# Soft Separation and Distillation: Toward Global Uniformity in Federated Unsupervised Learning

Hung-Chieh Fang
National Taiwan University

Hsuan-Tien Lin
National Taiwan University

Irwin King
The Chinese University of Hong Kong

Yifei Zhang
The Chinese University of Hong Kong

## Abstract

*Federated Unsupervised Learning (FUL) aims to learn expressive representations in federated and self-supervised settings. The quality of representations learned in FUL is usually determined by uniformity, a measure of how uniformly representations are distributed in the embedding space. However, existing solutions perform well in achieving intra-client (local) uniformity for local models while failing to achieve inter-client (global) uniformity after aggregation due to non-IID data distributions and the decentralized nature of FUL. To address this issue, we propose Soft Separation and Distillation (SSD), a novel approach that preserves inter-client uniformity by encouraging client representations to spread toward different directions. This design reduces interference during client model aggregation, thereby improving global uniformity while preserving local representation expressiveness. We further enhance this effect by introducing a projector distillation module to address the discrepancy between loss optimization and representation quality. We evaluate SSD in both cross-silo and cross-device federated settings, demonstrating consistent improvements in representation quality and task performance across various training scenarios. Our results highlight the importance of inter-client uniformity in FUL and establish SSD as an effective solution to this challenge.*

## 1. Introduction

Deep learning has achieved remarkable success across a wide spectrum of applications, from computer vision and natural language processing to speech recognition and reinforcement learning [1, 11, 18, 29]. This progress can be largely attributed to the availability of massive labeled datasets, particularly since the emergence of ImageNet [5], which has enabled the training of increasingly powerful neural networks. However, the practical deployment of deep learning in real-world scenarios faces two critical challenges. First, data in real-world applications is often non-Independent and Identically Distributed (non-IID) and cannot be freely shared due to privacy concerns, regulatory restrictions, or proprietary constraints. This data is typically generated across diverse sources, including user devices, healthcare institutions, and industrial settings. The distributed nature and sharing limitations of such data make centralized training approaches infeasible, leading to the development of Federated Learning (FL) [15, 20, 24] as a privacy-preserving distributed learning paradigm. Second, a vast proportion of available data remains unlabeled, necessitating effective methods for unsupervised representation learning [2, 3, 8, 12]. These methods aim to learn mean-

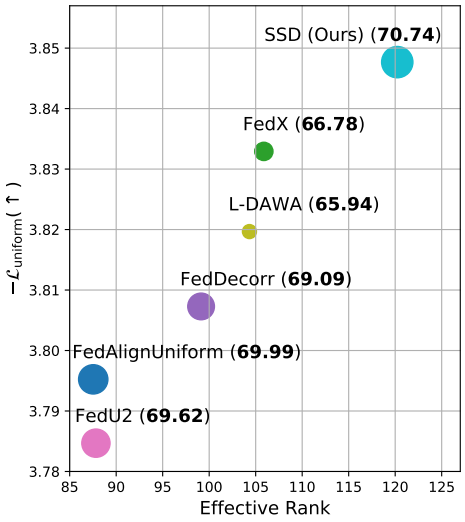



Figure 1. **Comparison with state-of-the-art methods.** Our SSD achieves the *highest* performance (**marked in brackets**) and the best representation quality, measured by uniformity [31] and effective rank [27], among existing FUL approaches.

ingful feature representations without relying on explicit labels, enabling models to capture underlying data structures and semantic relationships.

Federated Unsupervised Learning (FUL) emerges at the intersection of these two challenges, aiming to learn expressive representations in settings where data is both distributed and unlabeled. In FUL, the quality of learned representations is critically determined by two properties: alignment and uniformity [31]. Alignment measures how close similar data points are positioned in the representation space, relating to the model's ability to group semantically similar objects. Uniformity quantifies how evenly representations are distributed across the unit hypersphere, essentially measuring the entropy of the representation distribution. High uniformity prevents representation collapse and ensures that the learned features effectively utilize the available embedding dimensions.

However, existing FUL methods face a significant challenge that remains inadequately addressed. While current approaches successfully achieve good intra-client (local) uniformity for representations within each client, they struggle to maintain inter-client (global) uniformity after model aggregation. This limitation stems from two key factors: (1) the non-IID distribution of data across clients, which naturally leads to divergent updates, and (2) the decentralized nature of FL, where the server lacks direct access to raw data, preventing the application of explicit uniformity constraints across clients.

Most existing approaches have focused primarily on two aspects of this challenge. One line of research builds upon the framework established by FedProx [21], introducing proximal terms to constrain local updates within a bounded neighborhood of the global model. For instance, FedU [36] and FedEMA [37] dynamically adjust aggregation weights based on inter-client model divergence, while FedX [10] incorporates a global relational loss to align pairwise sample relationships across clients. These methods aim to maintain global model consistency but do not explicitly address representation uniformity.

Another research direction tackles the issue of dimensional collapse [14] caused by lower uniformity in local representations. FedDecorr [28] demonstrates that embeddings of local clients are often less uniformly distributed and mitigates this issue by decorrelating local features. Similarly, FedU2 [22] regularizes local updates to approximate a spherical Gaussian distribution, encouraging isotropic feature spaces. While these approaches enhance local uniformity, the improvements at the client level do not inherently translate to better global uniformity during model aggregation. This leads to a crucial question:

*How can we effectively improve inter-client uniformity in Federated Unsupervised Learning?*

To address this challenge, we propose Soft Separation and Distillation (SSD), a novel method that enhances inter-client uniformity without compromising local representation expressiveness. Our approach employs a dimension-scaled regularization strategy that softly separates each client's feature space, encouraging client representations to spread toward different directions. As illustrated in Figure 2, this technique reduces interference during client model aggregation, thereby improving global uniformity without imposing rigid boundaries that could distort the underlying feature distributions.

Furthermore, we empirically observe that the regularization effect of SSD may not always effectively transfer from loss optimization to the representation space due to the presence of a projector between the loss function and the encoder. While removing the projector might seem like a straightforward solution, doing so often degrades performance because the projector plays a crucial role in separating optimization objectives from feature representations [2, 9, 32]. To bridge this gap, we introduce a projector distillation module that minimizes the KL divergence between representations and embeddings, effectively encouraging the encoder to internalize the learned structure while preserving the projector's role in loss optimization.

We evaluate SSD across diverse federated learning scenarios, including cross-silo (few clients with large datasets) and cross-device (many clients with small datasets) settings. Our experiments span both in-distribution tasks and out-of-distribution (OOD) datasets to assess generalizability. Results demonstrate that SSD consistently outperforms existing methods in both downstream task performance (measured via linear probing and fine-tuning accuracy) and representation quality (quantified using effective rank [7] and uniformity [31] metrics).

Our main contributions are threefold:

- We identify and formalize the challenge of inter-client uniformity in FUL, establishing it as a critical direction for decentralized unsupervised representation learning.
- We propose Soft Separation and Distillation (SSD), a simple yet effective framework that addresses inter-client uniformity without additional communication overhead or compromising privacy.
- We conduct extensive experiments across varied FL settings and tasks, confirming SSD's superiority over baseline methods in both performance and robustness.

## 2. Preliminaries

**Federated unsupervised learning (FUL)** is a learning paradigm in which multiple clients collaboratively train a model without sharing new raw data, and where the local datasets contain unlabeled data. The goal is to learn a global representation that generalizes across all participat-

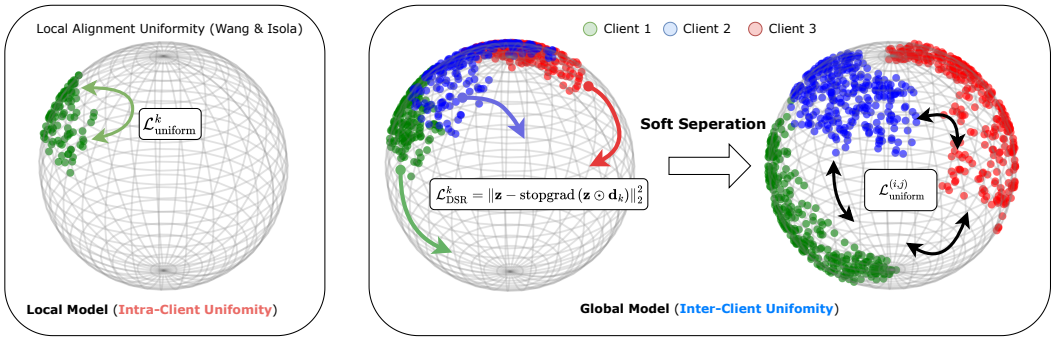



Figure 2. **Illustration of Intra-Client and Inter-Client Uniformity with Soft Separation.** Intra-client uniformity ensures that representations within each client are well-distributed, while inter-client uniformity promotes global representation consistency across clients. Our proposed Soft Separation method encourages each client's representations to spread in distinct directions, mitigating interference during model aggregation and improving global uniformity.

ing clients. Consider a federated learning system with $K$ local clients and a central server. Each client $k$ has access to an unlabeled local dataset $\mathcal{D}_k$, modeled as samples from a client-specific data distribution $p_k(\mathbf{x})$:

$$\mathcal{D}_k = \{\mathbf{x}_i^k\}_{i=1}^{n_k} \sim p_k(\mathbf{x}), \tag{1}$$

where $n_k$ denotes the number of samples in $\mathcal{D}_k$. The global objective in FUL can be formulated as:

$$\min_{\theta} \sum_{k=1}^{K} P_k \mathbb{E}_{\mathbf{x}\sim p_k(\mathbf{x})}\left[\mathcal{L}_k(\theta; \mathbf{x})\right], \tag{2}$$

where $P_k$ is the probability of drawing a data from client $k$, and $\mathcal{L}_k(\cdot)$ denotes the local unsupervised loss of client $k$.

**Self-Supervised Representation Learning** aims to learn meaningful representations that capture the underlying structure of data. A typical framework consists of the following components:

- **Encoder** $f(\cdot)$: Maps (augmented) input data $\tilde{\mathbf{x}} \sim \mathcal{T}(\mathbf{x})$ into a *representation* $\mathbf{h} = f(\tilde{\mathbf{x}}) \in \mathbb{R}^d$, where $d$ is the hidden dimension. It is common to use a deep neural network (*e.g.*, ResNet50) as the encoder.

- **Projector** $g(\cdot)$: Transforms the representation $\mathbf{h}$ into an *embedding* $\mathbf{z} = g(\mathbf{h}) \in \mathbb{R}^d$. The training loss function is usually applied on this embedding space, and the projector is often a small network such as a multilayer perceptron.

A fundamental principle in self-supervised representation learning is that similar samples should have similar representations. In other words, representations should be invariant to minor variations in input data. This is typically achieved by aligning the representations of two augmented versions of the same image (referred to as a positive pair) using the following alignment loss:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\tilde{\mathbf{x}},\tilde{\mathbf{x}}^+\sim\mathcal{T}(\mathbf{x})}||\mathbf{z} - \mathbf{z}^+||_2^2, \tag{3}$$

where $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}^+ \sim \mathcal{T}(\mathbf{x})$ denotes two independent augmentations of the same sample $\mathbf{x}$ and $\mathbf{z} = g(f(\tilde{\mathbf{x}})), \mathbf{z}^+ = g(f(\tilde{\mathbf{x}}^+))$ are the corresponding embeddings.

Although aligning positive pairs encourages similarity, it can lead to a degenerate *collapse* where all representations converge to a single point. To circumvent this, Wang and Isola [31] introduces a *uniformity* objective to ensure that features are evenly dispersed on the unit hypersphere. This uniformity is often measured using the log of the average Gaussian potential [4]:

$$\mathcal{L}_{\text{uniform}} = \log \mathbb{E}_{\mathbf{z}_i,\mathbf{z}_j \overset{\text{i.i.d}}{\sim} p(\mathbf{z})}\left[e^{-t||\mathbf{z}_i - \mathbf{z}_j||_2^2}\right] \tag{4}$$

where $p(\mathbf{z})$ is the distribution of embeddings obtained by mapping data samples through the encoder $f$ and projector $g$ and $t$ is the temperature hyperparameter.

## 3. Methodology

In this section, we introduce *Soft Separation and Distillation (SSD)*, a novel approach designed to enhance representation uniformity in federated learning (FL). We first analyze the limitations of uniformity in non-IID FL settings, then propose dimension-scaled regularization to softly separate client features, and finally introduce projector distillation to effectively transfer improved embedding uniformity to the representations.

**Limited Inter-Client Uniformity of FL.** Uniformity is a critical metric in representation learning [6, 30, 31], where higher uniformity indicates better preservation of information in the learned representations. In centralized training, jointly optimizing alignment loss $\mathcal{L}_{\text{align}}$ and uniformity loss $\mathcal{L}_{\text{uniform}}$ yields high-quality representations. However, the distributed nature of FL introduces a fundamental challenge to uniformity optimization. Specifically, the uniformity metric can be decomposed into intra-client and inter-

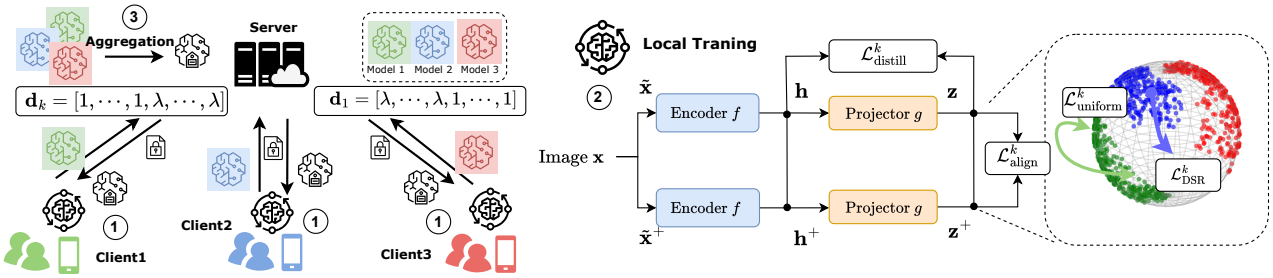



Figure 3. **Overview of our Training Pipeline.** ① The server *initially* assigns distinct weight vectors to each client. ② Clients perform local training by optimizing their respective loss functions on their own data. ③ The server aggregates updated parameters from all clients. Steps ② and ③ are iteratively repeated until convergence.

client components:

$$\mathcal{L}_{\text{uniform}} = \log\Bigg(\underbrace{\sum_{k=1}^{K} \mathbb{E}_{\mathbf{z},\mathbf{z}' \overset{\text{i.i.d}}{\sim} p_k(\mathbf{z})}\left[e^{-t\|\mathbf{z}-\mathbf{z}'\|_2^2}\right]}_{\text{intra-client } \mathcal{L}^k_{\text{uniform}}} + \underbrace{\sum_{i \neq j} \mathbb{E}_{\mathbf{z}\sim p_i(\mathbf{z}),\mathbf{z}'\sim p_j(\mathbf{z})}\left[e^{-t\|\mathbf{z}-\mathbf{z}'\|_2^2}\right]}_{\text{inter-client } \mathcal{L}^{(i,j)}_{\text{uniform}}}\Bigg), \tag{5}$$

where the first term represents intra-client uniformity $\mathcal{L}^k_{\text{uniform}}$ and the second term represents inter-client uniformity $\mathcal{L}^{(i,j)}_{\text{uniform}}$.

In federated learning, each client $i$ optimizes its local loss function using samples drawn from its own local distribution $p_i$, which means that only the intra-client term in the above equation is explicitly optimized. In a homogeneous setting, when client distributions are similar (i.e., $p_i \approx p_j$), optimizing intra-client uniformity naturally promotes good inter-client uniformity. However, in non-IID settings where client distributions differ significantly, optimizing only intra-client uniformity does not guarantee good inter-client uniformity, potentially limiting the quality of the globally aggregated model. The core challenge in enhancing inter-client uniformity lies in the federated learning constraint that the server has no access to raw client data or embeddings, making it impossible to directly impose a loss function that operates across different clients.

**Dimension-Scaled Regularization (DSR).** To address the inter-client uniformity challenge, we propose Dimension-Scaled Regularization (DSR), which improves global uniformity by encouraging client embeddings to spread in different directions of the representation space. This approach effectively increases the separation between clients without enforcing rigid boundaries that might distort the intrinsic data structure.

For each client $k$, we define a dimension-scaling vector $\mathbf{d}_k \in \mathbb{R}^d$, where $d$ is the dimensionality of the embedding space. This vector applies selective scaling to specific dimensions, with some dimensions scaled by a factor $\alpha \neq 1$ in a client-specific manner:

$$d_{i,k} = \begin{cases} \alpha, & \text{if } i \in \mathcal{S}_k \\ 1, & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathcal{S}_k$ is a set of dimensions uniquely assigned to client $k$, ensuring that the scaled dimensions are non-overlapping across clients (i.e., $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$). The size of each set $\mathcal{S}_k$ is approximately $\lfloor d/K \rfloor$, where $K$ is the number of clients.

We then regularize each client's embeddings by encouraging them to move toward their dimension-scaled versions through the following loss:

$$\mathcal{L}^k_{\text{DSR}} = \mathbb{E}_{\mathbf{z}\sim p_k(\mathbf{z})}\left[\|\mathbf{z} - \text{stopgrad}(\mathbf{z} \odot \mathbf{d}_k)\|_2^2\right], \tag{7}$$

where $\odot$ represents element-wise multiplication, and $\text{stopgrad}(\cdot)$ prevents gradient flow through the scaled target, ensuring that we pull the original embedding toward the scaled version rather than vice versa.

To understand why DSR enhances inter-client uniformity, we can analyze its effect on the representation distribution. When a vector $\mathbf{z}$ is scaled along specific dimensions by a factor $\alpha > 1$ and then normalized to the unit hypersphere, the resulting vector shifts toward those scaled dimensions. By assigning different scaling dimensions to each client, DSR effectively encourages client representations to occupy different regions of the unit hypersphere, as illustrated in Figure 5. Mathematically, if clients $i$ and $j$ have scaling vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ with non-overlapping scaled dimensions, their optimized embeddings will tend to have a smaller dot product:

$$\mathbb{E}_{\mathbf{z}_i\sim p_i(\mathbf{z}),\mathbf{z}_j\sim p_j(\mathbf{z})}[\mathbf{z}_i^\top \mathbf{z}_j] < \mathbb{E}_{\mathbf{z}_i,\mathbf{z}_j\sim p(\mathbf{z})}[\mathbf{z}_i^\top \mathbf{z}_j], \tag{8}$$

where $p(\mathbf{z})$ represents the distribution without DSR. This reduced dot product corresponds to increased angular separation, directly contributing to improved uniformity across the global distribution.

Unlike a hard separation approach that would restrict each client to entirely separate subspaces (which could

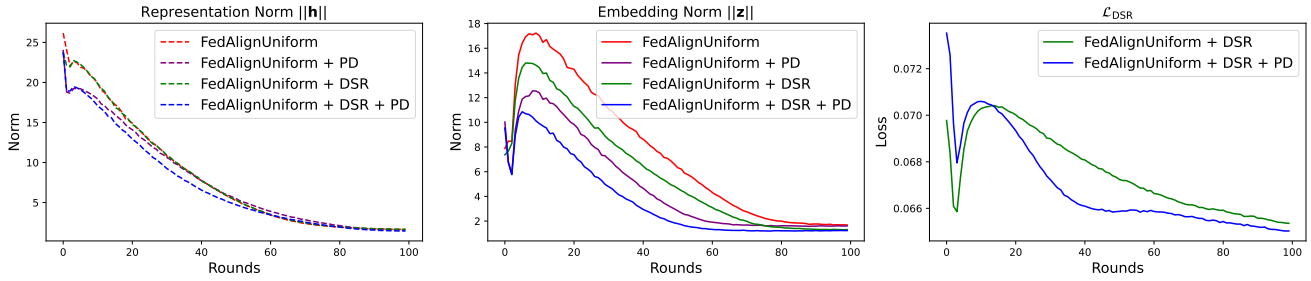



Figure 4. **Representation norm, embedding norm, and DSR loss during training.** Adding DSR primarily affects the embedding space, as reflected in the embedding norm, while leaving the representation norm unchanged. In contrast, incorporating PD directly influences the representation norm, and when combined with DSR, it leads to a more pronounced decrease in representation norm. Additionally, PD enhances the effect of DSR, resulting in a lower $\mathcal{L}_{\text{DSR}}$.

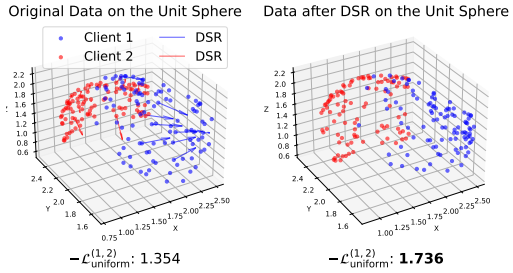



Figure 5. **Illustration of the effect of DSR.** The left figure shows the original data, where arrows indicate the transformation direction after applying DSR. The right figure presents the data after transformation. DSR increases separation and enhances inter-client uniformity.

severely constrain representation capacity), our soft separation allows clients to share most dimensions while gently pushing them toward different directions. This balanced approach preserves the flexibility needed for effective representation learning while significantly improving inter-client uniformity.

An extreme example of fully separating embeddings would be assigning each client a dedicated subspace (e.g., if $d = 500$ and $K = 10$, each client could occupy 50 distinct dimensions). Although this configuration maximizes inter-client uniformity, it may disrupt the intrinsic data structure and miss out on the collaborative benefits of FL. In contrast, our *soft* DSR loss encourages some degree of separation without completely isolating the clients' embeddings. We analyze this tradeoff further in Section 4.2.

**Projector Distillation (PD).** While DSR effectively enhances uniformity at the embedding level, we observe that this improvement may not fully transfer to the representation level. This discrepancy occurs because the projector $g(\cdot)$ placed between the encoder $f(\cdot)$ and the loss function can absorb much of the optimization effect, as shown in Figure 4. Although completely removing the projector might seem like a direct solution, prior work [2, 9, 32] has demonstrated that the projector plays a crucial role in separating optimization objectives from feature representations, thereby preventing overfitting to specific self-supervised tasks and improving downstream performance.

To bridge this gap, we introduce Projector Distillation (PD), which explicitly aligns the encoder's representations with the projector's embeddings:

$$\mathcal{L}^k_{\text{distill}} = \mathbb{E}_{x\sim p_k(\mathbf{x})}\left[D_{\text{KL}}\left(\sigma(\mathbf{h})\|\sigma(\mathbf{z})\right)\right], \tag{9}$$

where $\mathbf{h} = f(\mathbf{x})$ is the representation from the encoder, $\mathbf{z} = g(\mathbf{h})$ is the embedding from the projector, $\sigma(\cdot)$ denotes the softmax function, and $D_{\text{KL}}$ is the Kullback-Leibler divergence.

This distillation mechanism encourages the encoder to internalize the beneficial structure learned in the embedding space, effectively transferring the improved uniformity from embeddings to representations. By minimizing the KL divergence between these distributions, we ensure that the uniformity improvements achieved through DSR are reflected in the representations used for downstream tasks.

### 3.1. Overview of Training Pipeline

The complete SSD algorithm is outlined in Algorithm 1, with our key design components highlighted in red. The server initially assigns unique weight vectors to all clients for dimension-scaled regularization. During local training, each client optimizes a combination of the standard alignment and uniformity losses for self-supervised learning, augmented with our DSR and PD terms. The local training objective for each client $k$ is formulated as:

$$\mathcal{L}^k = \mathcal{L}^k_{\text{align}} + \beta\mathcal{L}^k_{\text{uniform}} + \gamma\mathcal{L}^k_{\text{DSR}} + \delta\mathcal{L}^k_{\text{distill}}, \tag{10}$$

where $\beta$, $\gamma$, and $\delta$ are hyperparameters that balance the contribution of each loss term.

After local training, clients upload their updated models to the server, which aggregates them using standard

**Algorithm 1:** *Soft Separation and Distillation (SSD)*

**Input:** communication rounds $T$, local epochs $E$, number of clients $K$, dataset $\mathcal{D} = \cup_{k\in[K]}\mathcal{D}_k$, augmentation function $\mathcal{T}(\cdot)$
**Output:** Final global encoder $f_\theta^T$

**Server:**
  Initialize global encoder $f_\theta^0$ and projector $g_\theta^0$;
  Assign weight vector $\{\mathbf{d}_k\}_{k\in[K]}$ for all clients;
  **for** $t = 0$ ***to*** $T-1$ **do**
    $S_t \leftarrow$ (randomly select a set of clients);
    **for** *each client* $k \in S_t$ *in parallel* **do**
      $f_\theta^{(t,k)}, g_\theta^{(t,k)} \leftarrow$ **Client**$(f_\theta^t, g_\theta^t, \mathbf{d}_k)$;
    **end**
    // Model aggregation;
    **for** *each parameter* $\theta$ *in* $f, g$ **do**
      $\theta^{t+1} \leftarrow \sum_{k\in S_t} w_k \theta^{(t,k)}$
    **end**
  **end**
  **return** $f_\theta^T$;
**Client** $(f_\theta^t, g_\theta^t, \mathbf{d}_k)$**:**
  // Initialize local models;
  $f \leftarrow f_\theta^t$, $g \leftarrow g_\theta^t$;
  **for** ***each local epoch*** $e = 0$ ***to*** $E-1$ **do**
    **for** ***each batch*** $\{\mathbf{x}_i\}_{i=1}^{n_B} \in \mathcal{D}_k$ **do**
      // Data augmentation;
      $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_i^+ = \mathcal{T}(\mathbf{x}_i), \mathcal{T}(\mathbf{x}_i)$ ;
      // Get representations and embeddings;
      $\mathbf{h}_i, \mathbf{h}_i^+ = f(\tilde{\mathbf{x}}_i), f(\tilde{\mathbf{x}}_i^+)$ ;
      $\mathbf{z}_i, \mathbf{z}_i^+ = g(\mathbf{h}_i), g(\mathbf{h}_i^+)$ ;
      // Optimize model using combined loss;
      Update $f, g$ by minimizing $\mathcal{L}^k$ (Eq (10));
    **end**
  **end**
  **return** $f, g$ // Model upload;
**return**

federated averaging [24]: $\theta^{t+1} = \sum_{k\in S_t} w_k \theta^{(t,k)}$ where $w_k = \frac{n_k}{\sum_{i=1}^K n_i}$ represents the aggregation weight based on the client's data size.

Importantly, our method preserves privacy since it requires no sharing of raw data. The only additional communication is the initial weight vectors $\{\mathbf{d}_k\}_{k=1}^K$, which are lightweight and contain no sensitive information. The computational overhead of SSD is minimal, making it highly practical for real-world federated learning deployments.

# 4. Experiments

**Datasets.** We evaluate our approach on CIFAR-10 and CIFAR-100, which contain 50K training images and 10K testing images, with 10 and 100 classes, respectively. Following prior work [22, 37], we simulate non-IID data across $K$ clients using a Dirichlet prior Dir($\boldsymbol{\alpha}$), where a smaller $\boldsymbol{\alpha}$ indicates a higher degree of data heterogeneity. We conduct experiments in two federated learning settings: cross-silo ($K = 10$) with full participation and cross-device ($K = 50$) with a participation rate of 0.2.

**Evaluation Metrics.** We evaluate our approach based on downstream performance and representation quality. For downstream performance, we follow prior work [10, 22, 36], using linear probing and fine-tuning with 1% and 10% of the data. For representation quality, we assess uniformity [6, 30, 31] as defined in Equation (4) and effective rank [7, 27, 34] (see the definition in Appendix A.2).

**Baselines.** We compare our method against three categories of baselines. (1) Adapting a centralized algorithm to the federated setting: FedAlignUniform [31]. (2) FUL methods that mitigate divergence from the global model: FedX [10] and L-DAWA [25]. (3) FUL methods addressing dimensional collapse in local clients: FedDecorr [28]. FedU2 [22] incorporates modules that tackle both divergence control and dimensional collapse.

**Implementation Details.** We follow prior work [22] for image augmentations, training configurations, and model architecture, using ResNet-18 [11] as the encoder and a two-layer linear projector. The hyperparameters $\beta$, $\gamma$, and $\delta$ are set to 1.0, 1.0, and 0.1, respectively, and the scaling factor $\alpha$ is set to 10.0. Further details are provided in Appendix A.1.

## 4.1. Main Results

**Performance on different FL settings.** Table 1 compares the performance of various methods in cross-silo and cross-device federated learning. SSD consistently achieves the highest accuracy and excels in fine-tuning tasks. FedAlignUniform, FedDecorr, and FedU2 are competitive but generally lag behind SSD, while FedX and L-DAWA—aimed at ensuring global model consistency—perform the worst across all settings. Although FedDecorr and FedU2 enhance intra-client uniformity (with FedU2 adding a consistency module), they still trail SSD. By improving inter-client uniformity, SSD underscores the importance of addressing inter-client variations in federated learning.

**Representation quality.** Figure 1 compares different methods based on effective rank [27] and uniformity [31], two key metrics for representation quality. Among state-of-the-art methods, FedX and L-DAWA, which emphasize global model consistency, achieve strong representation quality. In contrast, methods focusing on local uniformity, such as FedDecorr and FedU2, exhibit lower effective rank and uniformity, indicating that they fail to address global uniformity. Our method, SSD, surpasses all other federated approaches, achieving the highest effective rank and the best uniformity.

Table 1. **Results on Cross-Silo and Cross-Device Settings.** Accuracy (%) of linear probing (LP), fine-tuning (FT) 1%, and 10% labeled data on CIFAR10 and CIFAR100 ($\alpha = 0.1$). The highest score is highlighted in **bold**, and the second-highest score is underlined.

| | CIFAR10 | | | | | | CIFAR100 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cross-Silo (K=10) | | | Cross-Device (K=50) | | | Cross-Silo (K=10) | | | Cross-Device (K=50) | | |
| | LP | FT 1% | FT 10% | LP | FT 1% | FT 10% | LP | FT 1% | FT 10% | LP | FT 1% | FT 10% |
| FedAlignUniform [31] | 80.84 | 69.99 | 81.00 | 71.28 | 57.41 | 73.77 | 57.25 | 28.97 | 48.99 | 43.03 | 16.37 | 36.64 |
| FedX [10] | 78.4 | 66.78 | 80.01 | 71.01 | 56.91 | 73.24 | 57.34 | 27.46 | 49.50 | 43.07 | 16.04 | 35.46 |
| L-DAWA [25] | 77.67 | 65.94 | 79.34 | 67.65 | 53.75 | 71.22 | 56.90 | 27.08 | 49.57 | 42.58 | 15.54 | 34.82 |
| FedDecorr [28] | 80.13 | 69.09 | 80.33 | 71.49 | **58.19** | 73.97 | 57.25 | 29.38 | 49.53 | 44.74 | 17.67 | 36.68 |
| FedU2 [22] | 81.01 | 69.62 | 81.01 | 71.09 | 57.15 | 74.21 | **57.40** | 29.39 | 49.64 | 42.90 | 16.08 | 35.48 |
| **SSD** | **81.32** | **70.74** | **81.67** | **71.83** | 57.77 | **74.61** | 57.38 | **29.57** | **49.87** | **45.21** | **17.70** | **36.82** |

Table 2. **Generalization on OOD datasets.** Accuracy (%) of linear probing, uniformity, and effective rank when trained on CIFAR-100 and TinyImageNet-200, and evaluated on CIFAR-10.

| | CIRAR100 → CIFAR10 | | | TinyImageNet → CIFAR10 | | |
|---|---|---|---|---|---|---|
| | LP | $-\mathcal{L}_{\text{uniform}}(\uparrow)$ | ERank (↑) | LP | $-\mathcal{L}_{\text{uniform}}(\uparrow)$ | ERank (↑) |
| FedAlignUniform [31] | 77.66 | 3.65 | 66.99 | 79.86 | 3.71 | 76.31 |
| FedX [10] | 78.02 | **3.73** | 84.88 | 79.87 | 3.76 | 93.48 |
| L-DAWA [25] | 77.46 | 3.71 | 84.75 | 79.62 | 3.74 | 93.41 |
| FedDecorr [28] | 77.62 | 3.66 | 74.20 | 79.79 | 3.72 | 84.93 |
| FedU2 [22] | 77.74 | 3.66 | 68.3 | 79.74 | 3.69 | 75.97 |
| **SSD** | **78.48** | **3.73** | **86.95** | **80.00** | **3.77** | **98.45** |

Table 3. **Ablation Study.** PD alone has minimal impact on performance. Adding DSR alone provides a slight improvement in both performance and uniformity. Combining both DSR and PD leads to a significant boost in uniformity and achieves the best performance.

| | LP | FT 1% | FT 10% | $-\mathcal{L}_{\text{uniform}}(\uparrow)$ |
|---|---|---|---|---|
| FedAlignUniform [31] | 80.84 | 69.99 | 81.00 | 3.79 |
| +PD | 80.74 | 69.78 | 80.71 | 3.80 |
| +DSR | 81.05 | 69.77 | 81.15 | 3.81 |
| + DSR + PD (**SSD**) | **81.32** | **70.74** | **81.67** | **3.84** |

This demonstrates that SSD effectively enhances representation quality, outperforming existing methods.

**Generalization on OOD datasets.** Evaluating generalization to out-of-distribution (OOD) datasets helps assess whether models can learn transferable representations that perform well beyond their training distribution. Table 2 presents the generalization performance of different methods, where models trained on CIFAR-100 and TinyImageNet-200 are evaluated on CIFAR-10.

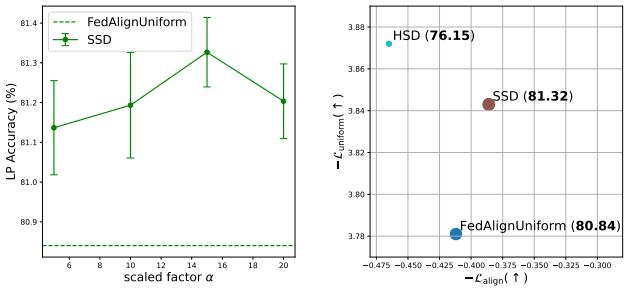
Among state-of-the-art methods, FedX, which exhibits high representation quality but performs poorly in in-distribution settings, achieves the best generalization in OOD scenarios. However, our method, SSD, consistently outperforms all baselines, achieving the highest accuracy in both settings. SSD also demonstrates the best uniformity [31] and effective rank [7], indicating superior feature representation quality. While FedAlignUniform, FedDecorr, L-DAWA and FedU2 perform competitively, they generally fall short of SSD, further reinforcing its effectiveness in handling distribution shifts.

## 4.2. Analysis

**Ablation Study.** We provide the ablation study in Table 3. It shows that adding projector distillation (PD) alone does not provide noticeable improvements. In contrast, adding dimension-scaled regularization (DSR) alone leads to slight improvements in LP and FT 10%, but the uniformity enhancement is minimal. However, when combining both DSR and PD (SSD), the method effectively transfers optimization benefits to representation quality, leading to a significant uniformity improvement and achieving the best overall performance.

**Robustness of DSR.** We evaluate our method from two perspectives: the scaled factor $\alpha$ and the selection of scaled dimensions for each client. The scaled factor $\alpha$ influences the overall discrepancy, while the selection of scaled dimensions affects the direction each client is guided toward. To assess robustness, each $\alpha$ is evaluated with three different selections of scaled dimensions. As shown in Figure 6a, our method maintains stable performance across different $\alpha$ values, introducing minimal variation while consistently outperforming FedAlignUniform. This demonstrates that our approach effectively balances alignment and uniformity without being overly sensitive to the choice of $\alpha$ or the specific dimension selection.

**Soft *vs*. hard client separation.** We examine the impact of client feature separation on alignment, uniformity, and overall performance. Our intuition is that increasing separation between clients enhances global uniformity. One way to enforce this separation is by restricting each client to its own subspace without sharing information with others. To illustrate the effects of such strict partitioning, we consider Hard Separation and Distillation (HSD) as a baseline. As shown in Figure 6b, while HSD achieves the highest uniformity, it does so at the cost of severely reduced alignment, ultimately leading to poor downstream performance. This performance drop can be attributed to the disruption of intrinsic feature structures. In contrast, our proposed SSD balances




(a) Robustness (b) Separation

Figure 6. (a) **Robustness of DSR** across different scaled factors $\alpha$, where each $\alpha$ is tested with three different randomly selected scaled dimensions. (b) **Soft *vs*. hard separation.** HSD achieves the highest uniformity but at the cost of reduced alignment, resulting in poor overall performance.

Table 4. **The Effect of the Projector.** Without the projector, DSR significantly improves uniformity and performance. Yet, the overall results remain inferior to those achieved with the projector.

| | Projector | LP | $-\mathcal{L}_{\text{uniform}}(\uparrow)$ |
|---|---|---|---|
| FedAlignUniform | ✗ | 73.16 | 3.72 |
| + DSR | ✗ | 76.14 (+2.98) | 3.77 (+0.05) |
| FedAlignUniform | ✓ | 80.84 | 3.79 |
| + DSR | ✓ | 81.05 (+0.21) | 3.81 (+0.02) |

both alignment and uniformity, mitigating the drawbacks of hard separation while enhancing uniformity, resulting in improved overall performance.

**Why not remove projector or apply loss on representations?** A natural idea for addressing the limited transfer of embedding-level loss optimization (in terms of representation uniformity) is to remove the projector or apply the loss directly to the representations. However, prior work [2, 9, 32] has demonstrated that the projector plays a critical role in preventing the encoder from overfitting on the upstream task. Indeed, as shown in Table 4, removing the projector and adding DSR does substantially improve uniformity and downstream performance, but these results still lag behind those achieved when the projector is retained. Therefore, it is crucial to develop a method that effectively promotes uniformity while preserving the benefits offered by the projector.

# 5. Related Work

## 5.1. Federated Learning with Non-IID Data

Federated learning (FL) enables collaborative model training across multiple clients without centralizing data. Since client data distributions are typically non-IID, model convergence and performance are challenging. FedAvg [24], which aggregates local model updates using simple averaging, is the foundational framework but suffers under heterogeneous data and imbalanced client participation.

To address these issues, various aggregation improvements have been proposed. FedProx [21] reduces client drift by adding a proximal term to limit local updates from straying from the global model. SCAFFOLD [16] corrects client drift using control variates to reduce variance in local updates. MOON [19] improves representation consistency with a contrastive loss. Other methods use globally shared data to improve generalization; FedShare [35] aligns client models using shared data, while FedDistill [13] distills knowledge from the global model to local models using shared data.

## 5.2. Federated Unsupervised Learning

Federated Unsupervised Learning (FUL) combines federated learning with un-/self-supervised learning, primarily addressing (1) non-IID data distribution and (2) representation collapse.

To handle non-IID data, FedCA [33] uses a shared dictionary module for better aggregation but risks privacy leakage. FedU [36] reduces this risk by selectively uploading the online network's encoder and deciding predictor updates based on divergence. FedEMA [37] extends this with exponential moving average updates. FedX [10] adds an alignment/contrastive term with the global model, while Orchestra [23] preserves global structural consistency using global centroids. FedU2 [22] ensures balanced updates across clients for better alignment but does not guarantee global representation uniformity under non-IID settings.

For representation collapse, FedDecorr [28] shows that local clients suffer from dimensional collapse, which propagates to the global model, and addresses it with a local decorrelation loss. FedU2 [22] encourages local representation uniformity by minimizing divergence with a spherical Gaussian. Our work extends this by showing that intra-client uniformity alone is insufficient inter-client uniformity must also be explicitly addressed.

# 6. Conclusion

We introduce Soft Separation & Distillation (SSD), a framework designed to enhance representation quality by improving inter-client uniformity in federated learning. SSD consists of a dimension-scaled regularization term that softly separates client embeddings while preserving the intrinsic data structure, and a projector distillation term that transfers the optimization benefits of the projector to the encoder, thereby improving representation quality. SSD achieves state-of-the-art performance in both representation learning and downstream tasks across various training and FL settings. Our work highlights the importance of global representation quality in federated unsupervised learning, opening new directions for future research.

## References


[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020. 1

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 1, 2, 5, 8, 11

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 1

[4] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007. 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[6] Xianghong Fang, Jian Li, Qiang Sun, and Benyou Wang. Rethinking the uniformity metric in self-supervised learning. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 6

[7] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International conference on machine learning*, pages 10929–10974. PMLR, 2023. 2, 6, 7

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1

[9] Kartik Gupta, Thalaiyasingam Ajanthan, Anton van den Hengel, and Stephen Gould. Understanding and improving the role of projection head in self-supervised learning. *arXiv preprint arXiv:2212.11491*, 2022. 2, 5, 8

[10] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *European Conference on Computer Vision*, pages 691–707. Springer, 2022. 2, 6, 7, 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 6, 11

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[13] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018. 8

[14] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. 2

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2): 1–210, 2021. 1

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 8

[17] Taehyeon Kim, Jaehoon Oh, Nak Yil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2628–2635. International Joint Conferences on Artificial Intelligence Organization, 2021. Main Track. 11

[18] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016. 1

[19] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 8

[20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020. 1

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 2, 8

[22] Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Fengyuan Yu, Huabin Zhu, Binhui Yao, Tao Wang, Xiaolin Zheng, and Yanchao Tan. Rethinking the representation in federated unsupervised learning with non-iid data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22841–22850, 2024. 2, 6, 7, 8

[23] Ekdeep Lubana, Chi Ian Tang, Fahim Kawsar, Robert Dick, and Akhil Mathur. Orchestra: Unsupervised federated learning via globally consistent clustering. In *Proceedings of the 39th International Conference on Machine Learning*, pages 14461–14484. PMLR, 2022. 8

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 6, 8

[25] Yasar Abbas Ur Rehman, Yan Gao, Pedro Porto Buarque De Gusmão, Mina Alibeigi, Jiajun Shen, and Nicholas D Lane. L-dawa: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16464–16473, 2023. 6, 7

[26] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22 (3):400 – 407, 1951. 11

[27] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007. 1, 6

[28] Yujun Shi, Jian Liang, Wenqing Zhang, Vincent Tan, and Song Bai. Towards understanding and mitigating dimensional collapse in heterogeneous federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 6, 7, 8

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[30] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021. 3, 6

[31] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020. 1, 2, 3, 6, 7, 11

[32] Yihao Xue, Eric Gan, Jiayi Ni, Siddharth Joshi, and Baharan Mirzasoleiman. Investigating the benefits of projection head for representation learning. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 5, 8

[33] Fengda Zhang, Kun Kuang, Long Chen, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Fei Wu, Yueting Zhuang, et al. Federated unsupervised representation learning. *Frontiers of Information Technology & Electronic Engineering*, 24(8):1181–1193, 2023. 8

[34] Yifei Zhang, Hao Zhu, Yankai Chen, Zixing Song, Piotr Koniusz, and Irwin King. Mitigating the popularity bias of graph collaborative filtering: A dimensional collapse perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 6

[35] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 8

[36] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4912–4921, 2021. 2, 6, 8

[37] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *International Conference on Learning Representations*, 2022. 2, 6, 8

# A. Experimental Details

## A.1. Training

We follow the image augmentations used in SimCLR [2] and adopt ResNet-18 [11] as the encoder, coupled with a two-layer linear projector. The model is trained for 5 epochs over 100 communication rounds with a batch size of 128. Both the encoder and projector produce output representations of 512 dimensions. Optimization is performed using SGD [26] for both local and global models, with a learning rate of 0.1. The hyperparameters $\beta, \gamma, \delta$ are set to 1.0, 1.0, and 0.1, respectively. The scaling factor $\alpha$ is set to 10.

## A.2. Effective Rank

**Definition 1** (Effective Rank)**.** Let matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times d}$ with $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ as its singular value decomposition, where $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_Q \geq 0$ with $Q = \min(N, d)$. The distribution of singular values is defined as the normalized form $p_i = \sigma_i / \sum_{k=1}^{Q} |\sigma_k|$. The effective rank of the matrix $\boldsymbol{Z}$, is defined as

$$\mathrm{ERank}(\boldsymbol{Z}) = \exp\left(H\left(p_1, p_2, \cdots, p_Q\right)\right) \tag{11}$$

where $H\left(p_1, p_2, \cdots, p_Q\right)$ is the Shannon entropy $H\left(p_1, p_2, \cdots, p_Q\right) = -\sum_{k=1}^{Q} p_k \log p_k$.

# B. Additional Experiments

**Distillation methods.** We compare two projector distillation methods, MSE and KL divergence, as studied in [17]. The results indicate that both methods achieve similar performance and consistently outperform the baseline.

Table 5. **Distillation methods.** Both MSE and KL divergence for PD achieve comparable performance and uniformity.

| | LP | FT 1% | FT 10% | $-\mathcal{L}_{\text{uniform}}(\uparrow)$ |
|---|---|---|---|---|
| FedAlignUniform [31] | 80.84 | 69.99 | 81.00 | 3.79 |
| SSD (MSE) | **81.88** | 70.61 | 81.62 | 3.83 |
| SSD (KL) | 81.32 | **70.74** | **81.67** | **3.84** |