# A. Related Work

## Federated Learning with Data Heterogeneity

In federated learning, client data often originates from different distributions, typically manifesting as label skew and domain skew. With label skew, the class distributions across clients are significantly imbalanced, while domain skew occurs when feature distributions for the same class vary due to differences in data sources. To address these issues, researchers have proposed methods that can be grouped into client regularization, server-side dynamic aggregation, and federated data augmentation [13].

Client regularization primarily focuses on adjusting local optimization objectives so that local models align more closely with the direction of the global model, reducing distributional shifts across clients [18, 22, 26, 31, 40, 41, 46, 50, 58, 60, 65]. Methods such as FedProx [29] and SCAFFOLD [16] introduce additional regularization terms to minimize the discrepancy between local and global models, thereby improving convergence speed and accuracy. MOON [25] leverages contrastive learning to align feature spaces across clients, addressing both label and domain skew. FPL [12] supervises the learning of local class prototypes by aggregating and sharing class prototypes across clients. However, involving a global model in the local optimization process deeply enlarges the local computation cost and linearly increases with the parameter scale.

Server-side dynamic aggregation methods optimize the global model by adaptively adjusting client weights [4, 10, 39]. FedOPT [48] and Elastic [3] use dynamic aggregation weights based on client model updates, enhancing the global model's generalization in heterogeneous data settings. Additionally, methods like FedDF [34] and FCCL [11] incorporate knowledge distillation modules on the server side, combined with auxiliary datasets to improve the adaptability of aggregation, making these approaches suitable for broader cross-client data distributions. However, these methods often require additional proxy datasets to support model adjustments, which is beneficial in scenarios with significant distributional differences across clients.

Federated data augmentation aim to bridge the gap between local data distributions and the ideal global distribution by generating more diverse data samples on clients [9, 37, 51]. For example, FedMix [63] and FEDGEN [68] use MixUp and its variants to augment client data, thus mitigating label skew. However, due to the lack of knowledge-based guidance, these methods largely rely on the diversity of local data. FedFA [67] assumes local data follows a Gaussian distribution and generates new samples centered on class prototypes. Nevertheless, the Gaussian assumption is overly idealistic [38], limiting the ability of generated samples to adequately reduce the discrepancy between local and global distributions. **Compared to** the Gaussian assumption, the geometric shapes proposed in this work provide a more accurate description of embedding distributions. GGEUR estimates the geometric shape of the global distribution without compromising privacy and leverages it to guide data augmentation on clients.

# B. Dataset

**Label Skew Datasets.** We evaluate our method on three single-domain image classification tasks.

- **Cifar-10** [20] contains 10 classes, with $50,000$ images for training and $10,000$ images for validation.
- **Cifar-100** [20] covers 100 classes, with $50,000$ training images and $10,000$ validation images.
- **Tiny-ImageNet** [6] is the subset of ImageNet with $100K$ images of size $64 \times 64$ with 200 classes scale.

Consistent with recent benchmarks [13], we set up 10 clients for each task. To simulate label skew across clients, we use a Dirichlet distribution, $Dir(\beta)$, where the parameter $\beta > 0$ controls the degree of label skew (i.e., class imbalance). When $\beta$ takes a smaller value, the local distributions generated on each client become more skewed, showing greater divergence from the overall distribution.

**Domain Skew Datasets.** We evaluated our method on the multi-domain image classification dataset Digits and conducted analyses on Office-Caltech and PACS.

- **Digits** [14, 21, 43, 49] includes four domains: MNIST, USPS, SVHN and SYN, each with 10 categories.
- **Office-Caltech** [8] includes four domains: Caltech, Amazon, Webcam, and DSLR, each with 10 categories.
- **PACS** [24] includes four domains: Photo (P) with $1,670$ images, Art Painting (AP) with $2,048$ images, Cartoon (Ct) with $2,344$ images and Sketch (Sk) with $3,929$ images. Each domain holds seven categories.

Consistent with recent benchmarks, in domain skew experiments, each domain is assigned to a separate client, with each client focusing on data from a specific domain. For Digits, each client is allocated $10\%$ of the data from its respective domain. For Office-Caltech and PACS, each client is allocated $30\%$ of the data from its corresponding domain.

**Dataset with Coexisting Label Skew and Domain Skew.** **Office-Home** [56] includes 4 domains: Art (A), Clipart (C), Product (P), and Real World (R), each containing 65 classes. To increase the challenge, we designed a new partitioning method for the multi-domain dataset Office-Home to create a scenario where label skew and domain skew coexist.

In the Office-Home dataset, while ensuring that each client corresponds to a single domain, we first generate a Dirichlet coefficient matrix, where the degree of class imbalance is controlled by $\beta$. For the 65-class, 4-domain Office-Home task, we generate a $4 \times 65$ matrix controlled by $\beta$ (with each column summing to 1). The four coefficients for each class are then allocated to the four clients, and each client uses its assigned coefficients to determine the number

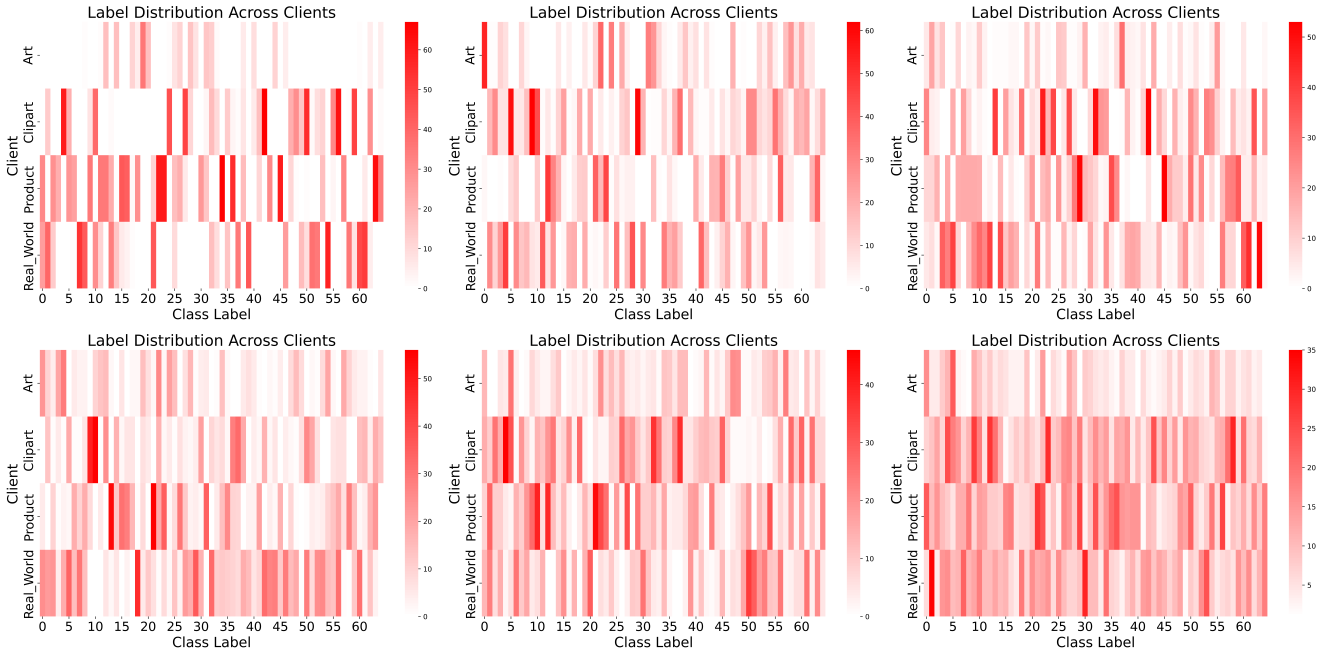


Figure 9. Number of samples per class across four clients when $\beta$ equals 0.1, 0.3, 0.5, 0.7, 1, and 5, with each client holding data from a different domain.

Table 8. Experiments Configuration of different federated scenarios. Image size is operated after the resize operation. $|C|$ denotes the classification scale. $|K|$ denotes the clients number. $E$ is the communication epochs for federation. $B$ is the training batch size.

| **Scenario** | **Size** | $\|C\|$ | **Network** $w$ | **Rate** $\eta$ | $\|K\|$ | $E$ | $B$ |
|---|---|---|---|---|---|---|---|
| **Label Skew Setting § 5.2** | | | | | | | |
| Cifar-10 | 224 | 10 | CLIP (ViT-B/16) | 1e-2 | 10 | 100 | 64 |
| Cifar-100 | 224 | 100 | CLIP (ViT-B/16) | 1e-1 | 10 | 100 | 64 |
| Tiny-ImageNet | 224 | 200 | CLIP (ViT-B/16) | 1e-2 | 10 | 100 | 64 |
| **Domain Skew § 5.3 and 5.4** | | | | | | | |
| Digits | 224 | 10 | CLIP (ViT-B/16) | 1e-2 | 4 | 50 | 16 |
| Office Caltech | 224 | 10 | CLIP (ViT-B/16) | 1e-3 | 4 | 50 | 16 |
| PACS | 224 | 7 | CLIP (ViT-B/16) | 1e-3 | 4 | 50 | 16 |
| Office-Home | 224 | 65 | CLIP (ViT-B/16) | 1e-3 | 4 | 50 | 16 |

of samples for that class. This setup results in a distribution that incorporates both domain shift (one domain per client) and Dirichlet-based class imbalance, presenting a scenario where the model faces both class distribution and domain differences, creating a more realistic, challenging, and diverse distribution for classification. We name the newly constructed dataset **Office-Home-LDS (Label and Domain Skew)**. Figure 9 shows the data distribution of Office-Home-LDS with different $\beta$ values. Dataset and Constructor published at: https://huggingface.co/datasets/WeiDai-David/Office-Home-LDS.

Table 9. Hyper-parameters chosen for different methods. Hyper-parameters in different methodologies may share the same notation but represent distinct meanings.

| **Methods** | **Hyper-Parameter** | **Parameter value** |
|---|---|---|
| SCAFFOLD | Global learning rate $lr$ | 0.25 |
| MOON | Contrastive temp $\tau$ | 0.5 |
| | Proximal weight $\mu$ | 1.0 |
| FedDyn | Proximal weight $\alpha$ | 0.5 |
| FedOPT | Global learning rate $\eta_g$ | 0.5 |
| FedProto | Proximal weight $\lambda$ | 2 |
| FedNTD | Distill temp $\tau$ | 1 |
| | Reg weight $\beta$ | 1 |

## C. Implementation Details

As for the uniform comparison evaluation, we follow [13] and conduct the local updating round $U = 10$. We use the SGD optimizer for all local updating optimization. The corresponding weight decay is $1 \times 10^{-5}$ and momentum is 0.9. The learning rate $\eta$ and communication epoch $E$ are different in various scenarios, as shown in Table 8. Notably, the communication epoch is set according to when all federated approaches have little or no accuracy gain with more communication epochs. The local training batch size is $B = 64$. Furthermore, the Table 9 plots the chosen hyper-parameter for different methods.

## D. Privacy Constraints

In our approach, the server only sends the eigenvectors and eigenvalues of the global covariance matrix back to clients, without sharing raw data or local covariance matrices. We demonstrate below that this information is insufficient for reconstructing any client's original data.

**(1) Eigenvectors and Eigenvalues Do Not Contain Raw Data.** The eigendecomposition provides only the geometric structure of the data distribution, without encoding individual sample details. Even if a client obtains eigenvectors and eigenvalues, reconstructing the original data is an **ill-posed problem**, as it admits infinitely many solutions.

**(2) Low-Rank Property Prevents Data Reconstruction.** The covariance matrix is typically **low-rank**, meaning: $\text{rank}(\Sigma_i) \ll d$, where $d$ is the original data dimension. This implies that even with full knowledge of eigenvectors and eigenvalues, clients can only access principal directions of the data and not its full details.

**(3) Aggregation Prevents Isolation of Individual Client Contributions.** The global covariance matrix is an aggregate of all clients' local covariance matrices: $\Sigma_i = \sum_{k=1}^{K} \frac{n_k^i}{N_i} \Sigma_k^i + \sum_{k=1}^{K} \frac{n_k^i}{N_i} (\mu_k^i - \mu_i)(\mu_k^i - \mu_i)^T$. Since each client's contribution is mixed through weighted averaging: I. No single client can **isolate another client's contribution** from Geometric Knowledge. II. Even if a client's data is removed, the impact on $\Sigma_i$ is distributed across all eigenvectors, making individual influences indistinguishable.

**(4) Existing Literature Supports the Privacy of Covariance Matrices.** Prior works confirm that sharing higher-order statistics (covariance matrices) poses lower privacy risks than sharing model gradients (Melis et al.).

## E. Large-Scale Client

In practical federated learning settings, the number of participating clients can significantly affect model performance due to increased data heterogeneity. To further evaluate the performance of our proposed method under larger-scale federated learning scenarios, we conducted additional experiments with an increased number of clients. Specifically, we conducted experiments on the label-skewed dataset CIFAR-10 with 100, 300, and 500 clients. As shown in Table 10, the results demonstrate that GGEUR remains robust and continues to enhance the performance of FedAvg (CLIP+MLP).

Table 10. Number of Clients $K$ Impact on Performance.

| Methods | CIFAR-10 ($\beta = 0.1$) | | |
|---|---|---|---|
| | $K = 100$ | $K = 300$ | $K = 500$ |
| FedAvg (CLIP+MLP) | 87.89 | 84.69 | 82.05 |
| + **GGEUR** | 93.55 (+5.66) | 92.17 (+7.84) | 90.43 (+8.38) |

**Algorithm 2** GGEUR (Multi-Domain Scenario)

**Require:** $X_k^i = [X_k^{(i,1)}, \ldots, X_k^{(i,n_k^i)}] \in \mathbb{R}^{p \times n_k^i}$: Sample set of class $i$ at client $k$, $GD_i = \{\xi_i^1, \ldots, \xi_i^p, \lambda_i^1, \ldots, \lambda_i^p\}$: Shared geometric shape (eigenvectors and eigenvalues) of class $i$, $\{\mu_{k'}^i\}$: Prototypes (means) of class $i$ from other domains, $N$: Number of new samples to generate per original sample in Step 1, $M$: Number of samples to generate per prototype in Step 2.

**Ensure:** $X_{\text{new}}^i$: Augmented sample set of class $i$ at client $k$

$X_{\text{new}}^i \leftarrow \emptyset$ ▷ Initialize augmented sample set

▷ Step 1: Local Domain Augmentation

**for** $j = 1$ to $n_k^i$ **do**

  $X_{\text{new}}^i \leftarrow X_{\text{new}}^i \cup \text{GGEUR}(X_k^{(i,j)}, GD_i, N)$

**end for**

▷ Step 2: Cross-Domain Simulation

**for** each prototype $\mu_{k'}^i$ from other domains **do**

  $X_{\text{new}}^i \leftarrow X_{\text{new}}^i \cup \text{GGEUR}(\mu_{k'}^i, GD_i, M)$

**end for**

**return** $X_{\text{new}}^i$

## F. Computational Cost

We conducted experiments on the domain-skewed dataset Digits, comparing the training time required to complete the full model for FedAvg (CLIP+MLP), SCAFFOLD (CLIP+MLP), and MOON (CLIP+MLP) before and after applying GGEUR. The results (Tab11) show that GGEUR introduces almost no additional training time overhead. Specifically, after applying GGEUR, the training time for the three methods increased by only 3.5s, 4.6s, and 3.3s, respectively.

Table 11. The average training time (s) per round.

| Methods | Digits | | |
|---|---|---|---|
| | FedAvg | SCAFFOLD | MOON |
| CLIP+MLP | 28.2 | 54.5 | 32.3 |
| + **GGEUR** | 31.7 (+3.5) | 59.1 (+4.6) | 35.6 (+3.3) |