Decentralized Unsupervised Learning of Visual Representations

Yawen Wu[1*] , **Zhepeng Wang**[2] , **Dewen Zeng**[3] , **Meng Li**[4] , **Yiyu Shi**[3] and **Jingtong Hu**[1]
[1]University of Pittsburgh
[2]George Mason University
[3]University of Notre Dame
[4]Facebook
{yawen.wu, jthu}@pitt.edu, zwang48@gmu.edu, {dzeng2, yshi4}@pitt.edu, meng.li@fb.com

Abstract

Collaborative learning enables distributed clients to learn a shared model for prediction while keeping the training data local on each client. However, existing collaborative learning methods require fully-labeled data for training, which is inconvenient or sometimes infeasible to obtain due to the high labeling cost and the requirement of expertise. The lack of labels makes collaborative learning impractical in many realistic settings. Self-supervised learning can address this challenge by learning from unlabeled data. Contrastive learning (CL), a self-supervised learning approach, can effectively learn visual representations from unlabeled image data. However, the distributed data collected on clients are usually not independent and identically distributed (non-IID) among clients, and each client may only have few classes of data, which degrades the performance of CL and learned representations. To tackle this problem, we propose a collaborative contrastive learning framework consisting of two approaches: feature fusion and neighborhood matching, by which a unified feature space among clients is learned for better data representations. Feature fusion provides remote features as accurate contrastive information to each client for better local learning. Neighborhood matching further aligns each client's local features to the remote features such that well-clustered features among clients can be learned. Extensive experiments show the effectiveness of the proposed framework. It outperforms other methods by 11% on IID data and matches the performance of centralized learning.

1 Introduction

Collaborative learning is an effective approach for multiple distributed clients to collaboratively learn a shared model from decentralized data. In the learning process, each client updates the local model by using local data, and then a central server aggregates the local models to obtain a shared model. In this way, collaborative learning enables learning from decentralized data [McMahan *et al.*, 2017] while keeping data local for privacy. Collaborative learning can be applied to healthcare applications, where many personal devices such as mobile phones collaboratively learn to provide early warnings to cognitive diseases such as Parkinson's and to assess mental health [Chen *et al.*, 2020b]. Collaborative learning can also be used for robotics, in which multiple robots learn a shared navigation scheme to adapt to new environments [Liu *et al.*, 2019]. Compared with local learning, collaborative learning improves navigation accuracy by utilizing knowledge from other robots.

Existing collaborative learning approaches assume local data is fully labeled so that supervised learning can be used for the model update on each client. However, labeling all the data is usually unrealistic due to high labor costs and the requirement of expert knowledge. For example, in medical diagnosis, even if the patients are willing to spend time on labeling all the local data, the deficiency of expert knowledge of these patients will result in large label noise and thus inaccurate learned model. The deficiency of labels makes supervised collaborative learning impractical. Self-supervised learning can address this challenge by pre-training a neural network encoder with unlabeled data, followed by fine-tuning for a downstream task with limited labels. Contrastive learning (CL), an effective self-supervised learning approach [Chen *et al.*, 2020a], can learn data representations from unlabeled data to improve the model. By integrating CL into collaborative learning, clients can collaboratively learn data representations by using a large amount of data without labeling.

In collaborative learning, data collected on clients are inherently far from IID [Hsu *et al.*, 2020], which results in two unique challenges when integrating collaborative learning with CL as collaborative contrastive learning (CCL) to learn high-quality representations. The *first challenge* is that each client only has a small amount of unlabeled data with limited diversity, which prevents effective contrastive learning. More specifically, compared with the global data (the concatenation of local data from all clients), each client only has a subset of the global data with a limited number of classes [McMahan *et al.*, 2017; Zhao *et al.*, 2018; Wu *et al.*, 2021b]. For instance, in real-world datasets [Luo *et al.*, 2019], each client only has one or two classes out of seven object classes. Since conventional contrastive learning frameworks [He *et al.*, 2020; Chen *et al.*, 2020a] are designed for centralized learning on

[*] Corresponding author

large-scale datasets with sufficient data diversity, directly applying them to local learning on each client will result in the low quality of learned representations.

The *second challenge* is that each client focuses on learning its local data without considering the data on the other clients. As a result, the features of data in the same class but from different clients may not be well-clustered even though they could have been clustered for improved representations. Data are decentralized in collaborative learning and even if two clients have data of the same class, they are unaware of this fact and cannot leverage it to collaboratively learn to cluster these data. Besides, even if one client has knowledge of other's data, since no labels are available, there is no easy way to identify the correct data clusters and perform clustering for better representations.

To address these challenges, we propose a collaborative contrastive learning (CCL) framework to learn visual representations from decentralized unlabeled data on distributed clients. The framework employs contrastive learning [He *et al.*, 2020] for local learning on each client and consists of two approaches to learn high-quality representations. The first approach is feature fusion and it provides remote features as accurate contrastive information to each client for better local learning. To protect the privacy of remote features against malicious clients, we employ an encryption method [Huang *et al.*, 2020] to encrypt the images before generating their features. The second approach is neighborhood matching and it further aligns each client's local features to the fused features such that well-clustered features among clients are learned.

In summary, the main contributions of the paper include:

- **Collaborative contrastive learning framework.** We propose a framework with two approaches to learning visual representations from unlabeled data on distributed clients. The first approach improves the local representation learning on each client with limited data diversity, and the second approach further learns unified global representations among clients.
- **Feature fusion for better local representations.** We propose a feature fusion approach to leverage remote features for better local learning while avoiding raw data sharing. The remote features serve as negatives in the local contrastive loss to achieve a more accurate contrast with fewer false negatives and more diverse negatives.
- **Neighborhood matching for improved global representations.** We propose a neighborhood matching approach to cluster decentralized data across clients. During local learning, each client identifies the remote features to cluster local data with and performs clustering. In this way, well-clustered features among clients can be learned.

2 Background and Related Work

Contrastive Learning. Contrastive learning is a self-supervised approach to learn an encoder (i.e. a convolutional neural network without the final classifier) for extracting visual representation vectors from the unlabeled input images by performing a proxy task of instance discrimination [Chen *et al.*, 2020a; He *et al.*, 2020; Wu *et al.*, 2018; Wu *et al.*, 2021a]. For an input image x, its representation vector z is obtained by $z = f(x),\ z \in \mathbb{R}^d$, where $f(\cdot)$ is the encoder. Let the representation vectors *query* q and *key* k^+ form a positive pair, which are the representation vectors from two transformations (e.g. cropping and flipping) of the same input image. Let Q be the *memory bank* with K representation vectors stored, serving as negatives. The positive pair *query* q and *key* k^+ will be contrasted with each vector $n \in Q$ (i.e. negatives) by the loss function:

$$\ell_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{n \in Q} \exp(q \cdot n/\tau)} \quad (1)$$

Minimizing the loss will learn an encoder to generate visual representations. Then a classifier can be trained on top of the encoder by using limited labeled data.

However, existing contrastive learning approaches are designed for centralized learning [Chen *et al.*, 2020a; He *et al.*, 2020] and require sufficient data diversity for learning. When applied to each client in collaborative learning with limited data diversity, their performance will greatly degrade. Therefore, an approach to increase the local data diversity on each client while protecting the shared information is needed.

Collaborative Learning. The goal of collaborative learning is to learn a shared model by aggregating locally updated models from clients while keeping raw data on local clients [McMahan *et al.*, 2017]. In collaborative learning, there are C clients indexed by c. The training data D is distributed among clients, and each client c has a subset of the training data $D_c \subset D$. There are recent works aiming to optimize the aggregation process [Reisizadeh *et al.*, 2020; Nguyen *et al.*, 2020]. While our work can be combined with these works, for simplicity, we employ a typical collaborative learning algorithm [McMahan *et al.*, 2017]. The learning is performed round-by-round. In communication round t, the server randomly selects $\beta \cdot C$ clients C^t and send them the global model with parameters θ^t, where β is the percentage of active clients per round. Each client $c \in C^t$ updates the local parameters θ_c^t on local dataset D_c for E epochs to get θ_c^{t+1} by minimizing the loss $\ell_c(D_c, \theta^t)$. Then the local models are aggregated into the global model by averaging the weights $\theta^{t+1} \leftarrow \sum_{c \in C^t} \frac{|D_c|}{\sum_{i \in C^t} |D_i|} \theta_c^{t+1}$. This learning process continues until the global model converges.

To improve the performance of collaborative learning on non-IID data, [Zhao *et al.*, 2018; Jeong *et al.*, 2018] share local raw data (e.g. images) among clients. However, sharing raw data among clients will cause privacy concerns [Li *et al.*, 2020]. Besides, they need fully labeled data to perform collaborative learning, which requires expert knowledge and potentially high labeling costs. Therefore, an approach to performing collaborative learning with limited labels and avoiding sharing raw data is needed.

3 Framework Overview

We propose a collaborative contrastive learning (CCL) framework to learn representations from unlabeled data on distributed clients. These distributed data cannot be combined in a single location to construct a centralized dataset due to privacy and legal constraints [Kairouz *et al.*, 2019]. The overview of the proposed framework is shown in Figure 1. There is a

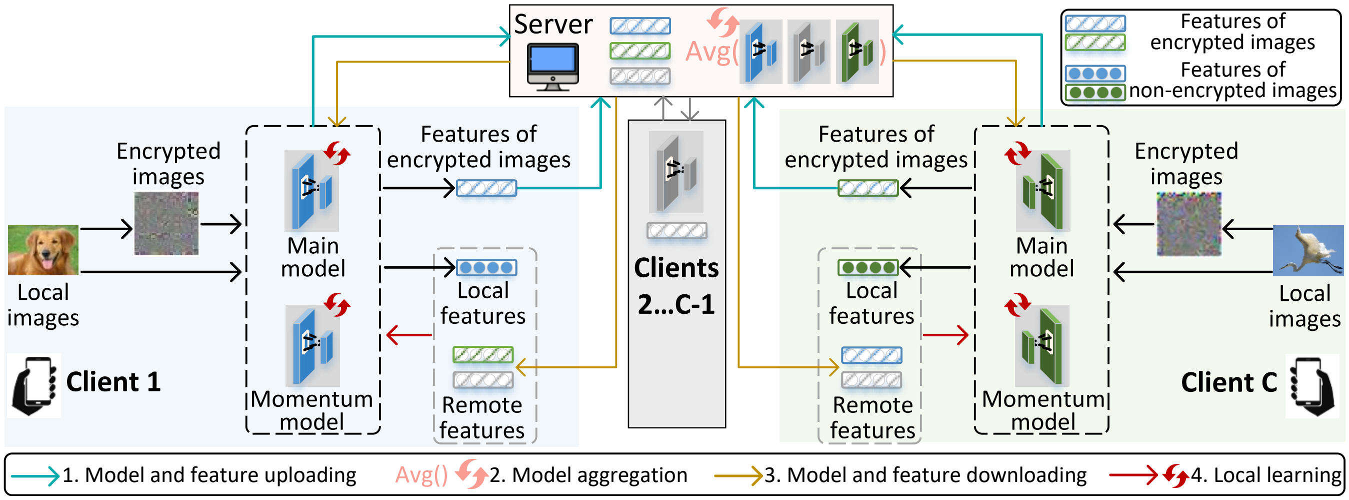

Server
Avg( )
Features of encrypted images
Features of non-encrypted images
Encrypted images
Local images
Main model
Momentum model
Features of encrypted images
Local features
Remote features
Client 1
Clients 2…C-1
Client C
1. Model and feature uploading
Avg() 2. Model aggregation
3. Model and feature downloading
4. Local learning

Figure 1: Overview of the proposed collaborative contrastive learning (CCL) framework. Four steps are performed in each learning round, including model and feature uploading from clients to the server, model aggregation on the server, model and feature downloading to clients, and local learning. Local learning is performed by the proposed feature fusion in Sect. 4 and neighborhood matching in Sect. 5.

central server that coordinates multiple clients to learn representations. The local model on each client is based on MoCo [He *et al.*, 2020]. CCL follows the proposed feature fusion technique to reduce the false negative ratio on each client for better local learning (Sect. 4). Besides, based on fused features, CCL further uses the proposed neighborhood matching to cluster representations of data from different clients to learn unified representations among clients (Sect. 5).

Before introducing the details of feature fusion and neighborhood matching, we present the proposed CCL process. CCL is performed round-by-round and there are four steps in each round as shown in Figure 1. First, each client c uploads its latest model (consisting of the main model f_q^c and momentum model f_k^c) and latest features $\overline{Q}_{l,c}$ of encrypted images to the server. Second, the server aggregates main models from clients by $\theta_q \leftarrow \sum_{c\in C} \frac{|D_c|}{|D|}\theta_q^c$ and momentum models by $\theta_k \leftarrow \sum_{c\in C} \frac{|D_c|}{|D|}\theta_k^c$ to get updated f_q and f_k, where $|D_c|$ is the data size of client c and $|D|$ is the total data size of $|C|$ clients. The server also combines features as $\overline{Q} = \{\overline{Q}_{l,c}\}_{c\in C}$. Third, the server downloads the aggregated models f_q and f_k and combined features $\overline{Q}$ excluding $\overline{Q}_{l,c}$ as $Q_{s,c} = \{\overline{Q} \setminus \overline{Q}_c\}$ to each client c. Fourth, each client updates its local models with f_q and f_k, and then performs local contrastive learning for multiple epochs with local features $Q_{l,c}$ and remote features $Q_{s,c}$ by using loss Eq.(12) including contrastive loss with fused features Eq.(6) and neighborhood matching Eq.(11). During local contrastive learning, to generate features $\overline{Q}_{l,c}$ for uploading in the next round, images x are encrypted by *InstaHide* [Huang *et al.*, 2020] as $\tilde{x}$ and fed into momentum model f_k^c. In this way, even if a malicious client can ideally recover $\tilde{x}$ from the features $\overline{Q}_{l,c}$, which is already very unlikely in practice, it still cannot reconstruct x from $\tilde{x}$ since *Instahide* effectively hides information contained in x. Next, we present the details of local contrastive learning, including feature fusion to reduce false negative ratio in Sect. 4 and neighborhood matching for unified representations in Sect. 5.

4 Local Learning with Feature Fusion

Next, we focus on how to perform local CL in each round of CCL. We first present the key challenge of CL on each client, which does not exist in conventional centralized CL. Then we propose feature fusion to tackle this challenge and introduce how to perform local CL with fused features.

Key challenge: *Limited data diversity* causes a high false negative (FN) ratio on each client. A low FN ratio is crucial to achieving accurate CL [Kalantidis *et al.*, 2020]. For one image sample q, FNs are features that we use as negative features but actually correspond to images of the same class as q. In centralized CL, the percentage of FNs is inherently low since diverse data are available. The model has access to the whole dataset D with data from all the classes instead of a subset D_c as in collaborative learning. Thus, when we randomly sample negatives from D, the FN ratio is low. For instance, when dataset D has 1000 balanced classes and the negatives n are randomly sampled, for any image q to be learned, only $\frac{1}{1000}$ of n are from the same class as q and are FNs.

However, in collaborative learning, the FN ratio is inherently high on each client due to the limited data diversity, which significantly degrades the performance of contrastive learning. For instance, in real-world datasets [Luo *et al.*, 2019], one client can have only one or two classes out of seven classes. With limited data diversity on each client, when learning image sample q, many negatives n to contrast with will be from the same class as q and are FNs. To perform contrastive learning by minimizing the contrastive loss in Eq.(1), the model scatters the FNs away from q, which should have been clustered since they are from the same class. As a result, the representations of samples from the same class will be scattered instead of clustered and degrade the learned representations.

4.1 Feature Fusion

To address this challenge, we propose feature fusion to share negatives in the feature space (i.e. the output vector of the encoder), which reduces FN and improves the diversity of negatives while avoiding raw data sharing. Let $Q_{l,c}$ be the memory bank of size K for local features of non-encrypted images on client c, and let $\overline{Q}_{l,c}$ be features of encrypted images. In one round t of CCL, features $\overline{Q}_{l,c}$ of encrypted images on each client c will be uploaded to the server (i.e. step 1 in Figure 1). The server also downloads combined features $\overline{Q}$ excluding $\overline{Q}_{l,c}$ to each client c (i.e. step 3 in Figure 1) to form its memory bank of remote negatives $Q_{s,c}$ as follows.

$$Q_{s,c} = \{\overline{Q}_{l,i} \mid 1 \leq i \leq |C|, i \neq c\} \qquad (2)$$

where C is the set of all clients.

On client c, with local negatives $Q_{l,c}$ and remote negatives $Q_{s,c}$, the loss for sample q is defined as:

$$\ell_q = -\log \left[\frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{n \in \{Q_l \cup Q_s\}} \exp(q \cdot n/\tau)} \right] \qquad (3)$$

where we leave out the client index c in $Q_{l,c}$ and $Q_{s,c}$ for conciseness. ℓ_q is the negative log-likelihood over the probability distribution generated by applying a softmax function to a pair of input q and its positive k^+, negatives n from both local negatives Q_l and remote negatives Q_s.

Effectiveness of feature fusion. The remote negatives Q_s reduce the FN ratio in local contrastive learning and improve the quality of learned representations on each client. More specifically, in collaborative learning with non-IID data, we assume the global dataset D has M classes of data, each class with the same number of data. Each client $c \in C$ has a subset $D_c \subset D$ of the same length in m classes ($m \leq M$) [Zhao *et al.*, 2018; McMahan *et al.*, 2017]. For a sample q on client c, when only local negatives $Q_{l,c}$ are used, $\frac{1}{m}|Q_{l,c}|$ negatives will be in the same class as q, which results in an FN ratio $R_{FN} = \frac{1}{m}$. Since m is usually small (e.g. 2) due to limited data diversity, the FN ratio R_{FN} will be large (e.g. 50%) and degrade the quality of learned representations. Different from this, when remote negatives are used, the FN ratio is:

$$R_{FN}(q) = \frac{\frac{1}{m}|Q_{l,c}| + \sum_{i \in C, i \neq c} \mathbb{I}(i,q)\frac{1}{m}|Q_{l,i}|}{|Q_{l,c}| + \sum_{i \in C, i \neq c} |Q_{l,i}|} \leq \frac{1}{m} \qquad (4)$$

where $\mathbb{I}(i,q)$ is an indicator function that equals 1 when client i has data of the same class as q, and 0 otherwise.

In most cases, $R_{FN}(q)$ is effectively reduced by the remote negatives. *First*, in the extreme case of non-IID data distribution, where the classes on each client are mutually exclusive [Zhao *et al.*, 2018], all $\mathbb{I}(i,q)$ equal 0 and $R_{FN}(q) = \frac{1}{m}\frac{|Q_{l,c}|}{|Q_{l,c}| + \sum_{i \in C, i \neq c}|Q_{l,i}|} = \frac{1}{m|C|} \ll \frac{1}{m}$. With the remote negatives, the FN ratio is effectively reduced by a factor $|C|$. *Second*, as long as not all clients have data of the same class as q, some elements in $\{\mathbb{I}(i,q)\}_{i=1,\ i \neq c}^{|C|}$ will be 0, and $R_{FN}(q)$ in Eq.(4) will be smaller than $\frac{1}{m}$. In this case, the FP ratio is also reduced. *Third*, even if the data on each client is IID and all $\mathbb{I}(i,q)$ equal 1, which is unlikely in realistic collaborative learning [Hsu *et al.*, 2020], the FN ratio $R_{FN}(q)$ will be $\frac{1}{m}$. In this case, while $R_{FN}(q)$ is the same as that without remote negatives, the increased diversity of negatives from other clients can still benefit the local contrastive learning.

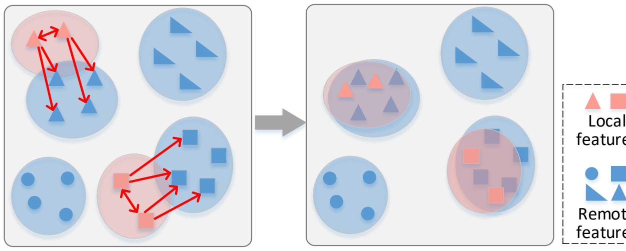
Further reducing the false negative ratio. To further reduce the FN ratio, we propose to exclude the local negatives by removing Q_l in the denominator of Eq.(3) and only keeping remote negatives Q_s. The corresponding FN ratio becomes:

$$R'_{FN}(q) = \frac{\sum_{i \in C, i \neq c} \mathbb{I}(i,q)\frac{1}{m}|Q_{l,i}|}{\sum_{i \in C, i \neq c}|Q_{l,i}|} \leq R_{FN}(q) \qquad (5)$$

As long as not all other clients have data in the same class as q, some $\mathbb{I}(i,q)$ will be 0. In this way, $R'_{FN}(q) < R_{FN}(q)$ and the FN ratio is further reduced. Based on the loss ℓ_q for one sample q in Eq.(3), the contrastive loss for one mini-batch B is:

$$\mathcal{L}_{contrast} = \frac{1}{|B|}\sum_{q \in B} \ell_q \qquad (6)$$

5 Local Learning with Neighborhood Matching



(a) Before neighborhood matching (b) After neighborhood matching

Figure 2: Neighborhood matching aligns each client's local features to the remote features such that well-clustered features among clients are learned.

In local contrastive learning, each client focuses on learning its local data without considering data on the other clients. As a result, the features of data in the same class but from different clients may not be well-clustered even though they could be clustered for improved representations.

Challenge: To cluster local features to the correct remote features, one has to identify local data and remote features that are in the same class. However, since no labels are available for local data and remote features, there is no easy way to identify the correct clusters to push local features to.

To address this challenge, we propose a neighborhood matching approach to identify the remote features to cluster local data with and define an objective function to perform the clustering. First, during local learning on one client, as shown in Figure 2, for each local sample we find N nearest features from both the *remote* and *local* features as neighbors. Then the features of the local sample will be pushed to these neighbors by the proposed entropy-based loss. Since the model is synchronized from the server to clients in each communication round, the remote and local features are encoded by similar models on different clients. Therefore, the neighbors are likely to be in the same class as the local sample being learned, and clustering them will improve the learned representations of global data. In this way, the global model is also improved when aggregating local models.

Identifying neighbors. To push each local sample close to its neighbors, we minimize the entropy of one sample's matching probability distribution to either a remote feature or a local feature. To improve the robustness, we match one sample to N nearest features at the same time, instead of only one nearest feature. By minimizing the entropy for N nearest neighbors, the sample's matching probability to each of the nearest neighbors will be individually certain.

For each local sample q_i, we regard top-N closest features, either from local or remote features, as neighbors. To find the neighbors, we first define the neighbor candidates as:

$$Q' = \{Q_{s+l,i} | \; i \sim \mathcal{U}(|Q_s| + K, K)\} \tag{7}$$

where $i \sim \mathcal{U}(|Q_s| + K, K)$ samples K integer indices from $[|Q_s| + K]$ randomly at uniform. $Q_{s+l,i}$ is the element with index *i* in the union of remote and local features $Q_s \cup Q_l$. For one local sample q_i, the neighbors $P(q_i)$, which are the top-N nearest neighbor candidates Q', is given by:

$$P(q_i) = \{Q'_j \mid j \in topN(S_{i,m}), 1 \leq m \leq K\} \tag{8}$$

where $S_{i,m} = sim(q_i, n_m) = q_i^T \cdot n_m / \|q_i\| \|n_m\|$ is the cosine similarity between q_i and one neighbor candidate $n_m \in Q'$.

Neighborhood matching loss. To make the probability of q_i matching to each $n_j \in P(q_i)$ individually certain, we consider the set:

$$L_j = \{n_j\} \cup \{Q' \setminus P(q_i)\} \in \mathbb{R}^{(K-N+1)\times d} \tag{9}$$

where d is the dimension of one feature vector. L_j contains one of the top-N nearest neighbors n_j and neighbor candidates excluding all other top-N nearest neighbors.

Given L_j, the probability that sample q_i is matched to one of the neighbors $n_a \in L_j$ is:

$$p_{i,j,a} = \frac{\exp(q_i^T \cdot n_a / \tau_{nm})}{\sum_{n \in L_j} \exp(q_i^T \cdot n / \tau_{nm})}, \; n_a \in L_j \tag{10}$$

The temperature τ_{nm} controls the softness of the probability distribution [Hinton *et al.*, 2015]. Since n_j has the largest cosine similarity with q_i for $n \in L_j$, $p_{i,j,a}$ will have the largest value when $n_a = n_j$ for $n_a \in L_j$. In this way, when minimizing the entropy of the probability distribution $\{p_{i,j,a}\}_{n_a \in L_j}$, the matching probability of q_i and n_j will be maximized.

For one mini-batch B, to match each sample to its N nearest neighbors, the entropy for all samples in this mini-batch is calculated as:

$$\mathcal{L}_{neigh} = -\frac{1}{|B|} \sum_{i \in B} \frac{1}{N} \sum_{j=1}^{N} \sum_{a=1}^{K-N+1} p_{i,j,a} \log(p_{i,j,a}) \tag{11}$$

where $K - N + 1$ is the number of features in L_j, and N is the number of nearest neighbors to match. By minimizing $\mathcal{L}_{neigh}$, each $i \in B$ will be aligned to its top-N nearest neighbors.

Final loss. Based on the contrastive loss with fused features in Eq.(6) and neighborhood matching loss in Eq.(11), the overall objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{contrast} + \lambda \mathcal{L}_{neigh} \tag{12}$$

where λ is a weight parameter.

6 Experimental Results

Datasets, model architecture, and distributed settings. We evaluate the proposed approaches on three datasets, including CIFAR-10 [Krizhevsky *et al.*, 2009], CIFAR-100 [Krizhevsky *et al.*, 2009], and Fashion-MNIST [Xiao *et al.*, 2017]. We use ResNet-18 as the base encoder and use a 2-layer MLP to project the representations to 128-dimensional feature space [Chen *et al.*, 2020a; He *et al.*, 2020]. For each of the three datasets, we consider one IID setting and two non-IID settings. The detailed collaborative learning settings and training details can be found in the Appendix.

Metrics. To evaluate the quality of learned representations, we use standard metrics for centralized self-supervised learning, including *linear evaluation* and *semi-supervised learning* [Chen *et al.*, 2020a]. Besides, we evaluate by *collaborative finetuning* for realistic collaborative learning. In linear evaluation, a linear classifier is trained on top of the frozen base encoder, and the test accuracy represents the quality of learned representations. We first perform collaborative learning by the proposed approaches without labels to learn representation. Then we *fix* the encoder and train a linear classifier on the 100% labeled dataset on top of the encoder. The classifier is trained for 100 epochs by the SGD optimizer following the hyper-parameters from [He *et al.*, 2020]. In semi-supervised learning, we first train the base encoder without labels in collaborative learning. Then we append a linear classifier to the encoder and *finetune* the whole model on 10% or 1% labeled data for 20 epochs with SGD optimizer following the hyper-parameters from [Caron *et al.*, 2020]. In collaborative finetuning, the learned encoder by the proposed approaches is used as the initialization for finetuning the whole model by supervised collaborative learning [McMahan *et al.*, 2017] with few locally labeled data on clients. Detailed collaborative finetuning settings can be found in the Appendix.

Baselines. We compare the proposed methods with multiple approaches. *Predicting Rotation* is a self-supervised learning approach by predicting the rotation of images [Gidaris *et al.*, 2018]. *DeepCluster-v2* is the improved version of DeepCluster [Caron *et al.*, 2020; Caron *et al.*, 2018] and achieves SOTA performance. *SwAV* and *SimCLR* are SOTA approaches for self-supervised learning [Caron *et al.*, 2020; Chen *et al.*, 2020a]. We combine these approaches with FedAvg as *FedRot*, *FedDC*, *FedSwAV*, and *FedSimCLR*. *FedCA* is the SOTA collaborative unsupervised learning approach with a shared dictionary and online knowledge distillation [Zhang *et al.*, 2020]. Besides, we compare with two methods as upper bounds. *MoCo* [He *et al.*, 2020] is a centralized contrastive learning method assuming all data are combined in a single location. We compare with MoCo since the local model in the proposed methods is based on it. *FedAvg* [McMahan *et al.*, 2017] is a fully supervised collaborative learning method.

6.1 Linear Evaluation

Linear evaluation on CIFAR-10. We evaluate the proposed approaches by linear evaluation with 100% data labeled for training the classifier on top of the *fixed* encoder learned with unlabeled data by different approaches. The proposed approaches significantly outperform other methods and even match the performance of the centralized upper bound method.

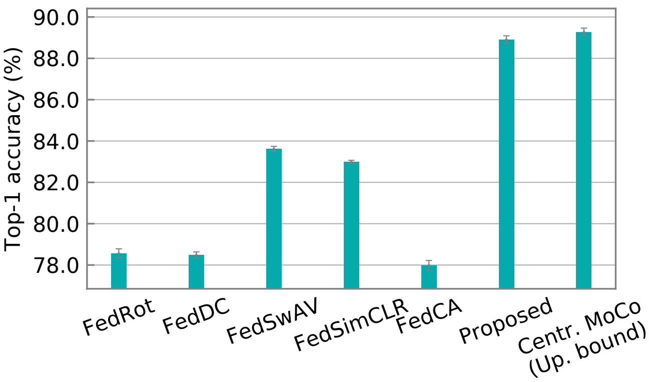


Figure 3: **Linear evaluation accuracy on CIFAR-10 in IID setting.** The classifier is trained by 100% labels on a *fixed* encoder learned by different approaches. Error bar denotes standard derivation over three independent runs. Centralized MoCo is the upper bound method.

The results on CIFAR-10 in the IID setting are shown in Figure 3. On CIFAR-10, the proposed approaches achieve 88.90% top-1 accuracy, only 0.38% below the upper bound method centralized MoCo. The proposed approaches also outperform the SOTA method FedCA by +10.92% top-1 accuracy and the best-performing baseline FedSwAV by +5.28%.

	CIFAR-10			CIFAR-100			FMNIST		
Method	IID	Non-1	Non-2	IID	Non-1	Non-2	IID	Non-1	Non-2
FedRot	78.57	75.88	70.98	45.80	44.57	43.15	83.74	82.65	82.90
FedDC	78.49	69.97	69.34	49.27	49.06	47.21	88.41	85.92	88.35
FedSwAV	83.62	75.07	75.36	55.51	51.45	53.77	89.63	87.11	89.75
FedSimCLR	82.99	71.23	73.30	48.83	45.67	48.46	88.45	84.41	86.23
FedCA	77.98	75.57	75.50	48.93	47.70	48.22	86.98	86.22	86.46
Proposed	**88.90**	**79.07**	**78.31**	**61.91**	**57.54**	**58.63**	**91.26**	**88.13**	**90.08**
Upper bounds									
MoCo (Centralized)	89.28	—	—	63.72	—	—	91.97	—	—
FedAvg (Supervised)	92.88	60.60	59.03	73.08	67.59	66.90	94.12	77.08	69.92

Table 1: **Linear evaluation on CIFAR-10, CIFAR-100, and FMNIST under the IID and two non-IID settings.** 100% labeled data are used for learning the classifier on the *fixed* encoder and top-1 accuracy is reported. Centralized MoCo and Supervised FedAvg are the upper bound methods.

Linear evaluation on various datasets and distributed settings. We evaluate the proposed approaches on different datasets and collaborative learning settings. The results under the IID setting, non-IID settings 1 and 2 on CIFAR-100 and FMNIST are shown in Table **??**. The linear classifier is trained on top of the *fixed* encoder learned with unlabeled data by different approaches. Under all the three collaborative learning settings and on both datasets, the proposed approaches significantly outperform the baselines.

For example, on CIFAR-100 the proposed approaches outperform the best-performing baseline by 6.40%, 6.09%, and 4.86% under three collaborative learning settings, respectively. Besides, compared with the two upper bound methods, the proposed methods match the performance of the upper bound centralized MoCo under IID setting, and outperforms supervised FedAvg on CIFAR-10 under non-IID settings.

	CIFAR-10		CIFAR-100		FMNIST	
Labeled ratio	10%	1%	10%	1%	10%	1%
FedRot	85.38	71.62	43.78	19.84	91.23	47.94
FedDC	78.88	44.18	40.69	11.93	88.97	30.25
FedSwAV	84.51	48.96	50.23	13.82	90.48	62.08
FedSimCLR	86.05	75.36	49.54	27.45	91.28	84.46
FedCA	84.15	41.25	48.57	8.13	91.67	36.93
Proposed	**89.27**	**84.79**	**58.49**	**40.71**	**92.18**	**85.63**
MoCo (Centralized)	88.44	81.75	57.76	37.79	92.46	86.78

	CIFAR-10		CIFAR-100		FMNIST	
Labeled ratio	10%	1%	10%	1%	10%	1%
FedRot	77.82	58.48	43.50	18.80	90.80	60.72
FedDC	71.25	31.85	40.85	11.42	86.80	37.91
FedSwAV	78.25	39.87	46.58	14.11	88.77	35.85
FedSimCLR	78.49	58.13	46.89	23.86	90.41	80.72
FedCA	79.75	58.76	48.10	8.07	89.60	38.98
Proposed	**84.01**	**67.87**	**54.85**	**31.29**	**91.29**	**82.13**
MoCo (Centralized)	88.44	81.75	57.76	37.79	92.46	86.78

Table 2: **Semi-supervised learning under IID setting (top) and non-IID setting 1 (bottom).** We *finetune* the encoder and classifier with different ratios of labeled data and report the top-1 accuracy.

6.2 Semi-Supervised Learning

We further evaluate the proposed approaches by *semi-supervised learning*, where both the encoder and classifier are *finetuned* with 10% or 1% labeled data after learning the encoder on unlabeled data by different approaches. We evaluate the approaches under the IID collaborative learning setting and two non-IID collaborative learning settings. Table **??** shows the comparison of our results against the baselines under the IID collaborative learning setting (top) and non-IID setting 1 (bottom). Our approach significantly outperforms the self-supervised baselines with 10% and 1% labels. Notably, the proposed methods even outperform the upper bound method centralized MoCo on CIFAR-10 and CIFAR-100 datasets under the IID setting. Results under non-IID collaborative learning setting 2 will be shown in the Appendix for conciseness.

6.3 Collaborative Finetuning

We evaluate the performance of the proposed approaches by collaborative finetuning the learned encoder with few locally labeled data on clients. The results under the IID setting and non-IID setting 1 are shown in Table **??** (top) and Table **??** (bottom), respectively. On both collaborative learning settings and three datasets, the proposed approaches consistently outperform the baselines.

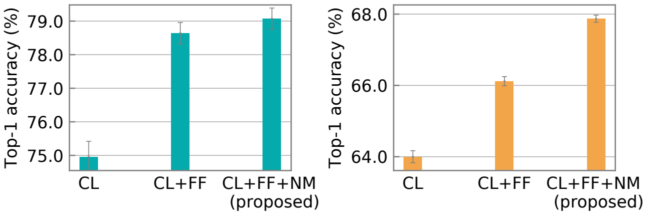
6.4 Ablations

Effectiveness of feature fusion and neighborhood matching. We evaluate three approaches. Contrastive learning (CL) is the approach without feature fusion (FF) or neighborhood matching (NM). CL+FF adds feature fusion, and CL+FF+NM further adds neighborhood matching. We evaluate the approaches by linear evaluation and semi-supervised learning (1% labels) under the non-IID collaborative learning

	CIFAR-10		CIFAR-100		FMNIST	
Labeled ratio	10%	1%	10%	1%	10%	1%
FedRot	85.16	74.25	49.97	16.65	90.49	82.81
FedDC	79.98	71.17	42.81	21.47	90.17	84.22
FedSwAV	85.23	78.92	51.67	26.75	91.33	85.10
FedSimCLR	83.52	75.10	51.73	15.32	91.64	84.31
FedCA	82.32	72.77	50.78	21.10	91.57	84.29
Proposed	**89.33**	**82.52**	**56.88**	**33.15**	**92.15**	**87.11**
FedAvg (Supervised)	74.71	39.35	33.16	8.07	87.95	75.68

	CIFAR-10		CIFAR-100		FMNIST	
Labeled ratio	10%	1%	10%	1%	10%	1%
FedRot	57.34	56.80	46.93	17.12	75.02	72.02
FedDC	60.37	49.28	40.29	21.20	77.12	73.16
FedSwAV	57.34	51.93	46.65	22.06	75.45	73.17
FedSimCLR	63.05	51.63	47.69	11.19	76.49	73.25
FedCA	59.52	57.33	49.14	21.50	73.23	71.34
Proposed	**65.80**	**59.30**	**50.75**	**28.25**	**78.81**	**76.88**
FedAvg (Supervised)	48.41	32.33	33.26	8.42	67.42	66.03

Table 3: **Collaborative finetuning under IID setting (top) and non-IID setting 1 (bottom).** We *finetune* the encoder and classifier with different ratios of locally labeled data on clients by supervised collaborative learning and report the top-1 accuracy.



(a) Linear evaluation.

(b) Semi-sup. learning.

Figure 4: Ablations on CIFAR-10 dataset under the non-IID setting. CL is vanilla contrastive learning. FF is feature fusion and NM is neighborhood matching. Top-1 accuracy of linear classifier and semi-supervised learning (1% labels) are reported. Error bar denotes standard derivation over three independent runs.

setting (non-IID setting 1). As shown in Figure 4, with linear evaluation, CL achieves 74.96% top-1 accuracy. Adding FF improves the accuracy by 3.68%, and adding NM further improves the accuracy by 0.43%. With semi-supervised learning (1% labels), CL achieves 64.00% top-1 accuracy. Adding FF improves the accuracy by 2.12% and adding NM further improves the accuracy by 1.75%. These results show the effectiveness of feature fusion and neighborhood matching.

7 Conclusion

We propose a framework for collaborative contrastive representation learning. To improve representation learning on each client, we propose feature fusion to provide remote features as accurate contrastive data to each client. To achieve unified representations among clients, we propose neighborhood matching to align each client's local features to the remote ones. Experiments show superior accuracy of the proposed framework compared with the state-of-the-art.

Acknowledgments

This work was supported in part by NSF CNS-2122320, CNS-212220, CNS-1822099, and CNS-2007302.

References

[Caron *et al.*, 2018] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[Caron *et al.*, 2020] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[Chen *et al.*, 2020a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 13–18 Jul 2020.

[Chen *et al.*, 2020b] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 2020.

[Gidaris *et al.*, 2018] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[He *et al.*, 2020] 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.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[Hsu *et al.*, 2020] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *ECCV 2020: 16th European Conference*, pages 76–92. Springer, 2020.

[Huang *et al.*, 2020] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, pages 4507–4518. PMLR, 2020.

[Jeong *et al.*, 2018] 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.

[Kairouz *et al.*, 2019] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[Kalantidis *et al.*, 2020] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Neural Information Processing Systems (NeurIPS)*, 2020.

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

[Li *et al.*, 2020] 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.

[Liu *et al.*, 2019] Boyi Liu, Lujia Wang, and Ming Liu. Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems. *IEEE Robotics and Automation Letters*, 4(4):4555–4562, 2019.

[Luo *et al.*, 2019] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Realworld image datasets for federated learning. *arXiv preprint arXiv:1910.11089*, 2019.

[McMahan *et al.*, 2017] 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.

[Nguyen *et al.*, 2020] Hung T Nguyen, Vikash Sehwag, Seyyedali Hosseinalipour, Christopher G Brinton, Mung Chiang, and H Vincent Poor. Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218, 2020.

[Reisizadeh *et al.*, 2020] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031, 2020.

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

[Wu *et al.*, 2021a] Yawen Wu, Zhepeng Wang, Dewen Zeng, Yiyu Shi, and Jingtong Hu. Enabling on-device self-supervised contrastive learning with selective data contrast. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 655–660. IEEE, 2021.

[Wu *et al.*, 2021b] Yawen Wu, Dewen Zeng, Zhepeng Wang, Yiyu Shi, and Jingtong Hu. Federated contrastive learning for volumetric medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 367–377. Springer, 2021.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Zhang *et al.*, 2020] 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.

[Zhao *et al.*, 2018] 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.