Collaborative Unsupervised Visual Representation Learning from Decentralized Data

Weiming Zhuang[1,3] Xin Gan[2] Yonggang Wen[2] Shuai Zhang[3] Shuai Yi[3]
[1]S-Lab, Nanyang Technological University, [2]Nanyang Technological University, [3]SenseTime Research
{weiming001,ganx0005}@e.ntu.edu.sg, ygwen@ntu.edu.sg, {zhangshuai,yishuai}@sensetime.com

Abstract

Unsupervised representation learning has achieved outstanding performances using centralized data available on the Internet. However, the increasing awareness of privacy protection limits sharing of decentralized unlabeled image data that grows explosively in multiple parties (e.g., mobile phones and cameras). As such, a natural problem is how to leverage these data to learn visual representations for downstream tasks while preserving data privacy. To address this problem, we propose a novel federated unsupervised learning framework, FedU. In this framework, each party trains models from unlabeled data independently using contrastive learning with an online network and a target network. Then, a central server aggregates trained models and updates clients' models with the aggregated model. It preserves data privacy as each party only has access to its raw data. Decentralized data among multiple parties are normally non-independent and identically distributed (non-IID), leading to performance degradation. To tackle this challenge, we propose two simple but effective methods: 1) We design the communication protocol to upload only the encoders of online networks for server aggregation and update them with the aggregated encoder; 2) We introduce a new module to dynamically decide how to update predictors based on the divergence caused by non-IID. The predictor is the other component of the online network. Extensive experiments and ablations demonstrate the effectiveness and significance of FedU. It outperforms training with only one party by over 5% and other methods by over 14% in linear and semi-supervised evaluation on non-IID data.

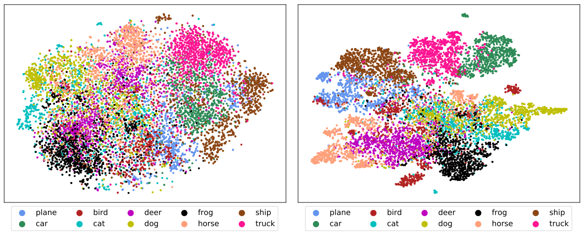
1. Introduction

Learning good visual representations without supervision is attracting considerable attention in recent years. These visual representations can facilitate efficient training of downstream tasks [29, 24] like image segmentation [19]. Researchers have proposed many unsupervised representation learning methods by designing pretext tasks [5, 37, 22, 6]. Among them, contrastive learning [9, 23] based on instance-level discrimination has achieved the state-of-the-art performance [33, 1, 10, 8, 2]. These unsupervised representation learning methods rely on the assumption that data can be collected and stored in a centralized database, such as images from the Internet.

However, in real-world scenarios, decentralized image data are growing explosively and the data collected in multiple parties may not be centralized due to data privacy regulations [3]. For example, photos taken on phones or images collected from street cameras could contain sensitive information. Centralizing them for training could reveal the identity and locality of individuals [42]. Besides, compared with training representations using only publicly available data from the Internet, representations learned from the actual data may be more representative for downstream tasks applied on similar scenarios. Utilizing the decentralized unlabeled image data to learn representations with privacy guaranteed is an important but overlooked problem.

Existing methods cannot leverage decentralized unlabeled data to learn a generic representation while preserving data privacy. Federated learning is an emerging distributed training technique [21] that have empowered multiple parties to collaboratively learn computer vision models, such as segmentation [18, 28], object detection [20], and person re-identification [42]. But traditional federated learning relies on data labels. Decentralized data is normally non-independent and identically distributed (non-IID), which is one of the key challenges for learning from multiple parties [38, 16]. For example, in real-world street datasets [20], cameras capture diverse two or three out of seven object categories. The state-of-the-art unsupervised learning approaches are effective, but they may not work well with non-IID data, as shown in Figure 1(a). Although Federated Contrastive Averaging (FedCA) [36] addresses the unlabeled and non-IID problems, it imposes potential privacy leakage risks by directly sharing features of clients' data.

In this paper, we propose a new federated unsupervised representation learning framework, *FedU*, to learn generic



(a) Single Client Training (b) Our Proposed FedU

Figure 1. Limitations of unsupervised learning (BYOL [8]) on non-IID data. (a) and (b) are t-SNE visualization of representations. We simulate five decentralized clients, where each client has unique two out of ten classes of CIFAR-10 [15] training data. Training in a single client cannot learn the representation properly.

visual representations collaboratively from decentralized unlabeled data in multiple parties while preserving data privacy. Built on contrastive learning for unlabeled data in each party [8], we propose to centralize the learned representations instead of raw data to a central server to protect data privacy. Each party uses Siamese networks in training: an online network with an online encoder and a predictor; a target network with a target encoder. FedU is not a trivial combination of contrastive learning and federated learning, because we implement two simple but effective methods to tackle the non-IID data problem. As studied in [38], non-IID data causes weight divergence, resulting in performance degradation. Through analyzing the characteristics of the Siamese networks and the impact of non-IID data, we first design a communication protocol to upload only the online encoders of parties for server aggregation and update them with the aggregated global encoder for the next training round. Moreover, we introduce a new module, divergence-aware predictor update (DAPU), to dynamically decide the choice of predictor update based on the degree of divergence. It updates the predictors in multiple parties with the aggregated predictor only when the divergence is smaller than a threshold.

We evaluate FedU on CIFAR datasets [15] using various backbones and settings. Extensive experiments demonstrate that FedU achieves promising results under three evaluation protocols: linear evaluation, semi-supervised learning, and transfer learning. Compared with existing methods, FedU achieves superior performance on all evaluation protocols. Specifically, the representation learned from FedU is much better than the one learned from only one party (Figure 1). It outperforms training with one party by over 5% and other methods by more than 14% under linear and semi-supervised evaluations on non-IID data. We also present ablations to illustrate the intuition and performance of FedU.

In summary, the contributions of the paper are:

- We introduce a new framework, *FedU*, to address an important but overlooked problem: leveraging unlabeled data from multiple parties to learn visual representations while preserving data privacy.
- We design the communication protocol to only aggregate and update the online encoders by analyzing the impact of non-IID data on Siamese networks.
- We propose a new module, divergence-aware predictor update (DAPU), to dynamically determine how to update predictors based on the degree of divergence caused by non-IID data.
- Extensive experiments and ablations demonstrate the effectiveness and significance of FedU.

2. Related Work

2.1. Unsupervised Representation Learning

The majority of unsupervised representation learning methods fall into two categories: generative and discriminative. Generative methods learn representations by generating pixels mapping to the input space via methods like auto-encoding [31, 13] or adversarial learning [7, 27]. Discriminative methods learn representations by performing proxy tasks, such as image in-painting [26] and solving jigsaw puzzles [22]. Among discriminative approaches, contrastive learning is the state-of-the-art method [33, 1, 10, 8, 2]. Contrastive learning [9, 23] aims to minimize the similarity of positive samples (two different augmentations of the same image), while maximizing the similarity of negative samples (two different images). The negative pairs are either generated from a memory bank like MoCo [10] or from a large batch size like SimCLR [1]. Methods like BYOL [8] and SimSiam [2] even bypass negative pairs and contrast only positive pairs. However, these methods do not perform well with non-IID data. In our proposed FedU, we introduce two methods to tackle the non-IID data challenge.

2.2. Federated Learning

Federated Learning (FL) is an emerging distributed training method that coordinates decentralized clients to train machine learning models [21]. FedAvg [21] is the standard algorithm for FL. Non-IID data is one of the key challenges of FL, which causes weight divergence (Figure 3) and performance drop as discussed in [38]. Many approaches have been proposed to address this challenge, such as sharing a public dataset [38], knowledge distillation [42], or regularizing training in clients [17]. However, researchers investigate these methods under supervised learning, not directly applicable to scenarios where data is unlabeled.

Although several existing works study federated learning with unlabeled data [40, 41], they mainly focus on specific applications. Other methods either bypass the non-IID

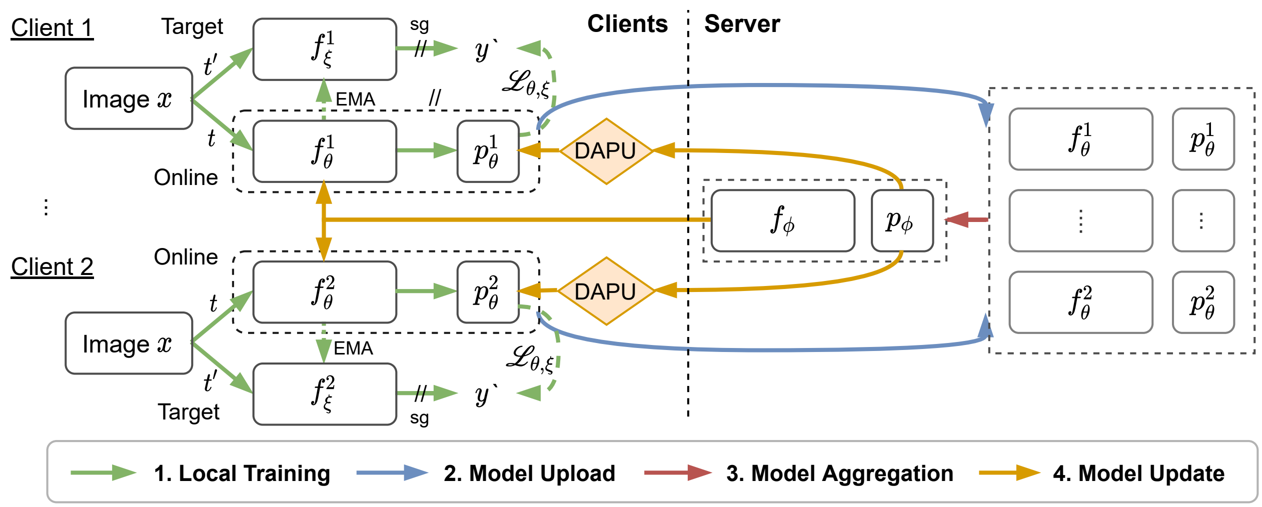


Figure 2. Overview of our proposed federated unsupervised representation framework (FedU) that trains representations collectively with multiple clients coordinated by a central server. Each training round of FedU consists of four stages: (1) **Local training**: each client k trains an online encoder f_θ^k and a predictor p_θ^k with contrastive loss and updates a target encoder f_ξ^k with exponential moving average (EMA). (sg means stop gradient.) (2) **Model Upload**: Each client k uploads f_θ^k and p_θ^k to the server. (3) **Model Aggregation**: The server aggregates all f_θ^k and p_θ^k to obtain a new global encoder f_ϕ and predictor p_ϕ. (4) **Model Update**: The server distributes and updates clients' local encoders f_θ with f_ϕ. The clients update their predictors p_θ based on our proposed divergence-aware predictor update (DAPU).

problem [30, 12] or impose potential privacy risks [36]. Specifically, FedCA [36] solves the non-IID problem by gathering features and data distribution from clients, compromising data privacy from potential attacks. Our proposed FedU is simpler, achieves better performance, and effectively preserves data privacy.

3. Methodology

In this section, we first define the problem and then introduce our proposed federated unsupervised representation learning framework (FedU) to address the problems.

3.1. Problem Definition

Before presenting the details of FedU, we define the problem and the assumptions first. Several parties aim to learn a generic representation f_ϕ for various downstream tasks without sharing data among these parties. We denote each party as a client k containing unlabeled data $\mathcal{D}_k = \{\mathcal{X}_k\}$. The global objective function is $\min_\phi h(\phi) := \sum_{k=1}^{N} p_k F_k(\phi)$, where N is the number of clients, $p_k = \frac{n_k}{n}$, and $n = \sum_k n_k$ is the total data size. For client k, $F_k(\phi) := \mathbb{E}_{x_k \sim \mathcal{D}_k}[h_k(\phi; x_k)]$ is the expected loss over data distribution $\mathcal{D}_k$, where x_k is the data and $h_k(\phi; x_k)$ represents the loss function to train models.

A key challenge is that data among decentralized clients are likely to be non-IID. For example, each client could contain only two out of seven object categories in real-world street datasets [20]. As discussed in [38], non-IID data causes weight divergence (illustrated in Figure 3). Training with non-IID data in one client using existing unsupervised learning methods could result in a poor representation, as shown in Figure 1(a). Besides, we cannot centralize data from clients to construct a big dataset due to privacy constraints. In this paper, we aim to leverage the growing amount of unlabeled image data from multiple parties to learn a generic representation f_ϕ without privacy leakage.

3.2. FedU Overview

Figure 2 presents the overview of our proposed framework, FedU, which tackles the problem defined above. FedU instructs a server to coordinate multiple clients with unlabeled data to train a generic representation f_ϕ. It follows the proposed communication protocol to upload the online encoder f_θ for server aggregation and update them with the global encoder f_ϕ. Besides, FedU introduces a new divergence-aware predictor update (DAPU) module to address the non-IID data challenge.

Before presenting technical details, we introduce each round's training pipeline of FedU with following stages: (1) *Local training*: each client k conducts unsupervised representation learning with contrastive loss (Equation 1), obtaining an online network with an online encoder f_θ^k and a predictor p_θ^k, as well as a target network with a target encoder f_ξ^k. (2) *Model Upload*: Each client k uploads the online encoder f_θ^k and the predictor p_θ^k to the server. (3) *Model Aggregation*: The server aggregates clients' online encoders and predictors to obtain a new global encoder with $f_\phi = \sum_{k=1}^{N} \frac{n_k}{n} f_\theta^k$ and a new global predictor with

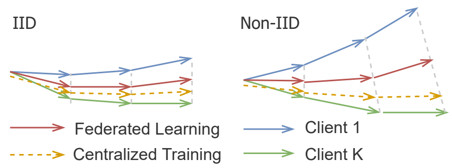


Figure 3. Illustration of weight divergence caused by non-IID data studied in [38]. Federated learning on non-IID data leads to significant divergence from the centralized training.

$p_\phi = \sum_{k=1}^{N} \frac{n_k}{n} p_\theta^k$, where n_k is the data volume of client k and n is the total data volume of N clients. (4) *Model Update*: The server sends the global encoder f_ϕ and predictor p_ϕ to all clients. Each client updates its online encoder f_θ with the global encoder f_ϕ and uses DAPU to dynamically decide whether updating local predictor p_θ with the global predictor p_ϕ (Equation 3). We summarize FedU in Algorithm 1. Next, we present the details of local training, the communication protocol, and DAPU.

3.3. Local Training

In local training, each client conducts contrastive learning with asymmetric Siamese networks: an online network and a target network, adopted from BYOL [8]. Traditional federated learning only needs one network in each client to perform supervised training. However, as data is unlabeled, FedU requires two networks to generate positive pairs from two augmentations of an image for contrastive learning. The online network consists of an online encoder f_θ and a predictor p_θ. The target network only contains a target encoder f_ξ. f_θ and f_ξ share the same architecture but differ in parameters.

Local training starts from taking two augmentations t and $t^{'}$ of the input image x. They are the inputs of the online and target networks, respectively. The role of the online network is to learn from the unlabeled data. We update its parameters θ with contrastive loss:

$$\mathcal{L}_{\theta,\xi} \triangleq \left\| y - y^{'} \right\|_2^2 \triangleq 2 - 2 \cdot \frac{\langle y, y^{'} \rangle}{\|y\|_2 \cdot \|y^{'}\|_2}, \tag{1}$$

where $y \triangleq p_\theta(f_\theta(t))$ is the output of the online network and $y^{'} \triangleq f_\xi(t^{'})$ is the output of the target network. The role of the target network is to generate a positive regression target for the online network to contrast. Instead of updating with gradient descent, its parameters ξ are updated with the exponential moving average (EMA) of the online encoder's parameters θ in every batch:

$$\xi = m\xi + (1-m)\theta, \tag{2}$$

where $m \in [0, 1]$ is the decay rate.

Each client trains E local epochs and then uploads model updates to the server. The communication protocol between the server and clients requires careful considerations to mitigate the adverse impact of non-IID data.

Algorithm 1: Federated Unsupervised Representation Learning Framework (FedU)

Input: learning rate η, threshold μ, total training rounds R, E, B, K, N, n_k, n
Output: f_ϕ

Server:
  Initialize the global encoder f_ϕ^0 and predictor p_ϕ^0;
  for *each round r = 0 to R-1* **do**
    $S_t \leftarrow$ (random selection of K clients);
    for *client* $k \in S_t$ *concurrently* **do**
      $f_\theta^k, p_\theta^k \leftarrow$ **Client**(f_ϕ^r, p_ϕ^r, r);
    end
    // Model aggregation;
    $f_\phi^{r+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} f_\theta^k$;
    $p_\phi^{r+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} p_\theta^k$;
  end
  return f_ϕ^R;
Client (f_ϕ, p_ϕ, r)**:**
  // Model update;
  $f_\theta \leftarrow f_\phi$;
  // Update using DAPU with Equation 3;
  if $\left\|\theta^r - \phi^{r-1}\right\|_2^2 < \mu$ **then**
    $p_\theta \leftarrow p_\phi$;
  end
  $\mathcal{Z} \leftarrow$ (construct data batches with batch size B);
  // Local training with Equation 1 and 2;
  for *local epoch e = 0 to E-1* **do**
    for $z \in \mathcal{Z}$ **do**
      $\theta \leftarrow \theta - \eta \nabla \mathcal{L}_{\theta,\xi}(\theta; z)$;
      $\xi \leftarrow m\xi + (1-m)\theta$;
    end
  end
  return f_θ, p_θ // Model upload;
return

3.4. Communication Protocol

FedU requires bidirectional communication between the server and the clients. In the model upload stage, clients send models to the server for aggregation. In the model update stage, the server distributes the aggregated model to clients and updates clients' local models. As FedU performs local training with two models of the same architecture, it leads to important design decisions on the communication protocol: (1) Which encoder (online or target) to *upload* for aggregation? (2) Which encoder (online, target, or both) to *update* with the aggregated encoder from the server?

We analyze the characteristics of both encoders and hypothesize that it is desirable to aggregate and update the *online encoder*. The target encoder is the exponential averaging of the online encoder, denoting the historical representation of a client. The online encoder is constantly updated from back-propagation in every training step, representing the latest representation of a client. Although both encoders capture local data characteristics, we argue that the latest representation learned in the online encoder is more representative of local data distribution. Thus, uploading the online encoder for server aggregation can better capture the characteristics of non-IID data.

In the model update stage, the online encoder also plays an important role. The server aggregated global encoder f_ϕ is the average of encoders from clients, so it has better generalizability. As the target encoder produces regression targets for the online encoder to contrast, we should not only update the target encoder with a more general global model f_ϕ because it would decrease its capability of providing local-representative targets. Updating both the online and target encoders could work, but it means that the local training of each client needs to adapt the general global model f_ϕ to non-IID local data again. Hence, we only update the online encoder f_θ with the global encoder f_ϕ while keeping the parameters of the target encoder for stable regression targets. Our ablation study of the online and target encoders (Table 4) validates our hypothesis — using the online encoder for aggregation and update.

3.5. Divergence-aware Predictor Update

Apart from considering the encoders for the communication protocol, another vital design choice is whether clients should update the predictors with the global predictor in every training round. Inspired by [38] that non-IID data causes weight divergence (Figure 3), we propose a new module, divergence-aware predictor update (DAPU), to dynamically decide whether updating the local predictor p_θ with the aggregated predictor p_ϕ. We make the decision based on the degree of divergence and formulate it as:

$$p_\theta = \begin{cases} p_\phi & \left\|\theta^r - \phi^{r-1}\right\|_2^2 < \mu \\ p_\theta & \text{otherwise} \end{cases} \tag{3}$$

where μ is a controllable threshold. θ^r and ϕ^{r-1} represents the parameters of the online encoder in round r and the global encoder in round $r-1$ respectively. As local encoder parameters θ^{r-1} is updated with the global encoder parameters ϕ^{r-1}, $\left\|\theta^r - \phi^{r-1}\right\|_2^2$ measures the divergence of model parameters occurred in local training.

The intuition of DAPU is that clients update the predictors with the local predictor p_θ when divergence is large and update them with the global predictor p_ϕ when divergence is small. The predictor is the last layer of the online network. Based on the study of characteristics of layers of convolutional neural network [34], the last layer captures information that is most related to specific classes and objects in the dataset. Since local data in clients is non-IID, the predictors p_θ of clients could have large variances. Simply updating predictors in clients with the global predictor p_ϕ could have side effects on learning. While on the other hand, always updating with local predictors limits clients' generalizability. Hence, we propose to dynamically update it with global predictor p_ϕ only when the divergence is small.

4. Experimental Evaluation

In this section, we evaluate the performance of the representation f_ϕ learned by FedU on CIFAR-10 and CIFAR-100 [15]. We first explain the experiment setup. Then we assess the representation in linear evaluation, semi-supervised learning, and transfer capabilities to other datasets.

4.1. Experiment Setup

Datasets and Federated Simulations For linear and semi-supervised evaluation, we use CIFAR-10 and CIFAR-100 [15] datasets. Both contain 50,000 training images and 10,000 testing images. CIFAR-10 and CIFAR-100 contain 10 classes and 100 classes with an equal number of images per class, respectively. For transfer learning evaluation, we train on the Mini-ImageNet dataset [32]. Mini-ImageNet contains 60,000 images in 100 classes extracted from ImageNet [4]. To simulate K clients, we divide training sets into K partitions. For IID simulation, each client contains an equal number of images of all classes. For non-IID simulation, each client contains $\frac{10}{K}$ CIFAR-10 classes and $\frac{100}{K}$ CIFAR-100/Mini-ImageNet classes.

Implementation Details We implement FedU in Python using EasyFL [39] based on PyTorch [25] framework. We simulate K clients of training using K NVIDIA® V100 GPUs, one GPU for one client. The server and clients communicate through PyTorch communication backend. We use ResNet-18 and ResNet-50 [11] as the network architecture for the encoders and use a multi-layer perceptron (MLP) as the predictor. For fair comparison with other methods, we run experiments with $K = 5$ clients for $R = 100$ training rounds, where each client performs $E = 5$ local epochs in each round. We use the threshold $\mu = 0.4$ and $\mu = 0.6$ for CIFAR-10 and CIFAR-100 experiments, respectively. For ablation studies, each client conducts $E = 1$ local epochs in each round for $R = 800$ rounds. We use decay rate $m = 0.99$, batch size $B = 128$, and SGD as optimizer with learning rate $\eta = 0.032$.

4.2. Linear Evaluation

We evaluate the representation learned from FedU using linear evaluation on CIFAR datasets, following the linear evaluation protocol described in [14, 8]: We first train a model without supervision using FedU and other baseline

Method	Architecture	Param.	CIFAR-10		CIFAR-100	
			IID	Non-IID	IID	Non-IID
Single client training	ResNet-18	11M	81.24	71.98	51.33	49.69
Single client training	ResNet-50	23M	83.16	77.84	57.21	55.16
FedSimCLR [1] [36]	ResNet-50	23M	68.10	64.06	39.75	38.70
FedCA [36]	ResNet-50	23M	71.25	68.01	43.30	42.34
FedSimSiam [2]	ResNet-50	23M	79.64	76.70	46.28	48.80
FedU (ours)	ResNet-18	11M	85.21	78.71	56.52	57.08
FedU (ours)	ResNet-50	23M	**86.48**	**83.25**	**59.51**	**61.94**
Upper-bound methods: centralized unsupervised learning and supervised federated learning						
FedAvg [21] (Supervised)	ResNet-50	23M	91.51	67.74	65.77	64.38
BYOL [8] (Centralized)	ResNet-50	23M	91.85	-	66.51	-

Table 1. Top-1 accuracy (%) comparison under the linear evaluation protocol on IID and non-IID settings of CIFAR datasets. Our proposed FedU outperforms other methods. It even outperforms supervised federated learning (FedAvg) on the non-IID setting of CIFAR-10 dataset.

Method	Architecture	Param.	CIFAR-10				CIFAR-100			
			IID		Non-IID		IID		Non-IID	
			1%	10%	1%	10%	1%	10%	1%	10%
Single client training	ResNet-18	11M	74.76	78.08	60.25	70.60	26.32	43.05	21.95	37.70
Single client training	ResNet-50	23M	74.80	80.33	63.65	74.30	25.91	45.29	23.18	41.43
FedAvg [21] (Super.)	ResNet-50	23M	26.68	40.44	17.72	21.69	8.09	5.37	14.47	13.98
FedSimCLR [1] [36]	ResNet-50	23M	50.00	60.67	26.03	33.83	23.01	31.56	14.02	20.01
FedCA [36]	ResNet-50	23M	50.67	61.02	28.50	36.28	23.32	32.09	16.48	22.46
FedU (ours)	ResNet-18	11M	79.40	82.61	68.28	78.52	**31.31**	**47.64**	30.36	**48.80**
FedU (ours)	ResNet-50	23M	**79.44**	**83.08**	**71.19**	**80.08**	29.35	47.14	**30.80**	48.76
BYOL [8] (Centralized)	ResNet-50	23M	89.07	89.66	-	-	41.49	60.23	-	-

Table 2. Top-1 accuracy (%) comparison under semi-supervised protocol using 1% and 10% of CIFAR datasets for fine-tuning. FedU outperforms other methods except the upper-bound — centralized unsupervised learning (BYOL).

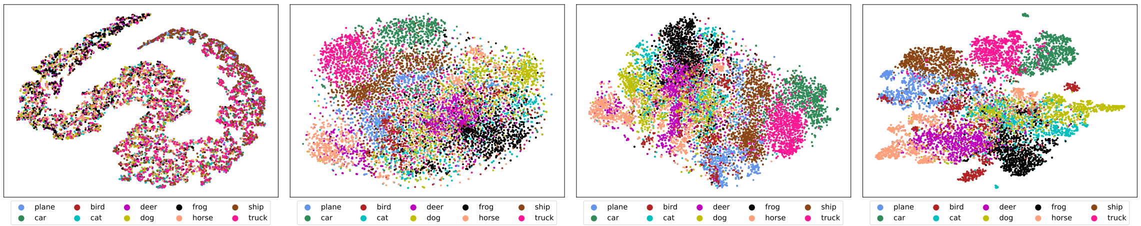
methods for 100 epochs; Next, we froze the model parameters of the backbone and train a new classifier on top of it for another 100 epochs. The following are the compared methods: (1) *Single Client Training*: each client learns a representation with their local data for 500 epochs using BYOL [8]; (2) *FedSimCLR*: simply combine federated learning and SimCLR [1] from paper [36]; (3) *FedSimSiam*: combine federated learning with SimSiam [2] (use SimSiam for local training instead of our method); (4) *FedCA*: the method proposed in [36]. All the experiments are conducted under the same settings. Besides, we also compare FedU with two potential upper bound methods: centralized unsupervised learning using BYOL [8] and supervised federated learning with FedAvg [21].

Table 1 reports the performance of these methods with different backbones, datasets, and under both IID and non-IID settings. It shows that FedU achieves better performance than other methods on linear evaluation. Specifically, it outperforms the existing method FedCA [36] by at least 14% and outperforms the single client training by more than 5% on non-IID settings. Compared with the theoretical upper-bound methods BYOL and FedAvg, FedU outperforms them on non-IID CIFAR-10 data. Besides, the results show that the performance of FedU increases using deeper backbones (ResNet-50 vs. ResNet-18 [11]).

4.3. Semi-Supervised Learning

We evaluate the representation learned from FedU on the semi-supervised protocol described in [35, 1], targeting the federated scenarios that only a small subset of data are labeled. We consider two semi-supervised learning settings: 1% or 10% are labeled. We first obtain the representations using FedU and other methods without the labeled data. Then, instead of fixing the model, we fine-tune the whole model with an additional new classifier in the semi-supervised protocol using the labeled data for 100 epochs. The methods compared are similar to the ones defined in the linear evaluation section.

As reported in Table 2, FedU outperforms other methods except centralized training on semi-supervised evaluation protocol. Supervised federated learning with FedAvg using only 1% or 10% of data has poor performance. On evaluation using the CIFAR-10 dataset, it outperforms FedCA [36] by more than 22% on IID data and more than 40% on



(a) Agg. Online, Update Target (b) Agg. Target, Update Online (c) Agg. Online, Update Online (d) FedU

Figure 4. T-SNE visualization of representations learned from four methods: (a) Aggregate the online encoder and update the target encoder; (b) Aggregate the target encoder and update the online encoder; (c) Aggregate and update the online encoder; (d) Our proposed FedU. (a), (b), and (c) always use the local predictor, while (d) uses DAPU to dynamically update the predictor. Aggregating and updating with the online encoder (c) achieves better clustering results than (a) and (b). FedU with DAPU (d) further improves the results.

Method	CIFAR-10		CIFAR-100	
	IID	Non-IID	IID	Non-IID
Random Init.	93.79	-	70.52	-
Single client training	94.67	94.33	75.25	75.14
FedSimCLR [1] [36]	94.87	93.97	71.85	70.91
FedCA [36]	94.94	94.16	71.98	71.32
FedU (ours)	**95.00**	**94.83**	**75.57**	**75.60**
BYOL [8] (Centralized)	95.37	-	77.02	-

Table 3. Top-1 accuracy (%) comparison under the transfer learning protocol: transferring from Mini-ImageNet to CIFAR datasets using ResNet-50. FedU outperforms other baseline methods.

Aggregate	Update	Accuracy (%)	
		Global Pred.	Local Pred.
Online	**Online**	**84.07**	**82.18**
Online	Target	9.99	19.22
Online	Both	81.24	18.23
Target	Online	82.10	78.06
Target	Target	9.99	25.02
Target	Both	82.32	29.03

Table 4. Top-1 accuracy comparison of using the *online encoder* or *target encoder* for aggregation and update. Both means updating both encoders. The predictors are updated with either the global or local predictor. Aggregating and updating the online encoder achieves the best performance.

non-IID data. Besides, it consistently outperforms single client training by around 3% regardless of the settings.

4.4. Transfer Learning

We assess the generalizability of representations learned from FedU by evaluating the learned representations on different classification datasets. Specifically, we learn representations on the Mini-ImageNet [32] dataset and evaluate how well they can be transferred to CIFAR datasets. After training, we fine-tune representations on the target datasets for 100 epochs.

Table 3 compares the transfer learning results using ResNet-50 [11]. Even though FedU only slightly outperforms FedCA on the CIFAR-10 dataset because even the random initialization can achieve relatively good performance, it achieves much better performance on the CIFAR-100 dataset, the closest to centralized unsupervised representation learning.

5. Ablation Study

In this section, we conduct ablation studies on the communication protocol, divergence-aware predictor update (DAPU), and hyperparameters of FedU. These ablations give intuitions of FedU's behavior and performance.

5.1. Online Encoder vs. Target Encoder

In Section 3.4, FedU uploads the online encoders for aggregation and updates clients' online encoders with the aggregated global encoder. To empirically verify this hypothesis, we conduct training with twelve combinations of the encoder to upload (online or target encoder), the encoder to update (online, target, or both), and the choice of the predictor (local or global predictor) using ResNet-50 on the CIFAR-10 non-IID setting. Local/global predictor means that clients always update the predictor with parameters of the local/global predictor.

As shown in Table 4, aggregating and updating the online encoder achieves the best performance, regardless of the choice of the predictor. Besides, we compare the t-SNE visualization of representations in Figure 4. Aggregating and updating the online encoder (Figure 4(c)) achieves better clustering results than Figure 4(a) and Figure 4(b). These results further verify our hypothesis on the behaviors and intuitions of encoders in clients. The target encoder provides regression targets for the online encoder, so only updating it results in poor performance. Updating both the online and target encoders achieves competitive results, but it is not comparable to the best performance.

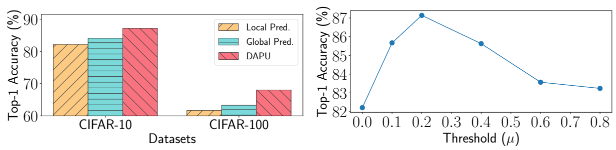
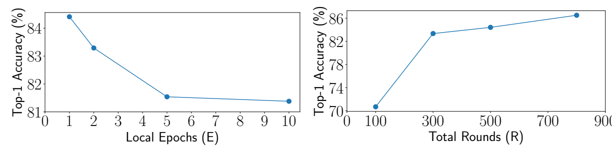


Figure 5. Ablation study on divergence-aware predictor update (DAPU): (a) Compare DAPU with always updating with the local or global predictor; (b) Analysis of threshold μ.



(a) Impact of Local Epochs (b) Impact of Total Rounds

Figure 6. Analysis of (a) impact of local epochs and (b) impact of total training rounds on the performance of FedU.

5.2. Divergence-aware Predictor Update

FedU contains a new module, divergence-aware predictor update (DAPU), to dynamically update clients' predictors with either their local predictors p_θ or the global predictor p_ϕ. To understand the effectiveness of DAPU for non-IID data, we compare DAPU with two static update methods: always updating with the local predictor p_θ and always updating with the global predictor p_ϕ. Besides, we also evaluate the impact of threshold μ. By default, we aggregate and update the online encoders.

Compare with Other Model Update Methods Figure 5(a) demonstrates that DAPU outperforms the other two model update methods by around 5%, under non-IID setting on CIFAR datasets. Moreover, the t-SNE visualization of representations of FedU with DAPU (Figure 4(d)) is better than always updating with the local predictor (Figure 4(c)). These results demonstrate that our proposed DAPU is effective and significant. Next, we present the ablations on the values of threshold μ.

Impact of Threshold Figure 5(b) compares the impact of values of μ on the CIFAR-10 dataset. FedU reaches the best performance when $\mu = 0.2$. As discussed in Section 3.4, the global encoder is the average of online encoders and the online encoders are updated with it in the next round. The divergence between the global encoder ϕ^{r-1} and online encoder θ^r reduces as training proceeds because the encoders obtain higher generalizability and converge gradually. Hence, an optimal threshold μ exists to balance updating with the local predictor p_θ or the global predictor p_ϕ. On the one hand, large μ results in updating with the global predictor too early when the divergence is still significant. On the other hand, small μ results in updating with the global predictor too late. Despite that the impact of divergence varies depending on datasets, these results also hold for the CIFAR-100 dataset (provided in the supplementary).

5.3. Analysis of FedU

Impact of Local Epoch Local epochs E represents the trade-off between accuracy and communication-cost. We measure the impact of different numbers of E by fixing the total computation executed in each client to be 500 epochs. As such, for each training round, each client trains E epochs locally and then transmits the model updates to the server. The total communication rounds are $\frac{500}{E}$. Figure 6(a) illustrates that the accuracy decreases as E increases. Smaller local epoch E achieves better performance but causes higher communication costs. In scenarios that network bandwidth is not bounded, we suggest using $E = 1$ as the default configuration for better performance.

Impact of Training Rounds We investigate the impact of training rounds by fixing $E = 1$ and varies total training rounds R from 100 to 800. Figure 6(b) shows that increasing total training rounds results in better performance. The accuracy improvement is especially significant when the number of total training rounds is small. FedU outperforms other methods with $R = 100$ in Section 4, though the accuracy can be further improved by increasing R.

6. Conclusion

In this work, we introduce a new framework, FedU, to learn generic representations from multiple parties by leveraging decentralized unlabeled data. It includes two simple but effective methods to tackle the non-IID challenge of decentralized data. Firstly, we design the communication protocol to aggregate and update the online encoders. Secondly, we propose a novel module, divergence-aware predictor update (DAPU), to dynamically decide how to update the predictors. We extensively evaluate FedU and conducts ablation studies to illustrate the intuitions and behaviors of the framework. FedU outperforms existing methods on all evaluation protocols. For future work, we will consider applying FedU on larger scale datasets and evaluating specific application scenarios. We hope that FedU will encourage the community to explore learning visual representations from decentralized data under privacy constraints.

Acknowledgements This study is supported by 1) the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s); 2) the National Research Foundation, Singapore, and the Energy Market Authority, under its Energy Programme (EP Award <NRF2017EWT-EP003-023>); 3) Singapore MOE under its Tier 1 grant call, Reference number RG96/20.

References

[1] 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, 6, 7

[2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 1, 2, 6

[3] Bart Custers, Alan M. Sears, Francien Dechesne, Ilina Georgieva, Tommaso Tani, and Simone van der Hof. *EU Personal Data Protection in Policy and Practice*. Springer, 2019. 1

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[5] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 1

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 1

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 1, 2, 4, 5, 6, 7

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 1, 2

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[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. 5, 6, 7

[12] Yilun Jin, Xiguang Wei, Yang Liu, and Qiang Yang. Towards utilizing unlabeled data in federated learning: A survey and prospective. *arXiv e-prints*, pages arXiv–2002, 2020. 3

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[14] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019. 5

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5

[16] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37:50–60, 2020. 1

[17] 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

[18] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 133–141. Springer, 2019. 1

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[20] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Real-world image datasets for federated learning. *arXiv preprint arXiv:1910.11089*, 2019. 1, 3

[21] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. 1, 2, 6

[22] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 1, 2

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2

[24] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014. 1

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[28] Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer, 2018. 1

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[30] Bram van Berlo, Aaqib Saeed, and Tanir Ozcelebi. Towards federated unsupervised representation learning. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 31–36, 2020. 3

[31] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[32] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016. 5, 7

[33] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1, 2

[34] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 5

[35] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 6

[36] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020. 1, 3, 6, 7

[37] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 1

[38] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018. 1, 2, 3, 4, 5

[39] Weiming Zhuang, Xin Gan, Yonggang Wen, and Shuai Zhang. Easyfl: A low-code federated learning platform for dummies. *arXiv preprint arXiv:2105.07603*, 2021. 5

[40] Weiming ZHuang, Xin Gan, Yonggang Wen, and Shuai Zhang. Towards unsupervised domain adaptation for deep face recognition under privacy constraints via federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[41] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 2

[42] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 955–963, 2020. 1, 2