Tackling Federated Long-Tailed Learning via Synthetic Feature-Based Decoupled Training

Huabin Zhu
Zhejiang University
School of Software Technology
Hangzhou, Zhejiang, China
zhb2000@zju.edu.cn

Chaochao Chen*
Zhejiang University
College of Computer Science and Technology
Hangzhou, Zhejiang, China
zjuccc@zju.edu.cn

Xinting Liao
Zhejiang University
College of Computer Science and Technology
Hangzhou, Zhejiang, China
xintingliao@zju.edu.cn

Pengyang Zhou
Zhejiang University
College of Computer Science and Technology
Hangzhou, Zhejiang, China
zhoupy@zju.edu.cn

Xiaolin Zheng
Zhejiang University
College of Computer Science and Technology
Hangzhou, Zhejiang, China
xlzheng@zju.edu.cn

Abstract

Federated learning (FL) enables collaborative training on decentralized data while preserving privacy by avoiding direct data sharing. However, long-tailed data distributions are common in real-world applications, often resulting in biased models with degraded performance. In FL, this issue is further complicated by privacy-preserving constraints and non-IID data, highlighting the importance of federated long-tailed learning (Fed-LT). To tackle the challenges of Fed-LT, we propose Synthetic Feature-based Decoupled training (SFD) method. To improve local training, we introduce Adaptive Bi-Branch Learning (ABBL) to jointly enhance feature representation and decision boundary learning for non-IID long-tailed data. To mitigate global model bias while preserving privacy, we propose Statistically Aligned Feature Synthesis (SAFS) for global classifier fine-tuning. SAFS constructs privacy-preserving synthetic features that approximate the global feature distribution. These synthetic features enable the global classifier to be fine-tuned without requiring clients to share local training data, thereby alleviating the model bias caused by non-IID long-tailed data. Extensive experiments show that SFD effectively addresses the challenges of Fed-LT and achieves superior performance on Fed-LT datasets.

CCS Concepts

• **Computing methodologies → Cooperation and coordination**; *Neural networks.*

Keywords

federated learning; long-tailed learning; class imbalance; non-IID

ACM Reference Format:
Huabin Zhu, Chaochao Chen, Xinting Liao, Pengyang Zhou, and Xiaolin Zheng. 2025. Tackling Federated Long-Tailed Learning via Synthetic Feature-Based Decoupled Training. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737143

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15564922.

1 Introduction

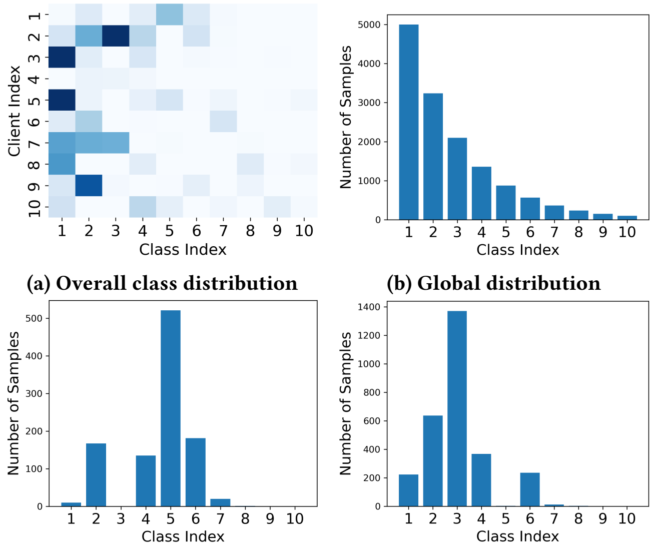
The widespread of long-tailed data distributions brings significant challenges for real-world machine learning systems, where a small subset of head classes dominate the majority of samples while most tail classes struggle with scarcity. Traditional centralized approaches for long-tailed learning become inadequate in emerging privacy-sensitive scenarios, where data ownership and privacy protection constraints prevent direct data sharing. This limitation has propelled federated learning (FL) to the forefront as a privacy-preserving machine learning paradigm. However, the intersection of FL with long-tailed learning—termed federated long-tailed learning (Fed-LT)—introduces unique complexities that remain underexplored. Compared with centralized long-tailed learning, Fed-LT is more difficult due to the combined difficulties of class imbalance, non-IID data, and privacy preserving. As shown in Fig. 1, in Fed-LT, both the global and local distributions are long-tailed, but these distributions are inconsistent with each other due to non-IID. Such inconsistency further deteriorates the performance of the global model [39].

To address the issues in Fed-LT, existing works generally focus on two main aspects. Firstly, some works [3, 33, 36, 40] focus on addressing the class imbalance issue during local training. These methods enhance the performance of the aggregated global model by reducing the bias of local models. However, they do not directly

*Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3737143



(a) Overall class distribution

(b) Global distribution

(c) Local distribution of client 1

(d) Local distribution of client 2

Figure 1: An example of Fed-LT, where the local distributions of each client and the global distribution exhibit inconsistent long-tailed class distributions. In subplot (a), darker colors in the heatmap represent a higher number of samples.

address the bias of global model, which may still result in a suboptimal global model due to the inconsistent local data distributions across clients. Secondly, some approaches [10, 17, 22, 28] focus on adjusting the global model's classifier, as the classifier (classification head), located in the model's last layer, tends to exhibit the most detrimental model bias [14, 22]. FedNCM [17] and Fed3R [10] leverage closed-form solution classifiers, using nearest-class-means and ridge regression respectively. However, these classifiers underperform compared with standard softmax classifiers. CCVR [22] and CReFF [28] fine-tune global classifiers through synthesizing features, using Gaussian mixture models and gradient matching respectively. However, they both face limitations: CCVR's Gaussian assumption often mismatches real feature distributions, and CReFF fails to adequately align feature distributions by only matching the average gradients.

In summary, Fed-LT methods face two key challenges: (1) improving local training to produce high-quality models, and (2) reducing global model bias in a privacy-preserving way. These challenges are closely intertwined, as improving local training alone is insufficient to address global model bias, while adjusting the global classifier relies on the model's ability to extract high-quality representations. However, existing methods typically focus on either local training improvement or global classifier adjustment. The former cannot directly mitigate global model bias, while the latter is limited by suboptimal feature representations—both hampering their performance in Fed-LT. Inspired by decoupled training [14], we propose Synthetic Feature-based Decoupled training (SFD) method to jointly tackle both challenges.

To address the first challenge, we introduce Adaptive Bi-Branch Learning (ABBL) to improve local training. ABBL employs adaptive adjustments based on the local data distribution of each client to enhance feature representations and refine decision boundaries. The classification branch adaptively adjusts the local training objective based on the class distribution and missing classes at each client, leading to improved decision boundaries and better recognition of long-tailed data. Meanwhile, the contrastive branch adopts an adaptive supervised contrastive loss to refine representations by adjusting gradients from negative samples. With these adaptive adjustments, the model produces well-separated features and maintains clear decision boundaries.

To address the second challenge, we propose Statistically Aligned Feature Synthesis (SAFS), where the server constructs synthetic features from aggregated statistics to approximate the global feature distribution. These synthetic features enable the global classifier to be fine-tuned without revealing clients' training data, thereby preserving privacy. This fine-tuning effectively reduces model bias caused by non-IID long-tailed data and boosts overall performance. SAFS consists of two components: (1) a MeanCov Aligner, which aligns the mean and covariance of synthetic features with the global feature distribution via Cholesky decomposition and affine transformation; and (2) a feature synthesis loss based on the random feature approximation of Maximum Mean Discrepancy (MMD), which further aligns the distribution of synthetic features with the global one. Compared with existing methods, our approach maintains an effective softmax classifier and produces high-quality synthetic features for fine-tuning, without requiring clients to share local training data.

In summary, our contributions are as follows: (1) We propose Synthetic Feature-based Decoupled training (SFD) method, which enhances both local training and global classifier fine-tuning to address the challenges of Fed-LT. (2) We propose Adaptive Bi-Branch Learning (ABBL), which improves local training by jointly enhancing feature representations and decision boundaries through adaptive, distribution-aware optimization. (3) We propose Statistically Aligned Feature Synthesis (SAFS), which leverages aggregated global statistics to construct synthetic features for global classifier fine-tuning while preserving data privacy. (4) We conduct experiments under various non-IID and class imbalance settings across three datasets, demonstrating the effectiveness of our SFD method.

2 Related Works

2.1 Long-Tailed Learning

Long-tailed data distributions are very common in real-world applications. To address the issue of degraded performance on tail classes, various long-tailed learning methods have been proposed, which can be broadly classified into three categories [41]: class re-balancing [2, 5, 20, 24], module improvement [14, 42], and information augmentation. Class re-balancing is the most commonly used approach, including techniques such as over/under-sampling, Class-Balanced Loss [5], LDAM [2], and Logit Adjustment [24]. Module improvement includes techniques such as ensemble learning, classifier design [42], and decoupled training [14]. A study [42] finds that while class re-balancing leads to good classifiers, it weakens the model's representation capability. To address this, the study proposes bilateral classifier branches that learn different patterns. Decoupled training [14] also identifies the issues with class re-balancing. It suggests decoupling representation learning and classifier learning into two stages, applying class re-balancing only during classifier fine-tuning. However, fine-tuning the global model's classifier in FL is challenging because training data is distributed across different clients, and clients are unable to share their local samples. Information augmentation aims to enhance model performance by introducing additional information through methods such as transfer learning and data augmentation.

2.2 Federated Learning

Federated learning (FL) is a privacy-preserving machine learning algorithm where clients collaboratively train a global model without sharing private data. In a typical FL algorithm, such as FedAvg [23], clients send their locally trained model parameters to the server, which averages them to update the global model. When data is distributed in a non-IID manner, FL faces challenges such as slow convergence and degraded performance. To address these issues, various non-IID-oriented FL methods have been proposed, such as FedProx [19] and FedDecorr [29]. However, they do not provide targeted improvements for the class imbalance issue in Fed-LT. Some approaches address both the non-IID and class imbalance in Fed-LT by enhancing local learning: FedLoGe [36] proposes a sparse Equiangular Tight Frame classifier to suppress noisy features in Fed-LT; FedLC [40] and FedRoD [3] replace cross-entropy loss with their improved loss functions. FLea [35] addresses the issues of label skew and missing classes by sharing intermediate layer features, which are used for mixup during local training. Some approaches focus on directly adjusting the global model's classifier. FedNCM [17] and Fed3R [10] replace softmax classifiers with closed-form solution classifiers. CCVR and CReFF synthesize features on the server to fine-tune the global classifier, with CCVR assuming Gaussian distributions for each class's features and CReFF optimizing a gradient matching loss. However, it is vital to tackle both the global and local imbalance at the same time in Fed-LT.

3 Preliminaries

3.1 Problem Statement of FL

Assume that there are K clients and a server in the FL system. Each client k holds a private local dataset $\mathcal{D}_k$. The goal of FL is to train a global model θ on $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ without transferring clients' private dataset. The overall objective of FL is defined as

$$\min_{\theta} \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbb{E}_{(x,y)\sim\mathcal{D}_k} [\mathcal{L}(x, y; \theta)]. \tag{1}$$

3.2 Maximum Mean Discrepancy and Random Feature Approximation

Maximum Mean Discrepancy (MMD) [11] is a measure of discrepancy between probability distributions. Given two datasets $\boldsymbol{U} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{V} \in \mathbb{R}^{m\times d}$, the empirical squared MMD with kernel function $k(\cdot,\cdot)$ is

$$\begin{aligned} \mathrm{MMD}^2(\boldsymbol{U},\boldsymbol{V}) &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} k(\boldsymbol{u}_i,\boldsymbol{u}_{i'}) \\ &+ \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m} k(\boldsymbol{v}_j,\boldsymbol{v}_{j'}) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(\boldsymbol{u}_i,\boldsymbol{v}_j). \end{aligned} \tag{2}$$

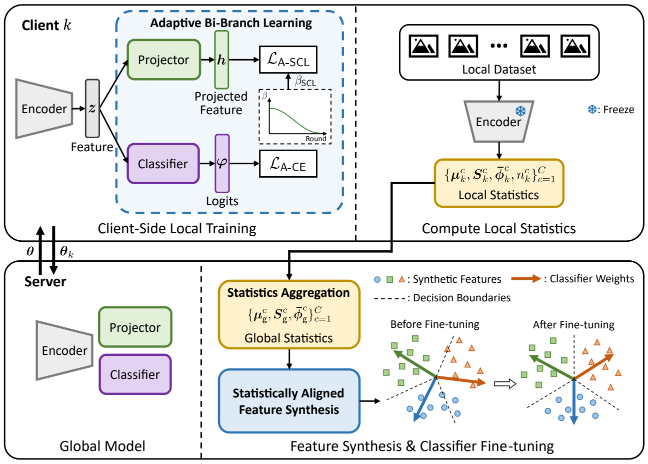

Client k
Adaptive Bi-Branch Learning
Encoder
z
Feature
Projector
h
Projected Feature
$\mathcal{L}_{\text{A-SCL}}$
β_{SCL}
β
Round
Classifier
φ
Logits
$\mathcal{L}_{\text{A-CE}}$
Client-Side Local Training
Local Dataset
Encoder
❄: Freeze
$\{\boldsymbol{\mu}_k^c, \boldsymbol{S}_k^c, \bar{\boldsymbol{\phi}}_k^c, n_k^c\}_{c=1}^C$
Local Statistics
Compute Local Statistics
θ
θ_k
Server
Encoder
Projector
Classifier
Global Model
Statistics Aggregation
$\{\boldsymbol{\mu}_g^c, \boldsymbol{S}_g^c, \bar{\boldsymbol{\phi}}_g^c\}_{c=1}^C$
Global Statistics
Statistically Aligned Feature Synthesis
● ■ ▲: Synthetic Features
→ : Classifier Weights
- - - -: Decision Boundaries
Before Fine-tuning
After Fine-tuning
Feature Synthesis & Classifier Fine-tuning

Figure 2: Overview of Synthetic Feature-based Decoupled training (SFD) method.

The random feature method [25] defines a mapping function $\tilde{\phi}(\cdot)$ for kernel approximation, such that $k(\boldsymbol{u},\boldsymbol{v}) \approx \langle\tilde{\phi}(\boldsymbol{u}), \tilde{\phi}(\boldsymbol{v})\rangle$. The formulation of $\tilde{\phi}(\cdot)$ is

$$\begin{aligned} \tilde{\phi}(z) = \sqrt{\frac{2}{D}}\Big[&\sin(\boldsymbol{\omega}_1^{\mathrm{T}} z),\ \cos(\boldsymbol{\omega}_1^{\mathrm{T}} z),\ \ldots, \\ &\sin(\boldsymbol{\omega}_{D/2}^{\mathrm{T}} z),\ \cos(\boldsymbol{\omega}_{D/2}^{\mathrm{T}} z)\Big]^{\mathrm{T}}, \end{aligned} \tag{3}$$

where $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{D/2}$ are randomly constructed with several possible implementations [25, 26, 37] and D is the dimension of the random feature. By replacing $k(\cdot,\cdot)$ with $\langle\tilde{\phi}(\cdot), \tilde{\phi}(\cdot)\rangle$ in Eq. (2), the following approximation can be obtained:

$$\mathrm{MMD}^2(\boldsymbol{U},\boldsymbol{V}) \approx \left\| \frac{1}{n}\sum_{i=1}^{n} \tilde{\phi}(\boldsymbol{u}_i) - \frac{1}{m}\sum_{j=1}^{m} \tilde{\phi}(\boldsymbol{v}_j) \right\|_2^2. \tag{4}$$

4 Methodology

Figure 2 provides an overview of the proposed SFD method. First, the model is trained for R rounds by exchanging parameters between clients and the server, and the global model is obtained via parameter averaging. During local training, we adopt Adaptive Bi-Branch Learning (ABBL) to learn high-quality representations and clearer decision boundaries from long-tailed data. Next, after the last round, we aggregate local statistics to obtain global statistics. These are then used in the proposed Statistically Aligned Feature Synthesis (SAFS) to construct synthetic features on the server. The synthetic features are employed to fine-tune the global classifier, which mitigates the bias caused by non-IID long-tailed data, and further improves recognition performance.

4.1 Adaptive Bi-Branch Learning

We propose Adaptive Bi-Branch Learning (ABBL), which adopts a bi-branch structure [34, 42] to jointly enhance feature representation and decision boundary learning in Fed-LT. ABBL leverages distribution-aware objectives in two complementary branches. The classification branch adaptively adjusts the local training objective based on the class distribution and the presence of missing classes at each client, thereby improving decision boundaries for more accurate classification. Meanwhile, the contrastive branch applies an adaptive supervised contrastive loss that adjusts gradients from negative samples to refine representations. Through these adaptive adjustments, ABBL encourages well-separated features and clearer decision boundaries.

4.1.1 Classification Branch. As shown in Fig. 2, the classification branch uses a classifier to predict the sample's class based on the feature extracted by the encoder. This branch is optimized by the proposed Adaptive Cross-Entropy loss:

$$\mathcal{L}_{\text{A-CE}}(x,y) = -\log \frac{\exp(\varphi_y(x) + \gamma \log \pi_k^y)}{\sum_{c=1}^{C} \exp(\varphi_c(x) + \gamma \log \pi_k^c)}, \tag{5}$$

where φ_c indicates the logit of class c, and γ is the hyperparameter for Logit Adjustment (LA) [24]. $\mathcal{L}_{\text{A-CE}}$ adaptively adjusts the local training objective based on the class distribution and the presence of missing classes at each client. Compared with the original LA loss, $\mathcal{L}_{\text{A-CE}}$ explicitly accounts for missing classes in FL clients by using π_k^c instead of n_k^c to adjust the logits:

$$\pi_k^c = \begin{cases} \beta_\pi n_k^+ & \text{if } n_k^c = 0 \\ n_k^c & \text{if } n_k^c > 0 \end{cases}, \text{ where } \begin{array}{c} n_k^+ = \min\limits_{n_k^j > 0} n_k^j, \\ \beta_\pi \in [0,1], \end{array} \tag{6}$$

n_k^c is the number of samples for class c on client k, n_k^+ is the minimum number of samples among non-missing classes on that client, and β_π is a hyperparameter. Logit adjustment is a practical long-tailed learning technique, but it is designed for centralized learning and does not consider that local data on FL clients may have missing classes. As a result, the original LA loss leads to $\frac{\partial \mathcal{L}_{\text{LA}}}{\partial \phi_c} = 0$ for any missing class c on the client. The absence of such a repulsive gradient $\frac{\partial \mathcal{L}_{\text{LA}}}{\partial \phi_c}$ may lead to inaccurate decision boundaries and representations. With the improvement of π_k^c in Eq. (6), any missing class c now has $\frac{\partial \mathcal{L}_{\text{A-CE}}}{\partial \phi_c} > 0$ when $\beta_\pi > 0$, providing an appropriate repulsive gradient to correct decision boundaries and representations. The detailed derivation can be found in Appendix A.1.

4.1.2 Contrastive Branch. As illustrated in Fig. 2, a projector is added after the encoder and serves as the contrastive branch. Supervised Contrastive Learning (SCL) [15] is an effective technique to enhance the quality of representations. Motivated by this, we adopt the contrastive branch to enhance the representation learning in Fed-LT. This branch is optimize by the proposed Adaptive Supervised Contrastive Learning loss, which adjusts the gradients from negative samples based on the class distribution. The loss is expressed as follows:

$$\mathcal{L}_{\text{A-SCL}}(x_i, y_i) = -\frac{1}{|A(i)|} \sum_{a \in A(i)} \log \frac{\exp(s_{i,a})}{\sum\limits_{b \neq i} \exp(s_{i,b} + \Delta_{y_b})}, \tag{7}$$
$$\text{where } s_{i,j} = \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle / \tau, \ \Delta_{y_b} = \log n_k^{y_b},$$

$A(i) = \{a \mid y_a = y_i\}$ denotes the indices of samples that belong to the same class as x_i, $\boldsymbol{h}_i$ is the projected feature of x_i, τ is the temperature in contrastive learning. Compared with standard SCL loss, the Δ_{y_b} in $\mathcal{L}_{\text{A-SCL}}$ increases the repulsive gradient from head-class negative samples. Specifically, when x_i is a tail-class sample, the loss encourages the separation between tail-class and head-class features, helping the model better distinguish tail-class samples from head-class samples. When x_i is a head-class sample, the loss increases the intra-class repulsive gradient, preventing intra-class representation collapse. According to previous study [27], intra-class representation collapse makes SCL less effective in FL, and maintaining a certain intra-class diversity can improve performance and enhance feature transferability across clients.

The total loss for local training is defined as:

$$\mathcal{L}_{\text{ABBL}} = \mathcal{L}_{\text{A-CE}} + \beta_{\text{SCL}} \mathcal{L}_{\text{A-SCL}}, \tag{8}$$

where β_{SCL} decays according to a cosine annealing strategy as the communication rounds progress. This process first enhances feature representations, and then refines the classifier to produce more accurate decision boundaries. More details about the algorithm can be found in Appendix B.

4.2 Statistics Aggregation

After R rounds of communication and training, we obtain a trained global model θ. After the last round, the server aggregates the local statistics from each client to obtain global statistics, as illustrated in in Fig. 2. First, the global encoder are sent to the clients. Then, the clients freeze the encoder, use it to extract features, and compute local statistics $\{\boldsymbol{\mu}_k^c, S_k^c, \bar{\boldsymbol{\phi}}_k^c, n_k^c\}_{c=1}^C$:

$$\begin{aligned} \boldsymbol{\mu}_k^c &= \frac{1}{n_k^c} \sum_{z \in \mathcal{Z}_k^c} z, \qquad & S_k^c &= \frac{1}{n_k^c} \sum_{z \in \mathcal{Z}_k^c} z z^{\mathrm{T}}, \\ \bar{\boldsymbol{\phi}}_k^c &= \frac{1}{n_k^c} \sum_{z \in \mathcal{Z}_k^c} \tilde{\phi}(z), \quad & n_k^c &= |\mathcal{Z}_k^c|, \end{aligned} \tag{9}$$

where C is the number of classes, z indicates the feature extracted by the encoder, and $\mathcal{Z}_k^c = \{z_i = \text{Encoder}(\boldsymbol{x}_i) \mid y_i = c\}$ indicates the extracted features of class c samples in client k.

Finally, local statistics from clients are aggregated to obtain the global statistics $\{\boldsymbol{\mu}_{\text{g}}^c, \Sigma_{\text{g}}^c, \bar{\boldsymbol{\phi}}_{\text{g}}^c\}_{c=1}^C$:

$$\begin{aligned} \boldsymbol{\mu}_{\text{g}}^c &= \frac{1}{N_c} \sum_{k=1}^{K} n_k^c \boldsymbol{\mu}_k^c, \qquad \bar{\boldsymbol{\phi}}_{\text{g}}^c = \frac{1}{N_c} \sum_{k=1}^{K} n_k^c \bar{\boldsymbol{\phi}}_k^c, \\ \Sigma_{\text{g}}^c &= \frac{1}{N_c} \sum_{k=1}^{K} n_k^c S_k^c - \boldsymbol{\mu}_{\text{g}}^c \left(\boldsymbol{\mu}_{\text{g}}^c\right)^{\mathrm{T}}, \end{aligned} \tag{10}$$

where $N_c = \sum_{k=1}^{K} n_k^c$. Notably, since Eq. (10) consists of summation and averaging, a secure aggregation protocol [1] can be applied so

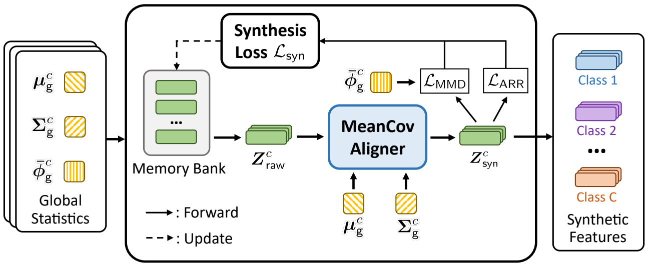


Figure 3: Overview of Statistically Aligned Feature Synthesis (SAFS).

Algorithm 1 Statistically Aligned Feature Synthesis (SAFS)

```
Input: Number of classes $C$, number of iterations $T$.
Output: Synthetic feature dataset $\mathcal{Z}_{\text{syn}}$.
$\mathcal{Z}_{\text{syn}} \leftarrow \{\}$
for each class $c = 1$ to $C$ do
    Randomly initialize a memory bank with size $M_c$
    for each iteration $t = 1$ to $T$ do
        $Z_{\text{raw}}^c \leftarrow$ A batch from memory bank
        $Z_{\text{syn}}^c \leftarrow$ Transform $Z_{\text{raw}}^c$ according to Eq. (14)
        Compute $\mathcal{L}_{\text{syn}}$ as defined by Eq. (17)
        Backward $\mathcal{L}_{\text{syn}}$ and update the memory bank
    end for
    $Z_{\text{raw}}^c \leftarrow$ Take out all $z_{\text{syn}}^c$ from the memory bank
    $Z_{\text{syn}}^c \leftarrow$ Transform $Z_{\text{raw}}^c$ according to Eq. (14)
    for each $z_{\text{syn}}^c$ in $Z_{\text{syn}}^c$ do
        Insert $(z_{\text{syn}}^c, c)$ into $\mathcal{Z}_{\text{syn}}$
    end for
end for
```

that the server can access only the aggregated global statistics, without revealing any individual client's local statistics. The algorithm of statistics aggregation is shown in Algorithm 2 in Appendix B.

4.3 Statistically Aligned Feature Synthesis

To reduce the model bias caused by non-IID long-tailed data, the server fine-tunes the global classifier using synthetic features, without accessing local samples and features from the clients. We propose Statistically Aligned Feature Synthesis (SAFS), which utilizes the aggregated global statistics in Eq. (10) to construct synthetic features that approximate the global feature distribution. SAFS runs only on the server, with no extra computational cost to clients. As shown in Fig. 3 and Algorithm 1, SAFS starts from a randomly initialized memory bank for each class, and the synthetic features are constructed by applying MeanCov Aligner's transformation and minimizing the feature synthesis loss.

4.3.1 MeanCov Aligner. We propose a MeanCov Aligner module, which aligns the mean and covariance of synthetic features with the global feature distribution, such that $\text{mean}(Z_{\text{syn}}^c) = \boldsymbol{\mu}_g^c$ and $\text{cov}(Z_{\text{syn}}^c) = \Sigma_g^c$. The unaligned synthetic features Z_{raw}^c, stored in a memory bank, are transformed into aligned synthetic features Z_{syn}^c via the MeanCov Aligner. Each row z_{syn}^c in the Z_{syn}^c indicates a synthetic feature sample, which is then used for subsequent classifier fine-tuning. The MeanCov Aligner operates four steps as follows.

Firstly, perform Cholesky decomposition on the global covariance Σ_g^c to obtain $\boldsymbol{L}_g$:

$$\Sigma_g^c = \boldsymbol{L}_g \boldsymbol{L}_g^{\mathrm{T}}. \tag{11}$$

Secondly, compute the mean $\boldsymbol{\mu}_{\text{raw}}^c$ and covariance Σ_{raw}^c of Z_{raw}^c, then perform Cholesky decomposition on Σ_{raw}^c to obtain $\boldsymbol{L}_{\text{raw}}$:

$$\Sigma_{\text{raw}}^c = \boldsymbol{L}_{\text{raw}} \boldsymbol{L}_{\text{raw}}^{\mathrm{T}}. \tag{12}$$

Thirdly, compute the transformation matrix $\boldsymbol{A}$ [1]:

$$\boldsymbol{A} = \boldsymbol{L}_g \boldsymbol{L}_{\text{raw}}^{-1}. \tag{13}$$

Lastly, apply transformation on Z_{raw}^c to obtain Z_{syn}^c:

$$Z_{\text{syn}}^c = \left(Z_{\text{raw}}^c - \mathbf{1} \cdot \boldsymbol{\mu}_{\text{raw}}^{c\,\mathrm{T}} \right) \boldsymbol{A}^{\mathrm{T}} + \mathbf{1} \cdot \boldsymbol{\mu}_g^{c\,\mathrm{T}}. \tag{14}$$

After the transformation, the mean and covariance of the synthetic features are aligned with the global feature distribution. The detailed derivation can be found in Appendix A.2.

4.3.2 Optimize Approximate MMD. Previous studies have explored generating data by optimizing MMD between real and generated data [9, 12, 18]. Inspired by these works, we refine the alignment between the synthetic and global feature distribution by minimizing their MMD. MeanCov Aligner focuses on aligning first and second-order statistics (mean and covariance), while RBF kernel MMD complements it by capturing higher-order distributional differences. However, the MMD defined in Eq. (2) cannot be directly computed in FL settings, as it requires pairwise kernel computations between samples. FL's privacy-preserving constraints prevent clients' local samples from being shared with the server or other clients. To address this, we utilize the random feature method [25] to achieve an MMD approximation under privacy-preserving constraints. Derived from Eq. (4), the approximate MMD loss for feature synthesis is defined as

$$\mathcal{L}_{\text{MMD}} = \left\| \bar{\boldsymbol{\phi}}_g^c - \frac{1}{M_c} \sum_{j=1}^{M_c} \tilde{\phi}\left(z_{\text{syn},j}^c \right) \right\|_1, \tag{15}$$

where M_c is the number of synthetic features, $\bar{\boldsymbol{\phi}}_g^c$ is one of the aggregated global statistics in Eq. (10), and $\frac{1}{M_c} \sum_{j=1}^{M_c} \tilde{\phi}\left(z_{\text{syn},j}^c \right)$ can be directly computed on the server. Unlike the commonly used L2-based MMD, we use an L1 form in our $\mathcal{L}_{\text{MMD}}$, which converges faster with the combination of learning rate annealing. This may be because, when the discrepancy between the two distributions is small, the L2 form results in vanishing gradients, whereas the L1 form maintains sufficient gradients for continued optimization.

[1] Note that $\boldsymbol{L}_{\text{raw}}$ in Eq. (13) is a lower triangular matrix, its inverse $\boldsymbol{L}_{\text{raw}}^{-1}$ can be efficiently computed using the forward substitution algorithm.

4.3.3 Activation-aware Range Regularizer. The encoder outputs features after applying an activation function. Taking the most commonly used ReLU activation as an example, ReLU is a function with a range of $[0, +\infty]$. Since the features extracted by the encoder pass through ReLU, the values of the real features are non-negative. To construct synthetic features with a reasonable value range, we propose the Activation-aware Range Regularizer (ARR):

$$\mathcal{L}_{\text{ARR}}\left(z^c_{\text{syn}}\right) = -\sum_{j=1}^{d} \min\left(0, z^c_{\text{syn},j}\right), \tag{16}$$

where d is the dimension of feature. $\mathcal{L}_{\text{ARR}}$ penalizes negative values in the synthetic features z^c_{syn}, encouraging the values to be non-negative.

The overall feature synthesis loss is

$$\mathcal{L}_{\text{syn}} = \mathcal{L}_{\text{MMD}} + \mathcal{L}_{\text{ARR}}. \tag{17}$$

Since all operations in the MeanCov Aligner are differentiable, the gradient of $\mathcal{L}_{\text{syn}}$ can be backpropagated to the memory bank, enabling the Z^c_{raw} in the memory bank to be updated via gradient descent. After constructing the synthetic features for each class, we use them to fine-tune the global model's classifier with cross-entropy loss. This classifier fine-tuning reduces the bias of the global model, thereby improving its performance.

5 Experiment

5.1 Experimental Setups

5.1.1 Datasets. We conduct experiments on three long-tailed datasets: CIFAR-10-LT, CIFAR-100-LT, and CINIC-10-LT, which are derived from commonly used image classification benchmarks in FL research [22, 28, 36]. CIFAR-10 [16] consists of 32×32 color images in 10 classes. CIFAR-100 [16] includes 100 classes images. CINIC-10 [6] is a larger dataset that incorporates a subset of ImageNet [7] and the entire CIFAR-10.

Following the common setup in long-tailed learning studies [2, 28, 36], we construct long-tailed training sets using exponential distributions. The degree of imbalance is controlled by the imbalance factor (IF), which is defined as the ratio of samples in the most frequent class to those in the least frequent class. A larger IF indicates a more severe class imbalance. For the test set, we keep the original class-balanced test set to ensure equal treatment of each class during evaluation. To simulate the non-IID data across clients in FL, we follow the common setup [38] in FL by distributing the training set across K clients according to Dirichlet distributions. The Dirichlet distribution is parameterized by α, with a smaller α indicating a higher degree of non-IID.

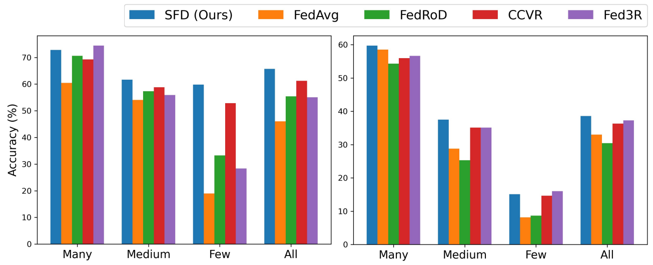
5.1.2 Comparison Methods. To evaluate the effectiveness of the proposed SFD method, we compare it against several baseline methods. The selected methods include both classic FL methods and recently proposed state-of-the-art methods for Fed-LT. The comparison methods are divided into three groups:

(1) Addressing general non-IID issues, including FedAvg [23], FedProx [19], and FedDecorr [29];
(2) Tackling class imbalance in local training, including FedLC [40], FedRoD [3], FLea [35], and FedLoGe [36];
(3) Adjusting the global classifier, including CCVR [22], CReFF [28], FedNCM [17], and Fed3R [10].

5.1.3 Implementation Details. In all experiments, we use ResNet-18 [13] as the encoder and its fully connected (FC) layer as the classifier. We follow the common practice in contrastive learning [4] by using a 2-layer MLP as the projector and L2-normalizing the projected features. We evaluate our method and comparison methods under different degrees of non-IID and class imbalance settings. The non-IID degree α is set to $\{0.05, 0.2\}$ for CIFAR and 0.2 for CINIC. The imbalance factor is set to $\{100, 50\}$. We use $K = 10$ clients with full participation, $R = 200$ communication rounds, and $E = 5$ local epochs. We run experiments for 3 times and report the average results. For local training, we use SGD optimizer with a momentum of 0.9 and a weight decay of 10^{-5} to optimize the local models. The learning rate is set to 0.01 and the batch size is set to 64. For our SFD method, we tune γ from $\{0.1, 1\}$ and β_π from $\{0, 1\}$. We recommend $\gamma = 0.1$ and $\beta_\pi = 1$ as they perform well in most cases. The temperature τ for contrastive learning is set to 0.07. More implementation details can be found in Appendix C.2.

5.2 Main Results

Table 1 presents the main experimental results. We observe that the group 2, class imbalance oriented method, generally outperforms group 1, general non-IID oriented method. This suggests that targeted improvements for long-tailed data can effectively boost performance in Fed-LT. Additionally, we find that the group 3, which focuses on adjusting the global classifier, performs better than group 1 and group 2 in most cases. This supports the view [14, 22] that the classifier layer has the most detrimental model bias, therefore adjusting this layer alone can yield substantial performance gains. Our SFD method outperforms comparison methods across various experimental settings. This advantage is due to improvements in both representation learning and classifier fine-tuning. SFD enables the model to learn a high-quality encoder and effectively adjust the classifier of the global model.



(a) CIFAR-10-LT (b) CIFAR-100-LT

Figure 4: Test accuracies of SFD and comparison methods across different class groups.

To demonstrate the performance across head and tail classes under a long-tailed data distribution, we follow a common practice in long-tailed learning research: we categorize the classes into three groups—Many-shot, Medium-shot, and Few-shot—based on the number of training samples, then calculate the accuracy for each group [21]. The detailed categorization method is provided in

Table 1: Test accuracies of SFD and comparison methods on CIFAR-10-LT, CIFAR-100-LT and CINIC-10-LT with diverse non-IID and class imbalance settings.

Dataset	CIFAR-10-LT				CIFAR-100-LT				CINIC-10-LT	
Non-IID	$\alpha = 0.05$		$\alpha = 0.2$		$\alpha = 0.05$		$\alpha = 0.2$		$\alpha = 0.2$	
Imbalance	IF = 100	IF = 50	IF = 100	IF = 50	IF = 100	IF = 50	IF = 100	IF = 50	IF = 100	IF = 50
FedAvg	46.08	53.80	61.74	66.34	33.01	38.73	34.56	38.63	51.44	54.18
FedProx	46.91	54.31	61.83	67.59	33.94	38.64	34.20	38.82	52.04	55.00
FedDecorr	46.65	55.16	59.45	66.89	31.40	35.03	31.28	36.02	49.91	55.31
FedLC	48.31	58.16	60.32	68.45	24.57	34.42	33.10	38.55	54.44	57.18
FedRoD	54.34	60.01	67.24	74.77	30.46	37.31	35.07	40.95	58.85	61.09
FLea	48.13	66.22	58.33	67.37	33.93	37.64	34.26	38.57	47.38	54.02
FedLoGe	50.10	52.80	66.72	67.68	34.25	40.14	36.84	41.64	51.83	58.36
CCVR	62.55	73.74	70.39	74.87	36.29	42.15	37.06	41.05	59.45	61.03
CReFF	53.21	70.45	68.93	72.04	33.05	39.49	35.89	39.27	57.08	60.38
FedNCM	55.75	66.87	66.62	70.19	35.66	41.41	35.68	39.72	57.71	59.67
Fed3R	55.06	67.50	65.33	70.05	37.29	42.47	37.46	40.75	53.77	57.50
SFD (Ours)	**65.69**	**73.81**	**70.69**	**77.07**	**38.58**	**43.72**	**39.79**	**44.61**	**60.07**	**61.39**

Appendix C.1. Fig. 4 presents the test accuracies of SFD and four representative baseline methods (FedAvg, FedRoD, CCVR, Fed3R) across the groups. The "All" label in the figure represents the overall accuracy for all classes. The global models used in the evaluation are trained under IF = 100 and $\alpha = 0.05$ setting. From the figure, we can observe that, compared with the baseline methods, the proposed SFD method achieves significant performance improvement on the Few-shot classes. Additionally, SFD also improves the recognition performance on the Many-shot and Medium-shot classes, indicating that SFD not only enhances the performance on tail classes but also effectively learns head classes. Overall, by improving the recognition performance on both head and tail categories, the SFD method achieves a significant improvement in the overall accuracy.

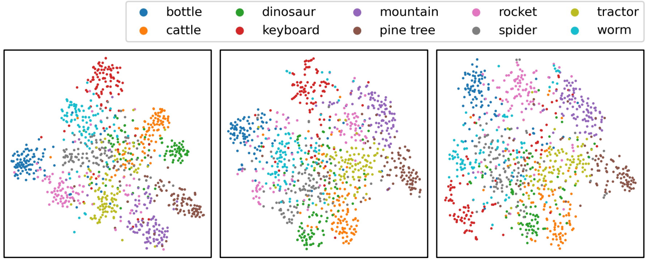
5.3 Ablation Studies

5.3.1 Ablation Study on Adaptive Bi-Branch Learning (ABBL). Table 2 shows the results of the ablation study on different components within ABBL. We conduct the ablation study on CIFAR-10-LT and CIFAR-100-LT with $\alpha = 0.05$ and IF = 100. Method variants in Table 2 modify only the ABBL in local training process, while keeping the feature synthesis and classifier fine-tuning process unchanged. This enables us to validate the improvements our method brings to local training. SFD-CE and SFD-LA are variants that do not make use of ABBL. SFD-CE uses cross-entropy loss for local training, while SFD-LA uses logit adjustment loss. SFD-LA outperforms on CIFAR-10-LT, whereas SFD-CE shows better performance on CIFAR-100-LT. This suggests that although logit adjustment can alleviate model bias in Fed-LT, it may sometimes do harm to representation learning. "w/o improved π_k^c" and "w/o Δ_{y_b} in $\mathcal{L}_{\text{A-SCL}}$" are variants that modify ABBL. In the case of "w/o improved π_k^c", we replace the loss of classification branch from $\mathcal{L}_{\text{A-CE}}$ to the original logit adjustment loss. For "w/o Δ_{y_b} in $\mathcal{L}_{\text{A-SCL}}$", we replace the loss of contrastive branch from $\mathcal{L}_{\text{A-CE}}$ to the original SCL loss. Both modifications in these variants result in a performance decline, indicating that the improvements we proposed for classification and contrastive losses are beneficial for local training in Fed-LT.

Table 2: Ablation study on different components within Adaptive Bi-Branch Learning (ABBL).

Method Variants	CIFAR-10-LT	CIFAR-100-LT
SFD (Full Method)	65.69	38.58
SFD-CE	62.73 (↓ 2.96)	36.92 (↓ 1.66)
SFD-LA	63.46 (↓ 2.23)	36.29 (↓ 2.29)
w/o improved π_k^c	65.69 (- 0.00)	38.03 (↓ 0.55)
w/o Δ_{y_b} in $\mathcal{L}_{\text{A-SCL}}$	64.53 (↓ 1.16)	38.48 (↓ 0.10)

The performance of both SFD-CE and SFD-LA is inferior to that of the full method using ABBL, indicating that ABBL improves the encoder's representation capability, enabling it to extract higher-quality representations. This, in turn, helps the model achieve better recognition performance after classifier fine-tuning. We further validate this view through feature visualization. We train models on CIFAR-100-LT (IF = 100, $\alpha = 0.05$) using ABBL, cross-entropy, and logit adjustment, respectively. Then, for visualization purposes, we evenly select 10 classes based on the number of training samples, from the most to the least frequent. We use t-SNE [31] to visualize the features of the selected classes extracted by encoders trained with different loss functions. As shown in Fig. 5, ABBL results in more compact features within each class and enhances the separability between features of different classes. In contrast, the features from CE and LA are more dispersed, with fuzzier boundaries between classes. This validates our view that ABBL enhances the quality of representations, helping SFD method achieve leading performance.



(a) ABBL (Ours) **(b) Cross-Entropy** **(c) Logit Adjustment**

Figure 5: T-SNE visualization of features extracted by encoders trained with ABBL, Cross-Entropy, and Logit Adjustment.

Table 3: Ablation studies on different components within Statistically Aligned Feature Synthesis (SAFS).

Method Variants	CIFAR-10-LT	CIFAR-100-LT
SFD (Full Method)	65.69	38.58
w/o SAFS & Fine-tune	51.42 (↓14.27)	34.76 (↓ 3.82)
w/o MeanCov Aligner	57.74 (↓ 7.95)	35.89 (↓ 2.69)
w/o $\mathcal{L}_{\text{MMD}}$	65.52 (↓ 0.17)	38.49 (↓ 0.09)
w/o $\mathcal{L}_{\text{ARR}}$	65.56 (↓ 0.13)	38.42 (↓ 0.16)

5.3.2 Ablation Study on Statistically Aligned Feature Synthesis (SAFS). Table 3 shows the results of the ablation study on different components within SAFS. We conduct the ablation study on CIFAR-10-LT and CIFAR-100-LT with $\alpha = 0.05$ and IF = 100. "w/o SAFS & Fine-tune" means training with ABBL only, without using SAFS to fine-tune the global model's classifier. The results show a significant performance drop without classifier fine-tuning, indicating that reducing model bias only at client-side local training is insufficient. SAFS directly addresses the bias of global model by fine-tuning its classifier, which directly enhances the global model's performance. "w/o MeanCov Aligner" means that features are synthesized solely by optimizing the feature synthesis loss. The results show that this variant outperforms the one without classifier fine-tuning, but its performance still falls short of the full method. This suggests that MeanCov Aligner, by explicitly aligning the mean and covariance through transformations, addresses the inefficiency of aligning feature statistics solely through loss function optimization. This leads to more efficient alignment of the feature distribution, thereby further improving the performance of the global model. We also attempt removing the feature synthesis loss (L_{MMD} and L_{ARR}), and the results show a performance drop in each case. This indicates that L_{MMD} and L_{ARR} further refines the alignment between the synthetic and the real global feature distribution, thus further improves the performance.

5.4 Further Studies

5.4.1 Different numbers of clients. We evaluated the performance of SFD under different numbers of clients. The experiments are conducted on CIFAR-10-LT and CIFAR-100-LT with IF = 100 and $\alpha = 0.2$, with the number of clients K set to $\{10, 20, 50, 100\}$. For $K = 50$ and $K = 100$, we randomly select 50% and 25% of the clients to participate in training during each round, respectively. We use a batch size of 32 for both $K = 50$ and $K = 100$. For CCVR, Fed3R, and SFD, global statistics are computed using the results from all clients after the last round of training. As shown in Table 4, SFD consistently demonstrates good performance across different numbers of clients, further validating its applicability and effectiveness in various federated learning scenarios.

Table 4: Test accuracies of our SFD method and comparison methods with varying number of clients (K).

Dataset	Method	$K = 10$	$K = 20$	$K = 50$	$K = 100$
CIFAR-10-LT	FedAvg	61.74	61.02	57.15	50.34
	FedRoD	67.24	69.59	69.84	58.56
	CCVR	70.39	72.32	70.66	63.52
	Fed3R	65.33	64.87	63.20	55.86
	SFD (Ours)	**70.69**	**72.35**	**71.51**	**63.79**
CIFAR-100-LT	FedAvg	34.56	31.69	30.62	27.70
	FedRoD	35.07	32.67	31.83	28.98
	CCVR	37.06	34.53	33.94	32.20
	Fed3R	37.46	34.40	33.49	31.22
	SFD (Ours)	**39.79**	**36.60**	**35.45**	**32.47**

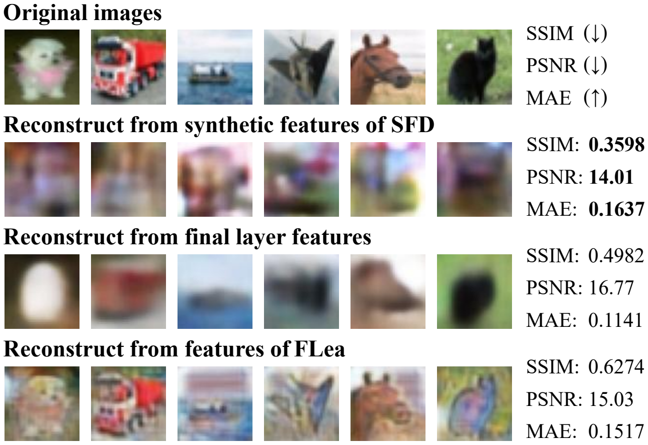


Figure 6: Left: examples of reconstructed images obtained by inverting different features. Right: reconstruction metrics on the entire dataset, where (↑) means better defense with higher values, and (↓) means better defense with lower values.

5.4.2 Preventing data reconstruction attack. To evaluate the effectiveness of SFD against data reconstruction attacks, we conduct image reconstruction experiments based on the approach in [35]. We recover the original images by inverting the synthetic features of SFD, the real final layer features, and the intermediate layer features shared in FLea [35], as described in [8, 32, 35]. The results of the image reconstruction attack are shown in Fig. 6. We use image similarity metrics such as SSIM, PSNR, and MAE to evaluate the resistance to image reconstruction attacks, as done in [30], where

(↑) means better defense with higher values, and (↓) means better defense with lower values. Since there is no one-to-one mapping between the synthetic features of SFD and the original image, we select the closest reconstructed image as the corresponding one. As shown in Fig. 6, performing a reconstruction attack on the synthetic features of SFD only results in images with no meaningful content, while attacking the real features often recovers the shape, contours, and colors of the image. This suggests that directly sharing sample features poses a risk of privacy leakage. Although FLea attempts to defend against reconstruction attacks through distance correlation regularization, individual information from the original image still inevitably remains in the features. In contrast, SFD protects privacy by avoiding direct data sharing, and both the visual results and numerical metrics of image reconstruction confirm the effectiveness of our approach. While protecting protection, our method maintains strong long-tail recognition performance. The details of the image reconstruction attack experiment can be found in Appendix C.3.

6 Conclusion

We propose Synthetic Feature-based Decoupled training (SFD) method to tackle the challenges of federated long-tailed learning (Fed-LT). To improve local training, we introduce Adaptive Bi-Branch Learning (ABBL), which promotes high-quality feature representations and refined decision boundaries. To mitigate global model bias, we propose Statistically Aligned Feature Synthesis (SAFS), which constructs synthetic features that approximate the global feature distribution in a privacy-preserving manner. These synthetic features are used to fine-tune the global model's classifier, directly reducing the bias caused by non-IID long-tailed data. Experimental results show that SFD effectively addresses the challenges of Fed-LT and and leads to improved performance on benchmark datasets.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2022YFB4501500, 2022YFB4501504).

References

[1] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM, 1175–1191. doi:10.1145/3133956.3133982

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 1565–1576. https://proceedings.neurips.cc/paper/2019/hash/621461af90cadfdaf0e8d4cc25129f91-Abstract.html

[3] Hong-You Chen and Wei-Lun Chao. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=I1hQbx10Kxn

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 9268–9277. doi:10.1109/CVPR.2019.00949

[6] Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. 2018. CINIC-10 is not ImageNet or CIFAR-10. *CoRR* abs/1810.03505 (2018). arXiv:1810.03505 http://arxiv.org/abs/1810.03505

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. 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*. IEEE Computer Society, 248–255. doi:10.1109/CVPR.2009.5206848

[8] Alexey Dosovitskiy and Thomas Brox. 2016. Inverting Visual Representations with Convolutional Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 4829–4837. doi:10.1109/CVPR.2016.522

[9] Gintare Karolina Dziugaite, Daniel M. Roy, and Zoubin Ghahramani. 2015. Training generative neural networks via Maximum Mean Discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, Marina Meila and Tom Heskes (Eds.). AUAI Press, 258–267. http://auai.org/uai2015/proceedings/papers/230.pdf

[10] Eros Fanì, Raffaello Camoriano, Barbara Caputo, and Marco Ciccone. 2024. Accelerating Heterogeneous Federated Learning with Closed-form Classifiers. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. PMLR. https://openreview.net/forum?id=cMige5MK1N

[11] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. 2012. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* 13 (2012), 723–773. doi:10.5555/2503308.2188410

[12] Frederik Harder, Kamil Adamczewski, and Mijung Park. 2021. DP-MERF: Differentially Private Mean Embeddings with RandomFeatures for Practical Privacy-preserving Data Generation. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, 1819–1827. http://proceedings.mlr.press/v130/harder21a.html

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. doi:10.1109/CVPR.2016.90

[14] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling Representation and Classifier for Long-Tailed Recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=r1gRTCVFvB

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

[16] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. *Learning Multiple Layers of Features from Tiny Images.* Technical Report. Department of Computer Science, University of Toronto. http://www.cs.toronto.edu/~kriz/cifar.html

[17] Gwen Legate, Nicolas Bernier, Lucas Page-Caccia, Edouard Oyallon, and Eugene Belilovsky. 2023. Guiding The Last Layer in Federated Learning with Pre-Trained Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/dcc0ac74ac8b95dc1939804acce0317d-Abstract-Conference.html

[18] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017. MMD GAN: Towards Deeper Understanding of Moment Matching Network. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 2203–2213. https://proceedings.neurips.cc/paper/2017/hash/dfd7468ac613286cdbb40872c8ef3b06-Abstract.html

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of the Third Conference on Machine Learning and Systems, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze (Eds.). mlsys.org. https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2999–3007. doi:10.1109/ICCV.2017.324

[21] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. 2019. Large-Scale Long-Tailed Recognition in an Open World. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2537–2546. doi:10.1109/CVPR.2019.00264

[22] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 5972–5984. https://proceedings.neurips.cc/paper/2021/hash/2f2b265625d76a6704b08093c652fd79-Abstract.html

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[24] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. Long-tail learning via logit adjustment. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=37nvvqkCo5

[25] Ali Rahimi and Benjamin Recht. 2007. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis (Eds.). Curran Associates, Inc., 1177–1184. https://proceedings.neurips.cc/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html

[26] Isaac Reid, Krzysztof Marcin Choromanski, Valerii Likhosherstov, and Adrian Weller. 2023. Simplex Random Features. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 28864–28888. https://proceedings.mlr.press/v202/reid23a.html

[27] Seonguk Seo, Jinkyu Kim, Geeho Kim, and Bohyung Han. 2024. Relaxed Contrastive Learning for Federated Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, 12279–12288. doi:10.1109/CVPR52733.2024.01167

[28] Xinyi Shang, Yang Lu, Gang Huang, and Hanzi Wang. 2022. Federated Learning on Heterogeneous and Long-Tailed Data via Classifier Re-Training with Federated Features. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 2218–2224. doi:10.24963/IJCAI.2022/308

[29] Yujun Shi, Jian Liang, Wenqing Zhang, Vincent Y. F. Tan, and Song Bai. 2023. Towards Understanding and Mitigating Dimensional Collapse in Heterogeneous Federated Learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=EXnIyMVTL8s

[30] Abhishek Singh, Ayush Chopra, Ethan Garza, Emily Zhang, Praneeth Vepakomma, Vivek Sharma, and Ramesh Raskar. 2021. DISCO: Dynamic and Invariant Sensitive Channel Obfuscation for Deep Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 12125–12135. doi:10.1109/CVPR46437.2021.01195

[31] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[32] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. In *20th International Conference on Data Mining Workshops, ICDM Workshops 2020, Sorrento, Italy, November 17-20, 2020*, Giuseppe Di Fatta, Victor S. Sheng, Alfredo Cuzzocrea, Carlo Zaniolo, and Xindong Wu (Eds.). IEEE, 933–942. doi:10.1109/ICDMW51313.2020.00134

[33] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. 2021. Addressing Class Imbalance in Federated Learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 10165–10173. doi:10.1609/AAAI.V35I11.17219

[34] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. 2021. Contrastive Learning Based Hybrid Networks for Long-Tailed Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 943–952. doi:10.1109/CVPR46437.2021.00100

[35] Tong Xia, Abhirup Ghosh, Xinchi Qiu, and Cecilia Mascolo. 2024. FLea: Addressing Data Scarcity and Label Skew in Federated Learning via Privacy-preserving Feature Augmentation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, Ricardo Baeza-Yates and Francesco Bonchi (Eds.). ACM, 3484–3494. doi:10.1145/3637528.3671899

[36] Zikai Xiao, Zihan Chen, Liyinglan Liu, Yang Feng, Joey Tianyi Zhou, Jian Wu, Wanlu Liu, Howard Hao Yang, and Zuozhu Liu. 2024. FedLoGe: Joint Local and Generic Federated Learning under Long-tailed Data. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. https://openreview.net/forum?id=V3j5d0GQgH

[37] Felix X. Yu, Ananda Theertha Suresh, Krzysztof Marcin Choromanski, Daniel N. Holtmann-Rice, and Sanjiv Kumar. 2016. Orthogonal Random Features. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 1975–1983. https://proceedings.neurips.cc/paper/2016/hash/53adaf494dc89ef7196d73636eb2451b-Abstract.html

[38] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian Nonparametric Federated Learning of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7252–7261. http://proceedings.mlr.press/v97/yurochkin19a.html

[39] Jing Zhang, Chuanwen Li, Jianzgong Qi, and Jiayuan He. 2023. A Survey on Class Imbalance in Federated Learning. *CoRR* abs/2303.11673 (2023). doi:10.48550/ARXIV.2303.11673 arXiv:2303.11673

[40] Jie Zhang, Zhiqi Li, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, and Chao Wu. 2022. Federated Learning with Label Distribution Skew via Logits Calibration. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (Eds.). PMLR, 26311–26329. https://proceedings.mlr.press/v162/zhang22p.html

[41] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep Long-Tailed Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 9 (2023), 10795–10816. doi:10.1109/TPAMI.2023.3268118

[42] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. 2020. BBN: Bilateral-Branch Network With Cumulative Learning for Long-Tailed Visual Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 9716–9725. doi:10.1109/CVPR42600.2020.00974

A Theoretical Derivations

A.1 Gradient of the Logit Adjustment Loss

The logit adjustment [24] loss is defined as

$$\mathcal{L}_{\mathsf{LA}}(x, y) = -\log \frac{\exp\left(\varphi_y(x) + \gamma \log n_k^y\right)}{\sum_{c=1}^{C} \exp\left(\varphi_c(x) + \gamma \log n_k^c\right)}, \tag{18}$$

where $\varphi_c(x)$ indicates the logit of class c, and n_c^k indicates the number of samples for class c on client k. Here, we focus on the repulsive gradient, denoted as $\frac{\partial \mathcal{L}_{\mathsf{LA}}(x,y)}{\partial \varphi_j}$ $(j \neq y)$, i.e., the gradient of the loss function with respect to the logit of incorrect classes. This repulsive gradient suppresses the logit of incorrect classes, encouraging them to decrease, thereby lowering the risk of misclassification.

However, in FL, the local datasets of clients may lack samples from certain classes. If class j is a missing class in the local dataset of client k, i.e., $n_k^j = 0$, then for any sample x, the repulsive gradient from class j is always 0, as shown in Eq. (19). Without repulsive gradients from certain classes, training becomes less effective at correcting misclassifications, and suffers in learning representations and decision boundaries.

$$\begin{aligned}\frac{\partial \mathcal{L}_{\mathrm{LA}}(x,y)}{\partial \varphi_j} &= \frac{\exp\left(\varphi_j + \gamma \log n_k^j\right)}{\sum_{c=1}^{C} \exp\left(\varphi_c + \gamma \log n_k^j\right)} \\ &= \frac{\left(n_k^j\right)^{\gamma} \cdot \exp(\varphi_j)}{\sum_{c=1}^{C} \exp\left(\varphi_c + \gamma \log n_k^j\right)} \\ &= \frac{0 \cdot \exp(\varphi_j)}{\sum_{c=1}^{C} \exp\left(\varphi_c + \gamma \log n_k^j\right)} = 0.\end{aligned} \tag{19}$$

A.2 Alignment of the Mean and Covariance

The following proves that after the transformation via the MeanCov Aligner, from Eq. (11) to Eq. (14), the synthetic features $\boldsymbol{Z}_{\mathrm{syn}}^c$ satisfy $\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \boldsymbol{\mu}_{\mathrm{g}}^c$ and $\mathrm{cov}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \Sigma_{\mathrm{g}}^c$. For clarity, we define the notation: symbols such as $\boldsymbol{Z}_{\mathrm{syn}}^c$ represent feature matrices, where each row corresponds to a feature sample. The functions $\mathrm{mean}(\cdot)$ and $\mathrm{cov}(\cdot)$ represent the mean vector and covariance matrix, respectively, of the features.

First, we prove that $\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \boldsymbol{\mu}_{\mathrm{g}}^c$. Taking the mean on both sides of Eq. (14) gives:

$$\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \mathrm{mean}\left(\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) \boldsymbol{A}^{\mathrm{T}} + \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{g}}^c}^{\mathrm{T}}\right), \tag{20}$$

where $\mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{g}}^c}^{\mathrm{T}}$ represents a matrix where each row is $\boldsymbol{\mu}_{\mathrm{g}}^c$, so we have $\mathrm{mean}(\mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{g}}^c}^{\mathrm{T}}) = \boldsymbol{\mu}_{\mathrm{g}}^c$. Similarly, we can deduce that $\mathrm{mean}(\mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}) = \boldsymbol{\mu}_{\mathrm{raw}}^c$. Since the mean operation is linear, Eq. (20) can be split as:

$$\begin{aligned}\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) &= \mathrm{mean}\left(\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) \boldsymbol{A}^{\mathrm{T}}\right) + \mathrm{mean}(\mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{g}}^c}^{\mathrm{T}}) \\ &= \boldsymbol{A} \cdot \mathrm{mean}\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) + \boldsymbol{\mu}_{\mathrm{g}}^c,\end{aligned} \tag{21}$$

where we apply the linearity of the mean operator to separate $\mathrm{mean}\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right)$, yielding:

$$\begin{aligned}\mathrm{mean}\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) &= \mathrm{mean}\left(\boldsymbol{Z}_{\mathrm{raw}}^c\right) - \mathrm{mean}\left(\mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) \\ &= \boldsymbol{\mu}_{\mathrm{raw}}^c - \boldsymbol{\mu}_{\mathrm{raw}}^c = \mathbf{0}.\end{aligned} \tag{22}$$

Substituting Eq. (22) into Eq. (21) gives:

$$\begin{aligned}\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) &= \boldsymbol{A} \cdot \mathbf{0} + \boldsymbol{\mu}_{\mathrm{g}}^c \\ &= \mathbf{0} + \boldsymbol{\mu}_{\mathrm{g}}^c = \boldsymbol{\mu}_{\mathrm{g}}^c,\end{aligned} \tag{23}$$

completing the proof for the mean alignment.

Next, we prove that $\mathrm{cov}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \Sigma_{\mathrm{g}}^c$. Taking the covariance on both sides of Eq. (14), and using the definition of the covariance matrix $\Sigma = \left(\boldsymbol{Z} - \mathbf{1} \cdot \boldsymbol{\mu}^{\mathrm{T}}\right)^{\mathrm{T}} \left(\boldsymbol{Z} - \mathbf{1} \cdot \boldsymbol{\mu}^{\mathrm{T}}\right)$, along with the previous result $\mathrm{mean}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \boldsymbol{\mu}_{\mathrm{g}}^c$, we obtain:

$$\mathrm{cov}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \boldsymbol{A}\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right)^{\mathrm{T}} \left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) \boldsymbol{A}^{\mathrm{T}}. \tag{24}$$

Note that the middle terms are precisely the covariance matrix Σ_{raw}^c, i.e.,

$$\left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right)^{\mathrm{T}} \left(\boldsymbol{Z}_{\mathrm{raw}}^c - \mathbf{1} \cdot {\boldsymbol{\mu}_{\mathrm{raw}}^c}^{\mathrm{T}}\right) = \mathrm{cov}(\boldsymbol{Z}_{\mathrm{raw}}^c) = \Sigma_{\mathrm{raw}}^c, \tag{25}$$

so Eq. (24) simplifies to:

$$\mathrm{cov}(\boldsymbol{Z}_{\mathrm{syn}}^c) = \boldsymbol{A} \Sigma_{\mathrm{raw}}^c \boldsymbol{A}^{\mathrm{T}}. \tag{26}$$

Substituting $\boldsymbol{A} = \boldsymbol{L}_{\mathrm{g}} \boldsymbol{L}_{\mathrm{raw}}^{-1}$ and $\Sigma_{\mathrm{raw}}^c = \boldsymbol{L}_{\mathrm{raw}} \boldsymbol{L}_{\mathrm{raw}}^{\mathrm{T}}$, we get:

$$\begin{aligned}\mathrm{cov}(\boldsymbol{Z}_{\mathrm{syn}}^c) &= \boldsymbol{L}_{\mathrm{g}} \boldsymbol{L}_{\mathrm{raw}}^{-1} \boldsymbol{L}_{\mathrm{raw}} \boldsymbol{L}_{\mathrm{raw}}^{\mathrm{T}} \left(\boldsymbol{L}_{\mathrm{raw}}^{-1}\right)^{\mathrm{T}} \boldsymbol{L}_{\mathrm{g}}^{\mathrm{T}} \\ &= \boldsymbol{L}_{\mathrm{g}} \left(\boldsymbol{L}_{\mathrm{raw}}^{-1} \boldsymbol{L}_{\mathrm{raw}}\right) \left(\boldsymbol{L}_{\mathrm{raw}}^{-1} \boldsymbol{L}_{\mathrm{raw}}\right)^{\mathrm{T}} \boldsymbol{L}_{\mathrm{g}}^{\mathrm{T}} \\ &= \boldsymbol{L}_{\mathrm{g}} \boldsymbol{L}_{\mathrm{g}}^{\mathrm{T}} = \Sigma_{\mathrm{g}}^c,\end{aligned} \tag{27}$$

completing the proof for the covariance alignment.

B Algorithm Details

In ABBL, the coefficient β_{SCL} of $\mathcal{L}_{\mathrm{A\text{-}SCL}}$ decays according to a cosine annealing schedule as the communication round r increases:

$$\beta_{\mathrm{SCL}}^{(r)} = \beta_{\mathrm{SCL}}^{(1)} \cdot \frac{1}{2}\left(1 + \cos\left(\frac{r}{R}\pi\right)\right), \tag{28}$$

where $\beta_{\mathrm{SCL}}^{(r)}$ denotes the weight coefficient at round r, and R represents the total number of rounds. We set $\beta_{\mathrm{SCL}}^{(1)}$ to 1 for CIFAR-100-LT and to 0.1 for other datasets.

In statistics aggregation, instead of directly transmitting the random mapping matrix $\left[\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{D/2}\right]$ to clients, only the random seed r_0 needs to be distributed. Each client can then locally build the matrix using the same seed. The process of statistics aggregation is illustrated in Algorithm 2. To reduce communication overhead, the local statistics are uploaded and aggregated only once, after the last round.

In SAFS, to ensure that Σ_{g}^c and Σ_{raw}^c can be Cholesky decomposed, a small positive value such as 10^{-5} is added to the matrix's diagonal elements to guarantee its positive definiteness. We synthesize $M_c = 2000$ features for the smallest class and $M_c = 600$ for the largest class, with M_c for the remaining classes linearly scaled between these two values based on the ranking of class sizes. This helps to alleviate the bias toward head classes.

C Experiment Details

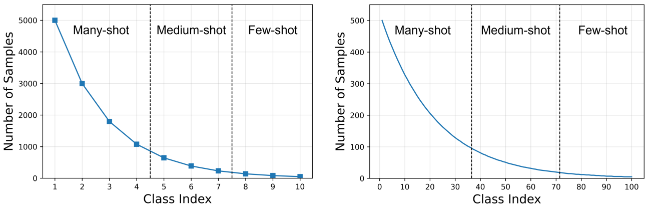
C.1 Dataset Details

To compute the group-wise accuracies in Section 5.2, for the CIFAR-100-LT dataset, we follow the partitioning scheme in [21]: classes with more than 100 training samples are assigned to the Many-shot group, classes with between 20 and 100 training samples are assigned to the Medium-shot group, and classes with fewer than 20 training samples are assigned to the Few-shot group. For the CIFAR-10-LT dataset, since there are no classes with fewer than 20 training samples, we group them as follows: classes with more than 1000 training samples are assigned to the Many-shot group, classes with between 200 and 1000 samples are assigned to the Medium-shot group, and classes with fewer than 200 samples are assigned to the Few-shot group. The class distribution and grouping for both datasets is shown in Fig. 7.

Algorithm 2 Statistics Aggregation

```
Input: Trained global model parameters $\theta$, number of clients $K$,
    number of classes $C$, random feature dimension $D$.
Output: Global statistics $\{\boldsymbol{\mu}_g^c, \Sigma_g^c, \bar{\boldsymbol{\phi}}_g^c\}_{c=1}^C$.
$r_0 \leftarrow$ Generate a random seed
for each client $k = 1$ to $K$ in parallel do
    $\{\boldsymbol{\mu}_k^c, S_k^c, \bar{\boldsymbol{\phi}}_k^c, n_k^c\}_{c=1}^C \leftarrow$ ClientStats($k$, $\theta$, $r_0$)
end for
for each class $c = 1$ to $C$ in parallel do
    $\{\boldsymbol{\mu}_g^c, \Sigma_g^c, \bar{\boldsymbol{\phi}}_g^c\}_{c=1}^C \leftarrow$ Aggregate to obtain global statistics according to Eq. (10)
end for

function ClientStats($k$, $\theta$, $r_0$):
    Set random seed to $r_0$
    for $i = 1$ to $D/2$ do
        Randomly generate $\boldsymbol{\omega}_i$ according to the specified random feature method
    end for
    Construct $\tilde{\phi}(\cdot)$ according to Eq. (3)
    Set the encoder model parameters as $\theta$
    for each class $c = 1$ to $C$ do
        $\mathcal{Z}_c = \{\text{Encoder}(\boldsymbol{x}_i) \mid y_i = c\}$
        $\boldsymbol{\mu}_k^c, S_k^c, \bar{\boldsymbol{\phi}}_k^c, n_k^c \leftarrow$ Compute local statistics according to Eq. (9)
    end for
    return $\{\boldsymbol{\mu}_k^c, S_k^c, \bar{\boldsymbol{\phi}}_k^c, n_k^c\}_{c=1}^C$
```



(a) CIFAR-10-LT **(b) CIFAR-100-LT**

Figure 7: Class grouping and class distribution of CIFAR-10-LT and CIFAR-100-LT (IF = 100).

C.2 Implementation Details

For FedProx, we tune the proximal term coefficient μ from $\{0.01, 0.001\}$. For FedDecorr, we set the coefficient β of the feature decorrelation term to the recommended value of 0.1. For FedLC, we tune the hyperparameter τ of the fine-grained calibrated cross-entropy loss from $\{1.0, 0.1\}$. For FedRoD, we evaluate the global model using the predictions from its generic head, as our focus is on generic FL performance. For FedLoGe, we similarly use the predictions from its global classifier to evaluate the global model. We tune the sparse ratio β of the SSE-C in FedLoGe from $\{0.4, 0.6\}$. For FLea, we follow the settings in the original paper, with the distillation loss coefficient λ_1 set to 1, the decorrelation regularization coefficient λ_2 set to 3, and each client sharing $\alpha = 10\%$ of their local features. We share the intermediate features output by the model's fifth convolutional layer, consistent with the original settings. For Fed3R, the regularization coefficient λ for ridge regression is set to the recommended value of 0.01 as suggested in the paper. For SFD, we use an RBF kernel (gamma = 0.01) for the MMD, and the dimensionality of random features D is set to 5000. For SFD, CCVR, and CReFF, we retrain (i.e., fine-tune) the global classifier on the server using SGD with a learning rate of 0.01 and momentum of 0.9.

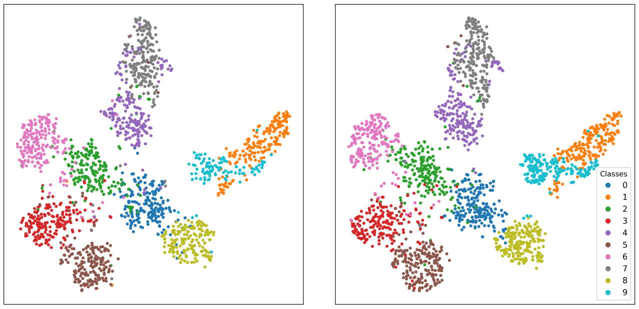
C.3 Data Reconstruction Attack

We follow the approach of prior studies [32, 35] to conduct image reconstruction attack experiments. Specifically, the attacker trains a decoder to recover original images from their features. Following the setting in [35], the attacker targets the global model in FL. In this experiment, we use the global model trained on CIFAR-10-LT (IF = 100, α = 0.05).

To train the decoder, the attacker typically leverages an external dataset. Here, we use the CIFAR-100 dataset as the attacker's external dataset, with its images serving as training data for the decoder. During training, the features extracted by the global model are used as input to the decoder, which generates reconstructed images. The decoder is optimized by minimizing the Mean Squared Error (MSE) and Mean Absolute Error (MAE) between the reconstructed images and the original images.

C.4 Authenticity of Synthetic Features

We visualize the real features on CIFAR-10-LT and the synthetic features constructed by SAFS using t-SNE (up to 200 features per class). As shown in Fig. 8, the two distributions are similar, demonstrating the authenticity of the synthetic features.



(a) Real features **(b) Synthetic features**

Figure 8: T-SNE visualization of the real features and the synthetic features constructed by SAFS.