \tau(n_y^{-1/4} - n_i^{-1/4})$ for all pairs of classes, the deviation bound becomes:*

$$D_{jr} = \sum_{y \in K, y \neq r} \Delta_{(r,y)} \frac{\overline{\widetilde{p}_y^{(r)}} ||\overline{h^{(r)}}||_2^2}{\overline{\widetilde{p}_r^{(j)}}\, \overline{h^{(r)}} \cdot \overline{h^{(j)}}}$$

*where $\overline{\widetilde{p}_r^{(j)}} = \frac{1}{n_j} \sum_{i \in O_j} \widetilde{p}_r(x_i)$ is averaged over all samples in majority class j, $\overline{\widetilde{p}_y^{(r)}} = \frac{1}{n_r} \sum_{i \in O_r} \widetilde{p}_y(x_i)$ is averaged over all samples in majority class r. And also we have $\widetilde{p}_r(x) = \frac{e^{f_r(x)}}{\sum_{i=1}^K e^{f_i(x)-\tau \cdot n_i^{-1/4}}}$, $\widetilde{p}_y(x) = \frac{e^{f_y(x)}}{\sum_{i=1}^K e^{f_i(x)-\tau \cdot n_i^{-1/4}}}$. The proof can be found in **Appendix C**.*

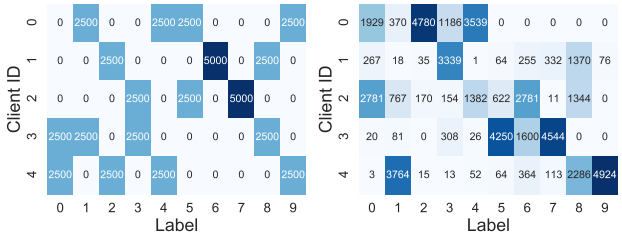

Figure 4. Visualizations of skewed CIFAR10 dataset on 5 clients. **Left:** quantity-based label skew ($\alpha = 4$); **Right:** distribution-based label skew ($\beta = 0.5$). The value in each rectangle is the number of data samples of a label belonging to a certain client.

Obviously, the modified loss adds pairwise label margin directly into the deviation bound to enlarge D_{jr} for minority class r and make $n_j/n_r \gg D_{jr}$ more difficult, which mitigate the deviation of $\{\Delta w_y\}_{y=1}^K$ during local training. Thus in this way, we can reduce the bias of local model updates for each client, which in turn benefits the global model. The complete pseudo code of FedLC can be found in **Appendix 1**.

5. Experiments

5.1. Type of Label Distribution Skew

To simulate label distribution skew, we follow the settings in (Li et al., 2021b) and introduce two frequently used label skew settings: quantity-based label skew and distribution-based label skew. An example of different types of label distribution skew is shown in Figure 4.

Quantity-based Label skew It is first introduced in FedAvg (McMahan et al., 2017), and has been frequently used in many recent studies (Li et al., 2020a;b; Li & Zhan, 2021; Shen et al., 2020). Suppose that there are n training samples distributed among m clients. Firstly, we sort the data by labels and divide it into $m \cdot \alpha$ sets, each set contains $\frac{n}{m \cdot \alpha}$ samples. Then we assign α sets to each client. We refer to this approach as $Q(\alpha)$, where α controls the degree of label skew. Note that there is no overlap between the samples of different clients. Each client's training data contains only a few labels, which means there are missing classes.

Distribution-based Label skew This partitioning strategy was first introduced in (Yurochkin et al., 2019), such a setting is also used in many other studies (Li et al., 2021c; Lin et al., 2020; Wang et al., 2020a; Zhang et al., 2021a). Each client is allocated a proportion of the samples of each label according to Dirichlet distribution. In detail, we sample the data by simulating $\mathbf{p}_k \sim \mathrm{Dir}(\beta)$ and allocate a portion of $\mathbf{p}_{k,j}$ of the samples in class k to client j. For ease of presentation, we use $D(\beta)$ to denote such a partitioning strategy. Here β controls the degree of skewness. Note that when using this partitioning strategy, the training data of each client may have majority classes, minority classes, or even some missing classes, which is more practical in real-world applications.

5.2. Experimental Setups

Datasets In this study, we conduct a number of experiments on popular image classification benchmark datasets: SVHN (Netzer et al., 2011), CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009), as well as federated datasets (Synthetic dataset and FEMNIST) proposed in LEAF (Caldas et al., 2019). According to (Li et al., 2021e), we generate the Imagenet-subset with size 64*64*3 , which consists of 12 labels for fast training. To better simulate label distribution skew in Synthetic, a mixed manner consists of quantity-based label skew and distribution-based label skew are used in our experiments. We denote it as Synthetic(λ, μ), where local data size follows a power law. Note that λ specifies how much local models differ from one another, and μ indicates how skewed the local data is at each client. We use a simple logistic model ($y = \arg\max(\mathrm{softmax}(Wx + b))$) to generate data samples. For FEMNIST, we use the default setting in LEAF (Caldas et al., 2019).

Baselines and Implementation Our method aims at improving the performance of federated learning with label distribution skew. As a result, we choose typical approaches to non-IID issues as our baselines, such as FedProx (Li et al., 2018),Scaffold (Karimireddy et al., 2020), FedNova (Wang et al., 2020b) and FedOpt (Reddi et al., 2021) as our baselines. For fair comparison, we also compare our method with FedRS (Li & Zhan, 2021), which focuses on the issue of label skew in federated learning. We implement the typical federated setting (McMahan et al., 2017) in Pytorch, and all experiments are conducted with 8 Tesla V100 GPUs. At default, there are 20 clients totally. We use two CNN architectures as our initial models, and the detailed model architecture can be seen in **Appendix A**. The size of local mini-batch is 128. For local training, each client updates the weights via SGD optimizer with learning rate $\eta = 0.01$ without weight decay. We run each experiment with 5 random seeds and report the average and standard deviation.

5.3. Experiments on Federated Datasets

In this section, we evaluate these algorithms on Synthetic and FEMNIST dataset. We divide the training data into 100 clients over 300 communication rounds. To manipulate heterogeneity more precisely, we synthesize unbalanced datasets with 3 different settings. Specifically, we follow the settings in (Li et al., 2018) and generate Synthetic(0,0),

Table 1. Performance overview for different degrees of distribution-based label skew.

Dataset	SVHN				CIFAR10				CIFAR100			
Skewness	$\beta = 0.05$	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.05$	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.05$	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$
FedAvg	$69.51_{\pm 1.45}$	$79.86_{\pm 1.46}$	$85.14_{\pm 0.83}$	$86.02_{\pm 1.15}$	$37.63_{\pm 1.36}$	$48.07_{\pm 1.38}$	$55.95_{\pm 0.83}$	$60.18_{\pm 1.78}$	$21.37_{\pm 0.87}$	$25.06_{\pm 1.04}$	$28.44_{\pm 1.51}$	$29.29_{\pm 1.32}$
FedProx	$71.42_{\pm 1.24}$	$81.39_{\pm 1.35}$	$86.30_{\pm 0.95}$	$87.53_{\pm 1.56}$	$39.03_{\pm 1.27}$	$49.57_{\pm 0.90}$	$57.88_{\pm 0.93}$	$62.13_{\pm 1.17}$	$22.92_{\pm 1.71}$	$26.44_{\pm 0.86}$	$30.16_{\pm 1.18}$	$31.20_{\pm 1.23}$
Scaffold	$71.23_{\pm 1.63}$	$81.80_{\pm 1.75}$	$86.32_{\pm 1.19}$	$87.13_{\pm 1.39}$	$38.84_{\pm 0.93}$	$49.12_{\pm 1.21}$	$57.39_{\pm 1.16}$	$61.54_{\pm 1.28}$	$22.61_{\pm 1.37}$	$26.30_{\pm 1.32}$	$29.96_{\pm 1.17}$	$31.26_{\pm 1.75}$
FedNova	$72.50_{\pm 1.21}$	$82.41_{\pm 1.40}$	$87.11_{\pm 1.38}$	$86.65_{\pm 1.25}$	$39.81_{\pm 1.18}$	$50.56_{\pm 1.42}$	$58.85_{\pm 0.93}$	$62.77_{\pm 0.86}$	$24.03_{\pm 0.91}$	$27.65_{\pm 0.99}$	$30.76_{\pm 0.95}$	$31.93_{\pm 0.98}$
FedOpt	$73.46_{\pm 1.07}$	$82.71_{\pm 1.13}$	$86.85_{\pm 0.85}$	$87.41_{\pm 1.72}$	$41.08_{\pm 1.01}$	$51.89_{\pm 0.86}$	$59.39_{\pm 1.68}$	$63.38_{\pm 1.62}$	$24.51_{\pm 1.71}$	$28.98_{\pm 1.08}$	$32.42_{\pm 1.66}$	$32.94_{\pm 1.28}$
FedRS	$75.97_{\pm 1.15}$	$83.27_{\pm 1.54}$	$87.01_{\pm 0.98}$	$87.40_{\pm 1.67}$	$44.39_{\pm 1.63}$	$54.04_{\pm 1.59}$	$62.40_{\pm 1.38}$	$66.39_{\pm 1.28}$	$27.93_{\pm 1.18}$	$32.89_{\pm 1.50}$	$36.58_{\pm 0.94}$	$38.98_{\pm 1.35}$
Ours	$\mathbf{82.36}_{\pm 0.67}$	$\mathbf{84.41}_{\pm 0.87}$	$\mathbf{88.02}_{\pm 1.19}$	$\mathbf{88.48}_{\pm 1.29}$	$\mathbf{54.55}_{\pm 1.70}$	$\mathbf{65.91}_{\pm 1.68}$	$\mathbf{72.18}_{\pm 0.86}$	$\mathbf{72.99}_{\pm 1.12}$	$\mathbf{38.08}_{\pm 0.84}$	$\mathbf{41.01}_{\pm 1.08}$	$\mathbf{44.23}_{\pm 1.70}$	$\mathbf{44.96}_{\pm 1.71}$

Table 2. Performance overview for different non-IID settings on CIFAR10 and CIFAR100 (quantity-based label skew).

Dataset	CIFAR10				CIFAR100			
Degree of skewness	$\alpha = 2$	$\alpha = 4$	$\alpha = 6$	$\alpha = 8$	$\alpha = 20$	$\alpha = 40$	$\alpha = 60$	$\alpha = 80$
FedAvg	$52.23_{\pm 1.01}$	$71.29_{\pm 1.49}$	$77.96_{\pm 0.92}$	$78.49_{\pm 1.30}$	$42.21_{\pm 1.38}$	$47.61_{\pm 1.20}$	$49.54_{\pm 1.60}$	$49.81_{\pm 1.84}$
FedProx	$52.97_{\pm 1.12}$	$72.51_{\pm 1.17}$	$78.74_{\pm 1.55}$	$80.16_{\pm 0.92}$	$43.33_{\pm 1.40}$	$48.53_{\pm 1.64}$	$51.16_{\pm 1.72}$	$51.25_{\pm 1.28}$
Scaffold	$53.85_{\pm 1.75}$	$72.33_{\pm 1.04}$	$78.89_{\pm 1.38}$	$79.88_{\pm 1.65}$	$43.69_{\pm 1.07}$	$49.26_{\pm 1.52}$	$50.95_{\pm 1.24}$	$51.42_{\pm 0.85}$
FedNova	$53.27_{\pm 1.40}$	$73.87_{\pm 1.44}$	$80.18_{\pm 1.03}$	$79.70_{\pm 1.59}$	$43.14_{\pm 1.35}$	$49.39_{\pm 1.69}$	$51.02_{\pm 1.02}$	$51.51_{\pm 0.92}$
FedOpt	$54.35_{\pm 1.29}$	$72.51_{\pm 1.52}$	$79.47_{\pm 1.46}$	$80.05_{\pm 1.28}$	$43.97_{\pm 1.39}$	$49.07_{\pm 1.13}$	$50.98_{\pm 0.96}$	$51.27_{\pm 1.95}$
FedRS	$55.91_{\pm 1.44}$	$74.44_{\pm 1.15}$	$80.62_{\pm 1.13}$	$80.29_{\pm 1.02}$	$45.58_{\pm 1.90}$	$49.95_{\pm 1.76}$	$50.83_{\pm 1.16}$	$51.99_{\pm 1.51}$
Ours	$\mathbf{61.87}_{\pm 0.94}$	$\mathbf{77.22}_{\pm 1.22}$	$\mathbf{81.41}_{\pm 1.18}$	$\mathbf{82.05}_{\pm 0.89}$	$\mathbf{47.77}_{\pm 0.84}$	$\mathbf{52.26}_{\pm 1.43}$	$\mathbf{52.89}_{\pm 1.34}$	$\mathbf{53.42}_{\pm 0.85}$

Table 3. Performance overview given different skewed ImageNet-subset datasets.

Degree of skewness	$\alpha = 2$	$\alpha = 4$	$\beta = 0.1$	$\beta = 0.3$
FedAvg	53.87	63.12	43.91	56.21
FedProx	55.87	65.19	45.75	58.13
Scaffold	56.27	64.64	46.34	57.92
FedNova	56.02	65.29	46.05	58.45
FedOpt	55.91	65.17	45.89	58.08
FedRS	57.23	67.78	47.23	59.83
Ours	**62.43**	**72.33**	**53.24**	**63.81**

Table 4. Test accuracy on various federated datasets.

Dataset	Synthetic(0,0)	Synthetic(0.5,0.5)	Synthetic(1,1)	FEMNIST
FedAvg	$72.09_{\pm 1.19}$	$67.55_{\pm 1.16}$	$63.11_{\pm 0.66}$	$84.14_{\pm 0.73}$
FedProx	$72.21_{\pm 0.47}$	$67.85_{\pm 0.82}$	$63.09_{\pm 0.79}$	$85.41_{\pm 1.28}$
Scaffold	$73.51_{\pm 0.67}$	$69.45_{\pm 0.82}$	$64.95_{\pm 0.83}$	$86.24_{\pm 1.29}$
FedNova	$73.22_{\pm 0.94}$	$69.01_{\pm 0.87}$	$64.67_{\pm 0.46}$	$85.32_{\pm 1.62}$
FedOPT	$73.42_{\pm 0.76}$	$68.64_{\pm 0.67}$	$63.69_{\pm 0.89}$	$86.56_{\pm 0.48}$
FedRS	$75.79_{\pm 1.25}$	$71.35_{\pm 0.81}$	$66.39_{\pm 1.41}$	$87.95_{\pm 0.92}$
Ours	$\mathbf{80.92}_{\pm 0.31}$	$\mathbf{78.47}_{\pm 0.34}$	$\mathbf{75.46}_{\pm 0.52}$	$\mathbf{92.78}_{\pm 0.58}$

Synthetic(0.5,0.5), and Synthetic(1,1).

Observed from Table 4, we find that a large λ and μ on the Synthetic datasets can lead to poor performance on test accuracy. Especially, for Synthetic(1,1), our method achieves a prediction accuracy of 75.45%, which is higher than that of the best baseline FedRS by 9.07%. Obviously, we can find that on all datasets, our method consistently outperforms much better than any other baseline. As a point of interest, more similarities in performance can be observed between these baselines, which indicates that these methods are not appropriate for highly skewed data distribution.

5.4. Experiments on Real-World Datasets

Results on SVHN, CIFAR10 and CIFAR100 All results are tested after 400 communication rounds. We mainly report the evaluations of these algorithms with different degrees of label skew. Table 1 summarizes the results for different types of distribution-based label skew. Evidently, in all scenarios, our method significantly achieves a higher accuracy than other SOTA methods. As data heterogeneity increases (i.e. , smaller β), all competing methods struggle, whereas our method displays markedly improved accuracy on highly skewed data. For example, for CIFAR-10 dataset with $\beta = 0.05$, our method gets a test accuracy of 54.55%, which is much higher than that of FedRS by 10.16%. In addition, we also report the performance of these methods for different types of quantity-based label skew in Table 2, which can further demonstrate the superiority of our method.

Results on ImageNet-subset In addition, we compare the prediction performance of these methods on ImageNet-subset dataset. We choose ResNet-18 as the default network. We compare these FL algorithms in terms of both quantity-based label skew and distribution-based label skew with $\alpha = \{2, 4\}$ and $\beta = \{0.1, 0.3\}$. As shown in Table 3, we have to emphasize here that our method still performs better even on such a complex dataset. In the case of highly skewed data with $\alpha = 2$ and $\beta = 0.1$, our method can achieve a mean accuracy of 62.43% and 54.43%, which is

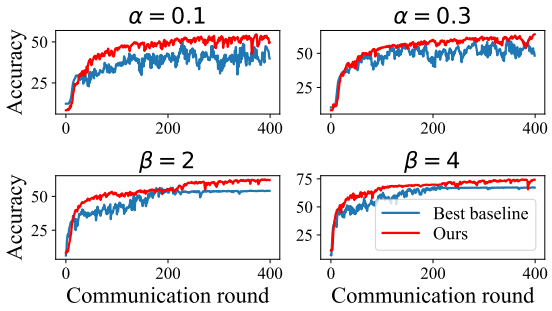

Figure 5. Test accuracy for different skewed ImageNet-subset datasets. We compare our method with the best performed baseline FedRS on 40 clients.

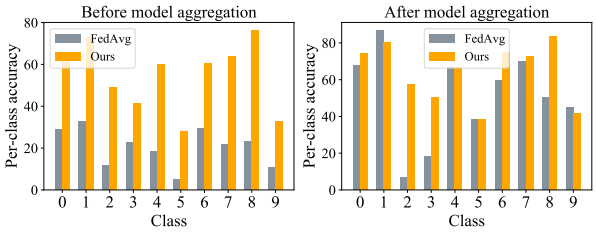

Figure 6. Average per-class accuracy before and after model aggregation. For fair comparisons, we use the same well-trained model for initialization and the same data partition on each client.

better than the best baseline FedRS by 7.2% and 6.01%, respectively. Besides, we plot the test accuracy curves in Figure 5, which means that our method is relatively stable on highly skewed data (e.g. , $\alpha = 0.1$).

5.5. Analyse of our method

FedMC can mitigate over-fitting To verify the effectiveness of our method, we compare the average per-class accuracy of our method with FedAvg before and after model aggregation. For fair comparison, we use a same well-trained federated model as the global model. Then the central server distributes the model parameter to all clients. Next, we use FedAvg and our method to train the local model with the same local data for only 1 epoch. At last, we report the average per-class accuracy of all clients. Note that there is only one difference in the whole process - the local update. According to the left sub-figure of Figure 6, our method's average per-class accuracy is much higher than that of FedAvg. As we mentioned before(see Figure 2), for FedAvg, due to the highly skewed label distribution, each client has very low accuracy on minority classes and missing classes. Therefore, after the local update, each client's model is biased, resulting in a lower average accuracy for each class. By contrast, with our method, each client has higher performance in all classes. The results show that the our new cross-entropy loss can indeed improve the performance on

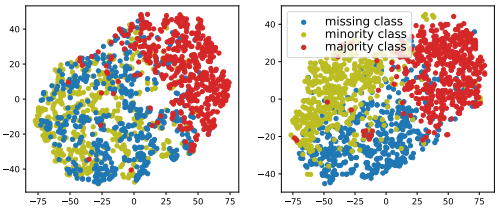

Figure 7. TSNE visualizations on majority, minority and missing classes. **Left:** For FedAvg, the samples from the minority class and missing class are mixed together and indistinguishable. **Right:** For our method, the data from minority class and missing class can be distinguished well, which indicates our method can learn more discriminative features.

Table 5. Performance overview on CIFAR10 and SVHN for different local epochs E. All experiments are conducted on 10 clients.

	Dataset	CIFAR10		SVHN	
# Local Epoch	Skewness	$\beta = 0.5$	$\alpha = 5$	$\beta = 0.5$	$\alpha = 5$
E=1	FedAvg	54.25	46.12	82.07	84.91
	FedProx	54.68	46.84	83.18	85.22
	FedNova	56.14	47.58	83.24	85.14
	Ours	**70.41**	**58.43**	**85.89**	**88.34**
E=5	FedAvg	77.02	75.81	89.01	88.59
	FedProx	77.48	75.14	90.01	89.12
	FedNova	78.64	75.88	90.17	88.79
	Ours	**80.25**	**77.76**	**91.83**	**91.25**
E=10	FedAvg	80.07	78.81	87.82	89.15
	FedProx	80.25	78.72	88.18	89.45
	FedNova	80.56	78.97	88.16	89.53
	Ours	**81.67**	**79.61**	**89.54**	**90.56**
E=20	FedAvg	80.61	79.21	87.39	89.42
	FedProx	80.17	79.13	88.01	89.14
	FedNova	80.45	79.32	87.52	89.24
	Ours	**81.99**	**80.19**	**88.75**	**89.46**

minority classes and missing classes, and further improve the overall performance of the global model. As shown in the right sub-figure of Figure 6, our method can also perform better than FedAvg after model aggregation, which means our method alleviates the over-fitting in local update.

T-SNE visualization on majority, minority and missing classes Additionally, we also show a t-SNE (Van der Maaten & Hinton, 2008) visualization in Figure 7 after local update. The number of training data is $\{0, 448, 2608\}$ for missing, minority and majority class. As illustrated in this figure, the test data from different classes are hard to be separated by FedAvg, only the features of the majority class are obvious, minority class and missing class are mixed together and difficult to distinguish. Actually, our method can learn more discriminative features, which indicates that our method certainly yields better performance.

5.6. Ablation Study

Performance for different local epochs E In this section, we add more computation per client on each round by increasing the local epochs E. We conduct many experiments

Table 6. Test accuracy with different number of clients.

# Clients	$m = 10$	$m = 30$	$m = 50$	$m = 100$
FedAvg	57.19	45.09	36.70	29.30
FedProx	59.68	47.11	39.07	31.74
Scaffold	59.59	47.14	38.20	31.09
FedNova	58.72	47.49	38.74	33.30
FedOpt	61.25	48.19	40.98	33.40
FedRS	62.62	50.05	41.98	35.95
Ours	**70.13**	**58.79**	**52.08**	**43.81**

on CIFAR10 and SVHN dataset. There are 20 clients totally. We compare these FL algorithms in terms of both quantity-based label skew and distribution-based label skew with $\alpha = 5$ and $\beta = 0.5$. As shown in Table 5, in cases where E=1, our method is superior to other baselines by a large margin. The model is not well trained when the number of local updates is too small. Our method aims to reduce such bias in local updates. For a large value of E, while the performance of these baselines is relatively similar, our approach still consistently outperforms other methods.

Performance for different number of clients m To show the effectiveness of FedLC, we train these methods with different numbers of clients m. Table 6 reports all the results across $m = \{10, 30, 50, 100\}$. As expected, our proposed FedLC achieves the best performance across all settings, which further validates that FedLC can be applied in most practical settings. As m increases, the performance of all methods decreases. We conjecture that the reason is that more clients in FL make the model harder to converge. However, our method can still achieve 43.81% accuracy when there are 100 clients.

Combination with other techniques As we focus on addressing the intra-client skew, we believe combining effective methods that target inter-client objective inconsistency with our approach can further improve the performance of the server model. In this section, we integrate additional regularization terms used in FedNova and FedProx into our method. As illustrated in Table 7, the combination of Scaffold and FedProx with our method leads to a better performance. We would like to argue that this is reasonable. Regularizing the inconsistent objectives can prevent local models from updating towards their local minima. Based on our analysis, we believe that incorporating approaches focused on intra-client label skew and inter-client objective inconsistency will result in more efficient and effective FL algorithms, which we leave as future work.

6. Conclusion

In this work, we propose a fine-grained calibrated loss to improve the performance of the global model with label distribution skew. The comprehensive experiments demonstrate the proposed method can effectively reduce the bias in local update. We hope that our study can inspire other researchers to explore more connections between intra-client label skew and inter-client objective inconsistency.

Table 7. Performance overview on the combination of Scaffold and FedProx with our method.

dataset	CIFAR10		SVHN	
Degree of skewness	$\alpha = 5$	$\beta = 0.3$	$\alpha = 5$	$\beta = 0.3$
FedAvg	61.54	60.18	84.95	84.85
Ours	69.89	70.38	88.19	88.31
Ours+Scaffold	70.45	71.47	89.34	89.45
Ours+FedProx	71.39	71.98	90.32	89.49

7. Acknowledgment

This work was supported by the National Key Research and Development Project of China (2021ZD0110400 No. 2018AAA0101900), National Natural Science Foundation of China (U19B2042), The University Synergy Innovation Program of Anhui Province (GXXT-2021-004), Zhejiang Lab (2021KE0AC02), Academy Of Social Governance Zhejiang University, Fundamental Research Funds for the Central Universities (226-2022-00064), Artificial Intelligence Research Foundation of Baidu Inc., Program of ZJU and Tongdun Joint Research Lab.

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*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Al-Shedivat, M., Gillenwater, J., Xing, E., and Rostamizadeh, A. Federated learning via posterior averaging: A new perspective and practical algorithms, 2021.

Anthony, M. Generalization error bounds for threshold decision lists. *Journal of Machine Learning Research*, 5 (Feb):189–217, 2004.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings, 2019.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019.

Charles, Z. and Konečný, J. Convergence and accuracy trade-offs in federated learning and meta-learning. In Banerjee, A. and Fukumizu, K. (eds.), *The 24th International Conference on Artificial Intelligence and*

Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event, volume 130 of *Proceedings of Machine Learning Research*, pp. 2575–2583. PMLR, 2021. URL http://proceedings.mlr.press/v130/charles21a.html.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Chen, Z., Li, B., Xu, J., Wu, S., Ding, S., and Zhang, W. Towards practical certifiable patch defense with vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15148–15158, June 2022.

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.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

He, H. and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

Jamal, M. A., Brown, M., Yang, M.-H., Wang, L., and Gong, B. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7610–7619, 2020.

Jørgensen, A., Kamma, L., and Larsen, K. G. Near-tight margin-based generalization bounds for support vector machines. *ArXiv*, abs/2006.02175, 2020.

Kairouz, P. and McMahan, H. B. Advances and open problems in federated learning, 2021.

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

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.

Kim, B. and Kim, J. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020.

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

Li, B., Sun, Z., and Guo, Y. Supervae: Superpixelwise variational autoencoder for salient object detection. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 8569–8576. AAAI Press, 2019a. doi: 10.1609/aaai.v33i01.33018569. URL https://doi.org/10.1609/aaai.v33i01.33018569.

Li, B., Sun, Z., Li, Q., Wu, Y., and Hu, A. Group-wise deep object co-segmentation with co-attention recurrent neural network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 8518–8527. IEEE, 2019b. doi: 10.1109/ICCV.2019.00861.

Li, B., Sun, Z., Tang, L., and Hu, A. Two-b-real net: Two-branch network for real-time salient object detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 1662–1666. IEEE, 2019c. doi: 10.1109/ICASSP.2019.8683022. URL https://doi.org/10.1109/ICASSP.2019.8683022.

Li, B., Sun, Z., Tang, L., Sun, Y., and Shi, J. Detecting robust co-saliency with recurrent co-attention neural network. In Kraus, S. (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 818–825. ijcai.org, 2019d.

Li, B., Sun, Z., Wang, Q., and Li, Q. Co-saliency detection based on hierarchical consistency. In Amsaleg, L., Huet, B., Larson, M. A., Gravier, G., Hung, H., Ngo, C., and Ooi, W. T. (eds.), *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pp. 1392–1400. ACM, 2019e.

Li, B., Xu, J., Wu, S., Ding, S., Li, J., and Huang, F. Detecting adversarial patch attacks through global-local consistency. In Song, D., Tao, D., Yuille, A. L., Anandkumar, A., Liu, A., Chen, X., Li, Y., Xiao, C., Yang, X., and Liu, X. (eds.), *ADVM '21: Proceedings of the 1st International Workshop on Adversarial Learning for Multimedia, Virtual Event, China, 20 October 2021*, pp. 35–41. ACM,

2021a. doi: 10.1145/3475724.3483606. URL https://doi.org/10.1145/3475724.3483606.

Li, Q., Wen, Z., and He, B. Practical federated gradient boosting decision trees. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4642–4649. AAAI Press, 2020a. URL https://aaai.org/ojs/index.php/AAAI/article/view/5895.

Li, Q., Diao, Y., Chen, Q., and He, B. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021b.

Li, Q., He, B., and Song, D. Practical one-shot federated learning for cross-silo setting. In Zhou, Z. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pp. 1484–1490. ijcai.org, 2021c. doi: 10.24963/ijcai.2021/205. URL https://doi.org/10.24963/ijcai.2021/205.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In Dhillon, I. S., Papailiopoulos, D. S., and Sze, V. (eds.), *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020b. URL https://proceedings.mlsys.org/book/316.pdf.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Feddane: A federated newton-type method, 2020c.

Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization, 2021d.

Li, X.-C. and Zhan, D.-C. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 995–1005, 2021.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., and Ma, X. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34, 2021e.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

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.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Reyzin, L. and Schapire, R. How boosting the margin can also boost classifier complexity. *Proceedings of the 23rd international conference on Machine learning*, 2006.

Shen, T., Zhang, J., Jia, X., Zhang, F., Huang, G., Zhou, P., Wu, F., and Wu, C. Federated mutual learning. *CoRR*, abs/2006.16765, 2020. URL https://arxiv.org/abs/2006.16765.

Tang, L. and Li, B. CLASS: cross-level attention and supervision for salient objects detection. In Ishikawa, H., Liu, C., Pajdla, T., and Shi, J. (eds.), *Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020, Revised Selected Papers, Part III*, volume 12624 of *Lecture Notes in Computer Science*, pp. 420–436. Springer, 2020. doi: 10.1007/978-3-030-69535-4_26. URL https://doi.org/10.1007/978-3-030-69535-4_26.

Tang, L., Li, B., Zhong, Y., Ding, S., and Song, M. Disentangled high quality salient object detection. In *2021*

IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, pp. 3560–3570. IEEE, 2021.

Tang, L., Li, B., Kuang, S., Song, M., and Ding, S. Rethinking the relations in co-saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D. S., and Khazaeni, Y. Federated learning with matched averaging. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 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. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b.

Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H. B., y Arcas, B. A., Al-Shedivat, M., Andrew, G., Avestimehr, S., Daly, K., Data, D., Diggavi, S., Eichner, H., Gadhikar, A., Garrett, Z., Girgis, A. M., Hanzely, F., Hard, A., He, C., Horvath, S., Huo, Z., Ingerman, A., Jaggi, M., Javidi, T., Kairouz, P., Kale, S., Karimireddy, S. P., Konecny, J., Koyejo, S., Li, T., Liu, L., Mohri, M., Qi, H., Reddi, S. J., Richtarik, P., Singhal, K., Smith, V., Soltanolkotabi, M., Song, W., Suresh, A. T., Stich, S. U., Talwalkar, A., Wang, H., Woodworth, B., Wu, S., Yu, F. X., Yuan, H., Zaheer, M., Zhang, M., Zhang, T., Zheng, C., Zhu, C., and Zhu, W. A field guide to federated optimization, 2021a.

Wang, L., Xu, S., Wang, X., and Zhu, Q. Addressing class imbalance in federated learning, 2020c.

Wang, L., Xu, S., Wang, X., and Zhu, Q. Addressing class imbalance in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10165–10173, 2021b.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, T. N., and Khazaeni, Y. Bayesian nonparametric federated learning of neural networks, 2019.

Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Xu, J., Ding, S., and Wu, C. A practical data-free approach to one-shot federated learning with heterogeneity. *arXiv preprint arXiv:2112.12371*, 2021a.

Zhang, J., Li, B., Xu, J., Wu, S., Ding, S., Zhang, L., and Wu, C. Towards efficient data free black-box adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15115–15125, June 2022.

Zhang, X., Hong, M., Dhople, S., Yin, W., and Liu, Y. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data, 2020.

Zhang, Y., Wei, X.-S., Zhou, B., and Wu, J. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3447–3455, 2021b.

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

Zhong, Y., Li, B., Tang, L., Tang, H., and Ding, S. Highly efficient natural image matting. *CoRR*, abs/2110.12748, 2021. URL https://arxiv.org/abs/2110.12748.

Zhong, Y., Li, B., Tang, L., Kuang, S., Wu, S., and Ding, S. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4504–4513, June 2022.

Appendix

The Appendix is organized as follows:

- Appendix A provides more details on experimental setup and also some additional experiments.
- Appendix B provides analysis for local update in detail.
- Appendix C reveals the meaning of deviation bound and gives proofs for our Theorem 1,3 and equations in Definition 3 related to deviation bound.
- Appendix D gives proofs for our Theorem 2 about selecting the optimal pairwise label margin.

A. Experimental Details

A.1. Model Architectures

Note that the state-of-the-art algorithms have achieved impressive performance on MNIST, FMNIST, and CIFAR10. Since our goal is to evaluate the performance of several baselines in the same settings rather than achieve the best accuracy on these datasets, we use two CNN architectures in our experiments, which are sufficient for our needs.

The CNN architectures we use consist of several convolutional layers followed by fully connected layers and then a linear transformation layer to produce logits. The shape for convolutional layers follows $(c_{in}, c_{out}, c_{kernel}, c_{kernel})$ and (c_{in}, c_{out}) for fully connected layers. All non-linear activation function in these architectures is ReLu. We show detailed information in Table 8 and Table 9.

Table 8. Detailed information of the CNN architecture used in our experiments for MNIST and FMNIST.

Parameter	Shape	Layer hyper-parameter
conv1.weight	[1, 10, 5, 5]	stride=1, padding=0
conv1.bias	[10]	N/A
pooling.max	N/A	stride=2,padding=2
conv2.weight	[10, 20, 5, 5]	stride=1, padding=0
conv2.bias	[20]	N/A
pooling.max	N/A	stride=2, padding=2
fc1.weight	[320, 50]	N/A
fc1.bias	[50]	N/A
fc2.weight	[50,10]	N/A
fc2.bias	[10]	N/A

Table 9. Detailed information of the CNN architecture used in our experiments for CIFAR10.

Parameter	Shape	Layer hyper-parameter
conv1.weight	[3, 128, 3, 3]	stride=1, padding=0
conv1.bias	[128]	N/A
pooling.max	N/A	stride=2, padding=2
conv2.weight	[128, 128, 3, 3]	stride=1, padding=0
conv2.bias	[128]	N/A
pooling.max	N/A	stride=2, padding=2
conv3.weight	[128, 128, 3, 3]	stride=1, padding=0
conv3.bias	[128]	N/A
pooling.max	N/A	stride=2, padding=2
fc1.weight	[2048,10]	N/A
fc1.bias	[10]	N/A

Algorithm 1 *Federated Learning via Logits Calibration (for Client)*

Input: local datasets $\mathcal{D}^i$, E local epochs , learning rate η

LocalUpdate(w^t):
  $w \leftarrow w^t$ // download global model
  for epoch $k = 1, 2, ..., E$ **do**
    for each batch $\mathbf{b} = \{x, y\}$ of $\mathcal{D}^i$ **do**
      $L(w; b) = \mathcal{L}_{\text{Cal}}(w; x)$// Equation 7
      $w \leftarrow w - \eta \nabla L(w; b)$
  return w to the server

B. Analysis for Update Process

This part we analyse the update process for $\{w_y\}_{y \in K}$ during the local training. According to (Wang et al., 2020c), in the same epoch, the extracted feature $h(x)$ of samples in the same class will be similar and thus the corresponding output probability vector $p(x)$ will be similar. Therefore, for any class y, we could calculate their average value, $\overline{h^{(y)}} = \frac{1}{n_y} \sum_{i \in O_y} h(x_i)$, $\overline{p^{(y)}} = \frac{1}{n_y} \sum_{i \in O_y} p(x_i)$ and $\overline{p_k^{(y)} h^{(y)}} = \frac{1}{n_y} \sum_{i \in O_y} p_k(x_i) h(x_i)$ averaged over all samples of class y. Based on the statistical data provided by (Wang et al., 2020c), the following property exists.

Property 1. *When the feature $h(x)$ over class y is similar, then $\overline{p_k^{(y)} h^{(y)}} \approx \overline{p_k^{(y)}}\,\overline{h^{(y)}}, \forall i \in K$,and the upper bound can be represented as*

$$||\overline{p_k^{(y)} h^{(y)}} - \overline{p_k^{(y)}}\,\overline{h^{(y)}}||_2 \leq \sigma_{k,y} \sqrt{(\sqrt{{c_v^{(y)}}^2 + 1} - \theta^{(y)}) \overline{||h^{(y)}||_2}} \tag{8}$$

where $c_v^{(y)}$ is the average coefficient of variation for multiplications $h(x_i) \cdot h(x_{i'})$ between all pairs of $h(x)$ over the same class y, $\theta^{(y)}$ is the average cosine similarity score of all pairs of $h(x)$ over class y, $\sigma_{k,y}$ is the standard deviation of p_k over the class y and $\overline{||h^{(y)}||_2} = \frac{1}{n_y} \sum_{i \in O_y} ||h(x_i)||_2$ is the average value of L^2 norm $||h(x)||_2$ over class y. Here, $c_v^{(y)}$ and $\theta^{(y)}$ are provided by (Wang et al., 2020c).

Then by the property above of $\overline{h^{(y)}}$ and $\overline{p^{(y)}}$, for softmax cross-entropy, the update process can be described as:

$$\begin{aligned}
\Delta w_y &= \eta \sum_{i \in O_y} (1 - p_y(x_i)) h(x_i) - \eta \sum_{y' \neq y} \sum_{i \in O_{y'}} p_y(x_i) h(x_i) \\
&= \eta \overline{[(1 - p_y(x_i)) h(x_i)]^{(y)}} n_y - \eta \sum_{y' \neq y} \overline{[p_y(x_i) h(x_i)]^{(y')}} n_{y'} \\
&\approx \eta (1 - \overline{[p_y(x_i)]^{(y)}})\overline{[h(x_i)]^{(y)}} n_y - \eta \sum_{y' \neq y} \overline{[p_y(x_i)]^{(y')} [h(x_i)]^{(y')}} n_{y'} \\
&= \eta (1 - \overline{p_y^{(y)}})\overline{h^{(y)}} n_y - \eta \sum_{y' \neq y} \overline{p_y^{(y')}}\,\overline{h^{(y')}} n_{y'}
\end{aligned} \tag{9}$$

where η is the learning rate and $\overline{[f(x_i)]^{(y)}} = \frac{1}{n_y} \sum_{i \in O_y} f(x_i)$,for any function $f(\cdot)$. According to the **property 1**, we replace "$\approx$" with "$=$" by omitting small error in the following contents.

Recall that $p_k(x_i) = \frac{w_k h(x_i)}{\sum_{y \in K} w_y h(x_i)}$. To reduce the classification error, for sample x_i in class k ,vector w_k is expected to be close to $h(x_i)$ and vector w_y,$y \neq k$ is expected to be away from $h(x_i)$, which will increase the probability $p_k(x_i)$. Therefore, during the update process, $\Delta w_k \overline{h^{(k)}}$ is expected to be positive and $\Delta w_y \overline{h^{(k)}}$ is expected to be negative.

The proof for **property 1** is as follow:

Assumption 1. *The cosine similarity score of different $h(x_i)$ of class y is independent with the L^2 norm of $h(x_i)$. Approximately, this assumption says that the length of vector $h(x_i)$ is independent with angles between different vectors.*

Here we consider one class y and without ambiguity, we use $\overline{f(x_i)}$ denote average value $\frac{1}{|O_y|} \sum_{i \in O_y} f(x_i)$ for all function $f(\cdot)$ and use $\overline{g(x_i, x_{i'})}$ denote average value $\frac{1}{|O_y|^2} \sum_{i,i' \in O_y} g(x_i, x_{i'})$ for all function $g(\cdot, \cdot)$. For simplicity, $\overline{||h^{(y)}||_2^k}$ is

used to represent $\overline{||h(x_i))||_2^k}$ for $k = 1, 2, 4$. Based on the assumption, we could have the following equations.

$$\begin{aligned}(\overline{h^{(y)}})^2 &= \frac{1}{n_y^2}\sum_{i,i'\in O_y}||h(x_i)||_2||h(x_{i'})||_2\cos < h(x_i), h(x_{i'}) > \\ &= \overline{||h(x_i)||_2||h(x_{i'})||_2\cos < h(x_i), h(x_{i'}) >} \\ &= \overline{||h(x_i)||_2||h(x_{i'})||_2}\ \overline{\cos < h(x_i), h(x_{i'}) >} \\ &= \overline{||h^{(y)}||_2}^2\theta^{(y)}\end{aligned} \tag{10}$$

$$\begin{aligned}\frac{1}{n_y^2}\sum_{i,i'\in O_y}(h(x_i)\cdot h(x_{i'}))^2 &= \frac{1}{n_y^2}\sum_{i,i'\in O_y}||h(x_i)||_2^2||h(x_{i'})||_2^2\cos^2 < h(x_i), h(x_{i'}) > \\ &= \overline{||h(x_i)||_2^2||h(x_{i'})||_2^2\cos^2 < h(x_i), h(x_{i'}) >} \\ &= \overline{||h^{(y)}||_2^2}\,\overline{\cos^2 < h(x_i), h(x_{i'}) >}\end{aligned} \tag{11}$$

Then we begin to calculate the average value $\overline{p_k^{(y)}h^{(y)}}$

$$\begin{aligned}\frac{1}{n_y}\sum_{i\in O_y}p_k(x_i)h(x_i) &= \frac{1}{n_y}\sum_{i\in O_y}(\Delta p_k(x_i) + \overline{p_k^{(y)}})(\Delta h(x_i) + \overline{h^{(y)}}) \\ &= \overline{p_k^{(y)}}\,\overline{h^{(y)}} + \frac{1}{n_y}\sum_{i\in O_y}\Delta p_k(x_i)\Delta h(x_i)\end{aligned} \tag{12}$$

Here $\Delta p_k(x_i)$ and $\Delta h(x_i)$ represent $p_k(x_i) - \overline{p_k^{(y)}}$ and $h(x_i) - \overline{h^{(y)}}$ respectively. Then combining the proposition of $\Delta p_k(x_i)$, that is, $|\Delta p_k(x_i)| < 1$ and the equation 10 12, we have the following inequation.

$$\begin{aligned}||\overline{p_k^{(y)}h^{(y)}} - \overline{p_k^{(y)}}\,\overline{h^{(y)}}||_2 &= ||\frac{1}{n_y}\sum_{i\in O_y}\Delta p_k(x_i)\Delta h(x_i)||_2 \\ &\le \sqrt{(\frac{1}{n_y}\sum_{i\in O_y}||\Delta h(x_i)||_2^2)(\frac{1}{n_y}\sum_{i\in O_y}\Delta p_k(x_i)^2)} \\ &= \sigma_{k,y}\sqrt{\overline{||h^{(y)}||_2^2} - \overline{h^{(y)}}\cdot\overline{h^{(y)}}} \\ &= \sigma_{k,y}\sqrt{\overline{||h^{(y)}||_2^2} - \overline{||h^{(y)}||_2}^2\theta^{(y)}}\end{aligned} \tag{13}$$

Based on $\frac{1}{n_y^2}\sum_{i,i'\in O_y}h(x_i)\cdot h(x_{i'}) = (\frac{1}{n_y}\sum_{i\in O_y}h(x_i))(\frac{1}{n_y}\sum_{i'\in O_y}h(x_{i'})) = \overline{h^{(y)}}\cdot\overline{h^{(y)}}$ and $\overline{cos^2 < h(x_i), h(x_{i'}) >} \ge \overline{cos < h(x_i), h(x_{i'}) >}^2 = {\theta^{(y)}}^2$, we use following equations and inequations to finish the proof:

$$\begin{aligned}\sqrt{\overline{h^{(y)}}\cdot\overline{h^{(y)}}}c_v^{(y)} &= \sqrt{\frac{1}{n_y^2\overline{h^{(y)}}\cdot\overline{h^{(y)}}}\sum_{i,i'\in O_y}(h(x_i)h(x_{i'}) - \overline{h(x_i)}\cdot\overline{h(x_{i'})})^2} \\ &= \sqrt{\frac{1}{n_y^2\overline{h^{(y)}}\cdot\overline{h^{(y)}}}\sum_{i\in O_y}\sum_{i'\in O_y}(h(x_i)h(x_{i'}))^2 - (\overline{h^{(y)}}\cdot\overline{h^{(y)}})^2} \\ &= \sqrt{\frac{\overline{||h^{(y)}||_2^2}^2\,\overline{\cos^2 < h(x_i), h(x_{i'}) >} - \overline{||h^{(y)}||_2}^4{\theta^{(y)}}^2}{\overline{||h^{(y)}||_2}^2\theta^{(y)}}} \\ &\ge \sqrt{\frac{\overline{||h^{(y)}||_2^2}^2\theta^{(y)^2} - \overline{||h^{(y)}||_2}^4{\theta^{(y)}}^2}{\overline{||h^{(y)}||_2}^2\theta^{(y)}}}\end{aligned} \tag{14}$$

By algebraic deformation, the equation above is equivalent to:

$$\overline{||h^{(y)}||_2^2} \leq \sqrt{{c_v^{(y)}}^2\overline{||h^{(y)}||_2}^4 + \overline{||h^{(y)}||_2^4}} = \sqrt{{c_v^{(y)}}^2 + 1}\overline{||h^{(y)}||_2}^2 \tag{15}$$

Then, by equations 13 15 above we have:

$$||\overline{p_k^{(y)}h^{(y)}} - \overline{p_k^{(y)}}\,\overline{h^{(y)}}||_2 \leq \sigma_{k,y}\sqrt{\overline{||h^{(y)}||_2^2} - \overline{||h^{(y)}||_2}^2\theta^{(y)}} \leq \sigma_{k,y}\sqrt{(\sqrt{{c_v^{(y)}}^2 + 1} - \theta^{(y)})}\overline{||h^{(y)}||_2} \tag{16}$$

C. Missing Proofs for Deviation Bound

Before this part, please refer to Appendix B for some notations and analysis of local update in detail. In this part, first we will present the proof of **Theorem 1** and the meaning of **deviation bound** D_{jr} will be revealed in the proof.

Proof of Theorem 1

According to **Appendix B**, during update process, $\Delta w_r\overline{h^{(r)}}$ is expected to be positive and $\Delta w_j\overline{h^{(r)}}$ is expected to be negative. To prove the **Theorem 1**, we will show that when $n_j/n_r \gg D_{jr} > 0$, $\Delta w_r\overline{h^{(r)}}$ is much more likely to be negative and $\Delta w_j\overline{h^{(r)}}$ is much more likely to be positive, which deviate from out expectation. Based on equation 9, $\Delta w_r\overline{h^{(r)}}$ and $\Delta w_j\overline{h^{(r)}}$ can be expressed as:

$$\begin{aligned}
\Delta w_r\overline{h^{(r)}} &= \eta(1 - \overline{p_r^{(r)}})\overline{h^{(r)}} \cdot \overline{h^{(r)}}n_r - \eta\sum_{k\neq r}\overline{p_r^{(k)}}\overline{h^{(k)}} \cdot \overline{h^{(r)}}n_k \\
\Delta w_j\overline{h^{(r)}} &= \eta(1 - \overline{p_j^{(j)}})\overline{h^{(j)}} \cdot \overline{h^{(r)}}n_j - \eta\sum_{k\neq j}\overline{p_j^{(k)}}\overline{h^{(k)}} \cdot \overline{h^{(r)}}n_k
\end{aligned} \tag{17}$$

For $\Delta w_r\overline{h^{(r)}}$, consider the items for class r and class j, which are $(1 - \overline{p_r^{(r)}})\overline{h^{(r)}} \cdot \overline{h^{(r)}}n_r$ and $\overline{p_r^{(j)}}\overline{h^{(j)}} \cdot \overline{h^{(r)}}n_j$. By algebraic deformation, it is obviously that when $n_j/n_r > D_{jr} = \frac{(1-\overline{p_r^{(r)}})\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}} > 0$, $(1 - \overline{p_r^{(r)}})\overline{h^{(r)}} \cdot \overline{h^{(r)}}n_r - \overline{p_r^{(j)}}\overline{h^{(j)}} \cdot \overline{h^{(r)}}n_j < 0$ exists. Then without omitting other items, for class j which is extremely large, $n_j \gg 0$, when $n_j/n_r \gg D_{jr} > 0$, we can say that $\Delta w_r\overline{h^{(r)}}$ is much more likely to be negative.

For $\Delta w_j\overline{h^{(r)}}$, consider the items for class j and class r, which are $(1 - \overline{p_j^{(j)}})\overline{h^{(j)}} \cdot \overline{h^{(r)}}n_j$ and $\overline{p_j^{(r)}}\overline{h^{(r)}} \cdot \overline{h^{(r)}}n_r$. Note that $(1 - \overline{p_j^{(j)}})(1 - \overline{p_r^{(r)}}) \geq \overline{p_j^{(r)}} \cdot \overline{p_r^{(j)}}$ and thus $\frac{(1-\overline{p_r^{(r)}})\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}} \geq \frac{\overline{p_j^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{(1-\overline{p_j^{(j)}})\overline{h^{(j)}}\cdot\overline{h^{(r)}}} > 0$ exists when $n_j/n_r > D_{jr} > 0$. By algebraic deformation, it is obviously that when $n_j/n_r > D_{jr} = \frac{(1-\overline{p_r^{(r)}})\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}} \geq \frac{\overline{p_j^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{(1-\overline{p_j^{(j)}})\overline{h^{(j)}}\cdot\overline{h^{(r)}}} > 0$, $(1 - \overline{p_j^{(j)}})\overline{h^{(j)}} \cdot \overline{h^{(r)}}n_j - \overline{p_j^{(r)}}\overline{h^{(r)}} \cdot \overline{h^{(r)}}n_r > 0$ exists. Then without omitting other items, for class j which is extremely large, $n_j \gg 0$, when $n_j/n_r \gg D_{jr} > 0$, we can say that $\Delta w_j\overline{h^{(r)}}$ is much more likely to be positive.

Proof of Definition 3

In the proof above, we have shown the meaning of the definition $D_{jr} = \frac{(1-\overline{p_r^{(r)}})\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}}$. The equation $D_{jr} = \frac{(1-\overline{p_r^{(r)}})\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}} = \sum_{y\in K, y\neq r}\frac{\overline{p_y^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}}$ in **Definition 3** exists, based on the following equation:

$$(1 - \overline{p_r^{(r)}}) = \frac{1}{n_r}\sum_{i\in O_r}(1 - p_r(x_i)) = \frac{1}{n_r}\sum_{i\in O_r}\sum_{y\neq r}p_y(x_i) = \sum_{y\in K, y\neq r}\overline{p_y^{(r)}} \tag{18}$$

Then we present the proof of **Theorem 3**.

Proof of Theorem 3

After adding the margin $\Delta_{(y,i)} = \tau(n_y^{-1/4} - n_i^{-1/4})$, according to equation 7, the probability $p_y(x)$ becomes $p_y(x) =$

$\frac{e^{f_y(x)}-\tau n_y^{-1/4}}{\sum_{k=1}^K e^{f_k(x)-\tau\cdot n_k^{-1/4}}}$. Consider the definition of $\widetilde{p}_y(x)=\frac{e^{f_y(x)}}{\sum_{y=1}^K e^{f_k(x)-\tau\cdot n_k^{-1/4}}}$, the following equations exist.

$$\overline{p_y^{(r)}}=\frac{1}{n_r}\sum_{i\in O_r}p_y(x_i)=\frac{1}{n_r}\sum_{i\in O_r}e^{-\tau n_y^{-1/4}}\widetilde{p}_y(x_i)=e^{-\tau n_y^{-1/4}}\frac{1}{n_r}\sum_{i\in O_r}\widetilde{p}_y(x_i)=e^{-\tau n_y^{-1/4}}\overline{\widetilde{p}_y^{(r)}} \tag{19}$$

$$\overline{p_r^{(j)}}=\frac{1}{n_j}\sum_{i\in O_j}p_r(x_i)=\frac{1}{n_j}\sum_{i\in O_j}e^{-\tau n_r^{-1/4}}\widetilde{p}_r(x_i)=e^{-\tau n_r^{-1/4}}\frac{1}{n_j}\sum_{i\in O_j}\widetilde{p}_r(x_i)=e^{-\tau n_r^{-1/4}}\overline{\widetilde{p}_r^{(j)}} \tag{20}$$

Then the **deviation bound** becomes the following formation, which finishes the proof.

$$D_{jr}=\sum_{y\in K,y\neq r}\frac{\overline{p_y^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{p_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}}=\sum_{y\in K,y\neq r}\frac{e^{-\tau n_y^{-1/4}}\overline{\widetilde{p}_y^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{e^{-\tau n_r^{-1/4}}\overline{\widetilde{p}_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}}=\sum_{y\in K,y\neq r}\Delta_{(r,y)}\frac{\overline{\widetilde{p}_y^{(r)}}\overline{h^{(r)}}\cdot\overline{h^{(r)}}}{\overline{\widetilde{p}_r^{(j)}}\overline{h^{(j)}}\cdot\overline{h^{(r)}}} \tag{21}$$

D. Missing Theorems and Proofs for Optimal Margin

Margin Error Bounds for skewed Data For each client, we focus on K-classes classification, and $D=(x_i,y_i)$ is the training data of i-th client. Let $f_{\boldsymbol{\theta}}(x)=\left(f_{\boldsymbol{\theta}}^1(x),\cdots,f_{\boldsymbol{\theta}}^K(x)\right)$ denote the score function of the local model, and the predicted label is $\hat{y}=\arg\max f_{\boldsymbol{\theta}}(x)$. For an input data (x,y), the margin is defined as:

$$d_{\boldsymbol{\theta}}(x,y)=f_{\boldsymbol{\theta}}^y(x)-\max_{j\neq y}f_{\boldsymbol{\theta}}^j(x). \tag{22}$$

Then the training margin of class j is :

$$d_{\boldsymbol{\theta}}^j=\min_{i\in S_j}d_{\boldsymbol{\theta}}(x_i,y_i). \tag{23}$$

Where $S_j=\{i:y_i=j\}$ is the example indices corresponding to class j. A large body of recent work has examined margin analysis for measuring generalization error bounds (Anthony, 2004; Jørgensen et al., 2020; Reyzin & Schapire, 2006). In this study, we extend these analyses to the context of skewed data in FL.

Theorem 4. *Let $L_\theta^{01}(x,y)=\mathbb{I}[yf(x)\leq 0]$ denote the 0-1 loss and $y\in\{+1,-1\}$. Let F be a set of real-valued functions that $f\in F$, and we have $\sup_{\mathbf{x}\in\mathcal{X}}|f(\mathbf{x})|\leq C$. Note that $R_n(F)$ is the Rademacher complexity of a function class F, and n is the size of samples. Then, with probability at least $1-\delta$ over the sample, for all margins $d>0$ and all $f\in F$ we have,*

$$L(f)\lesssim 4\frac{R_n(F)}{d}+\sqrt{\frac{\log\left(\log_2\frac{4C}{d}\right)}{n}}+\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{24}$$

Where $L(f)=\mathbb{E}[\ell(f(x),y)]$ is the expected of loss of $f:\mathcal{X}\to\mathbb{R}$. Thus with the rademacher complexity upper bound $R(F)\leq\sqrt{\frac{\Gamma(F)}{n}}$, for some complexity measure $\Gamma(F)$, the test error bound can be scaled as:

$$L(f)\lesssim\frac{4}{d}\sqrt{\frac{\Gamma(F)}{n}}<\frac{1}{d\sqrt{n}}.$$

For ease of presentation, we start from $K=2$ for binary classification. We apply Theorem 4 in binary classification, the generalization error bound can be simply formulated as:

$$\frac{1}{d_1\sqrt{n_1}}+\frac{1}{d_2\sqrt{n_2}}. \tag{25}$$

Obviously, the bound is related with the label distribution and sample size. Let Δ denote the optimal threshold of margins between minority classes and majority classes. The optimal value should satisfy $d_1'=d_1-\Delta$ and $d_2'=d_2+\Delta$. A illustration is in shown in Figure 8. To encourage the minority classes to have larger margins, the optimization of error bound can be formulated as:

$$\min_{d_1'+d_2'=Q}\frac{1}{d_1'\sqrt{n_1}}+\frac{1}{d_2'\sqrt{n_2}},$$

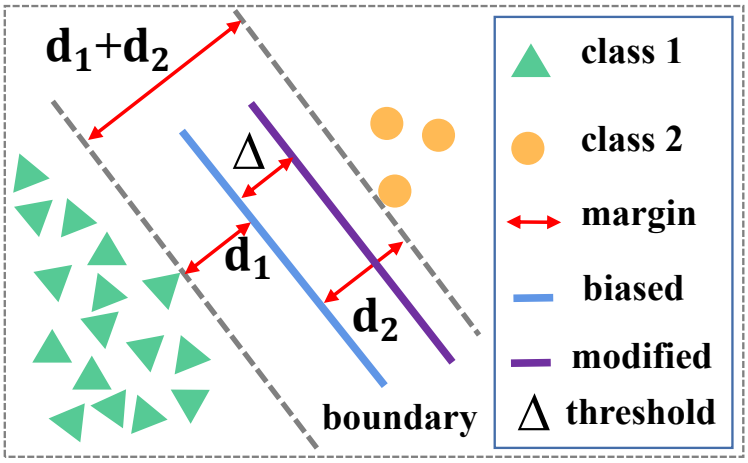

Figure 8. For binary classification, with a fixed margin of $d_1 + d_2$, we can slightly modify the direction of the decision boundary by Δ. Then the modified margin of each class is $d_1 + \Delta$ and $d_2 - \Delta$.

where we suppose a total sum of margins $d_1 + d_2 = Q$. Let the derivative be 0, we can get (see a detailed proof in D.2)

$$d_1^\star = \frac{Q n_2^{1/4}}{n_1^{1/4} + n_2^{1/4}}, \; d_2^\star = \frac{Q n_1^{1/4}}{n_1^{1/4} + n_2^{1/4}}. \tag{26}$$

Inspired by Equation 26, Δ can be set as a label independent variable, with multiple classes of the form:

$$\Delta = \tau \cdot n_i^{-1/4}, \; i = 1, 2, \cdots, K. \tag{27}$$

Where τ is a constant. Thus, the logits (the output of the last fully connected layer) can be modified by the enforced margins Δ to adjust the biased decision boundary.

D.1. Proof of Theorem 4

Recall the definition of the Rademacher complexity of a function class H, then with IID assumption, a standard normal random variables ϵ_i independently takes values in {-1,+1} with equal probability. Then

$$\mathcal{R}_n(\mathcal{H}) = \mathbb{E}\left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} h\left(\mathbf{x}_i\right) \epsilon_i\right] \tag{28}$$

Define the margin of an example (x, y) as $\beta(x, y) = h(x)_y - \max_{j \neq y} h(x)_j$, then training margin for class j can be defined as

$$\beta_j = \min_{i \in L_j} \beta\left(x_i, y_i\right) \tag{29}$$

where $L_j = \{i : y_i = j\}$ denote the example indices corresponding to class j.

For any margin $\beta > 0$, we define margin loss as bellow:

$$\ell_\beta(x) = \begin{cases} 0 & \text{if } \beta \leq x \\ 1 - x/\beta & \text{if } 0 \leq x \leq \beta \\ 1 & \text{if } x \leq 0 \end{cases} \tag{30}$$

we can apply the standard margin-based generalization bound to margin loss function $\ell_\beta(x)$ which has Lipschitz constant $1/\beta$. Then fix $\beta > 0$, then for any $\delta > 0$, with probability at least $1 - \delta$, each of the following holds for all $p \in (0, 1)$ and $h \in H$:

$$R(h) \leq \widehat{R}_\beta(h) + \frac{4}{\beta} \mathfrak{R}_m(H) + \sqrt{\frac{\log \log_2 \frac{2}{\beta}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} \tag{31}$$

where we let $R(h)$ and $\widehat{R}_{\beta}(h)$ denote the expected and empirical loss based on margin loss. Thus, we remove the low order term, we use $C(H)$ to be some complexity measure of $\mathfrak{R}_m(H)$, then the test error can be bounded as:

$$R(h) \lesssim \sqrt{\frac{C(H)}{n \cdot \beta_{min}^2}} \tag{32}$$

D.2. Proof of Equation 26

Since for a fixed margin $\beta_1 + \beta_2 = \beta$, we modify the decision boundary to be closer to majority class, the we have $\beta_2 = \beta - \beta_1$. Now we apply Theorem 1 to get the generalization error bound for binary classification:

$$\text{test error} \lesssim \sqrt{\frac{C(H)}{n_1 \cdot \beta_1^2}} + \sqrt{\frac{C(H)}{n_2 \cdot \beta_2^2}} \tag{33}$$

To solve this question, we should solve

$$\min_{\beta_1+\beta_2=\beta} \frac{1}{\beta_1}\sqrt{\frac{1}{n_1}} + \frac{1}{\beta_2}\sqrt{\frac{1}{n_2}} \tag{34}$$

Accordingly, for formula 34, we set it derivative to 0, obtaining

$$\frac{1}{(\beta - \beta_1)^2 \sqrt{n_2}} - \frac{1}{\beta_1^2 \sqrt{n_1}} = 0 \tag{35}$$

Then we can obtain

$$\beta_1^\star = \frac{\beta n_2^{1/4}}{n_1^{1/4} + n_2^{1/4}}, \quad \beta_2^\star = \frac{\beta n_1^{1/4}}{n_1^{1/4} + n_2^{1/4}} \tag{36}$$

Conclude from above proofs, we can confirm that a lower generalization bounds can be implemented by a proper margin.