Clustering-Guided Federated Learning of Representations

Runxuan Miao [1] **Erdem Koyuncu** [1]

Abstract

Federated self-supervised learning (FedSSL) methods have proven to be very useful in learning unlabeled data that is distributed to multiple clients, possibly heterogeneously. However, there is still a lot of room for improvement for FedSSL methods, especially for the case of highly heterogeneous data and a large number of classes. In this paper, we introduce federated representation learning through clustering (FedRLC) scheme that utilizes i) a crossed KL divergence loss with a data selection strategy during local training and ii) a dynamic upload on local cluster centers during communication updates. Experimental results show that FedRLC achieves state-of-the-art results on widely used benchmarks even with highly heterogeneous settings and datasets with a large number of classes such as CIFAR-100.

1. Introduction

By considering information security and accommodating low-resource computing devices, federated learning (FL) provides a means to train a neural network model over distributed data across multiple machines. However, most existing FL methods (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020; Li et al., 2021a) rely on labeled data for supervised learning. Recently, self-supervised learning (SSL) methods have been proposed for learning representations on unlabeled data. Most SSL paradigms (Chen et al., 2020; Grill et al., 2020; He et al., 2020; Caron et al., 2020; Chen & He, 2021; Le-Khac et al., 2020; Zbontar et al., 2021) assume that the data is centralized.

Recently, Federated SSL (FedSSL) (Zhuang et al., 2022; 2021; Miao & Koyuncu, 2022; Zhang et al., 2020; Wang et al., 2023) methods have been developed to learn representations of unlabeled data that is distributed to several local machines. For example, FedU (Zhuang et al., 2021) and FedEMA (Zhuang et al., 2022) directly adapt a fundamental centralized SSL model that is referred to as "Bootstrap Your Own Latent (BYOL) (Grill et al., 2020)" to federated learning. However, combining BYOL and FL directly can raise challenges: When the number of classes are large, the learned representations from BYOL are typically non-uniformly distributed over the representation space, leading to sub-optimal performance.

In this paper, we aim to address the challenges of FedSSL via our proposed Federated Representation Learning through Clustering (FedRLC) framework. A key idea of FedRLC is to solve the clustering task to guide and aid in finding accurate representations. We do this by introducing a novel crossed KL divergence loss with a data selection strategy to optimize the cluster centers and the BYOL neural networks simultaneously. Intuitively, well-learned cluster centers are beneficial to extract more distinct information between different classes. Experimental results show that FedRLC improves the performance of existing FedSSL methods by a considerable margin and achieves state-of-the-art results on benchmark datasets such as CIFAR-100.

The rest of this paper is organized as follows: We introduce the BYOL approach and the FedSSL problem in Section 2. In Section 3, we introduce our proposed FedRLC scheme. Numerical results are provided in Section 4. We draw our main conclusions in Section 5.

2. Preliminaries

Contrastive learning and non-contrastive learning are two main directions in SSL learning. In this work, we focus on a non-contrastive approach based on the BYOL scheme. In fact, the existence of a large number of negative samples in contrastive learning causes class collision issues. The non-contrastive nature of BYOL circumvents this problem and typically provides a better performance; the FedEMA scheme (Zhuang et al., 2022) follows a similar approach.

In the following, we provide an overview of BYOL (Grill et al., 2020), and its straightforward federated generalization that we shall refer to as FedBYOL. Let $\mathcal{D}_k$ denote the local unlabeled dataset on Client k. Given some data

[1] Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, USA. Correspondence to: Runxuan Miao <rmiao6@uic.edu>, Erdem Koyuncu <ekoyuncu@uic.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

$x_i \in \mathcal{D}_k$, where i represents the data index, two samples $x_i^a \triangleq t^a(x_i)$ and $x_i^b \triangleq t^b(x_i)$ are generated through the augmentations t^a and t^b, respectively. The augmented data $t^\alpha(x_i)$, $\alpha \in \{a, b\}$ are then processed by the so-called online and target networks. The online network consists of an online encoder f^O and an online predictor g^O, which are trained by gradient descent.[1] The target network only consists of a target encoder f^T. The weights of f^T are updated via the exponential moving average (EMA) of the online encoder f^O, as will be explained in the following. Now, let $z_i^{\alpha,O} \triangleq g^O\left(f^O(t^\alpha(x_i))\right)$, $\alpha \in \{a, b\}$ and $z_i^{\alpha,T} \triangleq f^T(t^\alpha(x_i))$, $\alpha \in \{a, b\}$ denote the d-dimensional representations that one would obtain from the online and the target networks, respectively. Defining the scaled cosine similarity loss function as $\delta(x, y) \triangleq 2 - 2\frac{x^T y}{\|x\|\|y\|}$, BYOL uses the symmetrized loss $x_i \mapsto L(x_i) \triangleq \delta(z_i^{a,O}, z_i^{b,T}) + \delta(z_i^{b,O}, z_i^{a,T})$. The local objective of user k is then given by

$$(f^O, g^O) \mapsto L_{\text{INS}} \triangleq \textstyle\sum_{x_i \in D_k} L(x_i), \tag{1}$$

which signifies that only the online networks f^O, g^O are updated via gradient descent. The subscript "INS" means that we consider the instance-level loss without any consideration about the clusters. The target network parameters are instead updated through the EMA

$$f^T \leftarrow \sigma f^T + (1 - \sigma) f^O, \tag{2}$$

where $\sigma \in [0, 1]$.

FedSSL aims to learn a global model over the dataset $\mathcal{D} \triangleq \bigcup_{k=1}^K \mathcal{D}_k$. For example, one can aggregate the BYOL loss functions of all clients in order to attain the objective

$$\min \textstyle\sum_{k=1}^K \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}_k} L(x_i). \tag{3}$$

The objective (3) can be solved by using numerous FL algorithms (Li et al., 2020; Karimireddy et al., 2020; McMahan et al., 2017) such as the classic federated averaging (FedAvg) (McMahan et al., 2017). However, simply extending the loss (1) to FL leads to suboptimal performance as the learned features from BYOL are not uniform between distinct classes, especially for heterogeneous data. In this paper, we propose FedRLC as a new alternative for FedSSL. FedRLC learns accurate representations through updating and keeping track of the centroids of each class.

3. The FedRLC Framework

In this section, we will introduce the proposed FedRLC framework. Our scheme relies on clustering to guide and achieve good representations. The clustering is center-based. Hence, at each client, we keep track of M cluster centers, where the number of clusters M is assumed to be known a-priori. We start by introducing a novel crossed KL divergence loss with data selection for optimizing cluster centers to improve the quality of learned representations during local training. We will then present a dynamic rule to update the local cluster centers as well as the local neural networks during training. The block diagram of the FedRLC framework is illustrated in Figure 1 for local training at a certain Client k. The first stages to obtain the instance representations $z_i^{\alpha,\nu}$ for sample x_i (until L^{INS}) apply verbatim from the BYOL scheme. Note that we have similarly omitted to indicate the dependence of the representations on the client index for brevity. We now describe the next steps.

3.1. Crossed KL divergence loss with data selection

In FedRLC, we define a novel crossed KL divergence loss (CKL) to learn a well-separated representation. CKL aims at optimizing M cluster centers by a crossed divergence between probabilities calculated from the online network and the target distribution from the target network. Specifically, let $\mu_1, \ldots, \mu_M \in \mathbb{R}^d$ denote cluster centers at a certain Client k . In practice, the cluster centers are initialized randomly. Given $\alpha \in \{a, b\}$ and $\nu \in \{O, T\}$, let $q_{i,m}^{\alpha,\nu}$ denote the probability that the representation $z_i^{\alpha,\nu}$ belongs to cluster m with center μ_m. Following DEC (Xie et al., 2016), we model these cluster assignment probabilities with a student t-distribution with one degree of freedom

$$\Delta_m(z, \{\mu_n\}_{n=1}^M) \triangleq \frac{(1 + \|z - \mu_m\|^2)^{-\frac{1}{2}}}{\sum_n (1 + \|z - \mu_n\|^2)^{-\frac{1}{2}}}. \tag{4}$$

Specifically, we set

$$q_{i,m}^{\alpha,\nu} = \Delta_m(z_i^{\alpha,\nu}, \{\mu_n\}_{n=1}^M),\ m \in \{1, \ldots, M\},$$
$$\alpha \in \{a, b\},\ \nu \in \{O, T\},\ \forall i. \tag{5}$$

Effectively, each representation is assigned a probability distribution. According to (4), the closer the representation to a cluster center with index (say) m, the higher the belief/probability that the corresponding sample should belong to Cluster m.

To facilitate SSL, we now define a target distribution of the probabilities $q_m^{\alpha,T}$ that originate from the target networks described in Section 2. Following (Xie et al., 2016), we set

$$p_{i,m}^{\alpha,T} = \frac{(q_{i,m}^{a,T})^2 / \sum_i q_{i,m}^{a,T}}{\sum_n \left[(q_{i,n}^{a,T})^2 / \sum_i q_{i,n}^{a,T}\right]}. \tag{6}$$

The target distribution is computed by squaring the probability and normalizing it by the frequency of each class. Squaring "hardens" the soft assignments, while frequency normalization penalizes imbalanced clusters.

We can now compare the probabilities $q_{i,m}^{\alpha,O}$ induced by the online networks with the probabilities $p_{i,m}^{\alpha,T}$ of the target

[1] We refer to the composition of the encoder and the projector in the original BYOL work as simply the "encoder" in this paper.

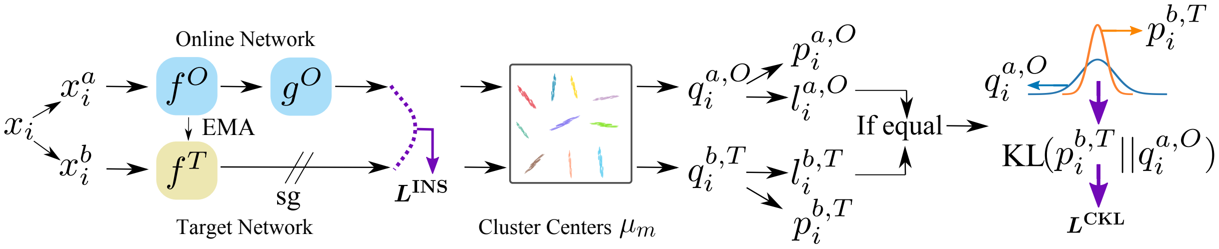


Figure 1. The FedRLC framework during local training. sg means stop gradient. In the figure, we illustrate the construction of the first terms of the symmetric loss function in (8); the second terms are similar.

networks. In this work, we utilize the KL divergence to compare the probability distributions. Letting $\mathrm{KL}(p||q) \triangleq \sum_m p_m \log \frac{p_m}{q_m}$, the crossed KL divergence objective can be defined as

$$L^{\text{CKL}_0} \triangleq \tfrac{1}{N}\textstyle\sum_i\big[\mathrm{KL}(p_i^{b,T}||q_i^{a,O}) + \mathrm{KL}(p_i^{a,T}||q_i^{b,O})\big], \quad (7)$$

where N represents the batch size. The crossed KL objective (7) intends to optimize the local cluster centers by incorporating information from both augmented tviews of the input. The two augmented samples are supposed to share similar probabilities because they are created from the same data under different transformations.

Another novelty that we incorporate in FedRLC is to make sure that the augmentations that are involved in the crossed KL objective in (7) are not too far. Indeed, intuitively, completely irrelevant augmentations would harm, instead of benefit the overall performance. This is why we only incorporate pairs whose hard decisions match in the KL divergence losses. Let $l_i^{\alpha,\nu} = \mathrm{argmax}_m(q_{i,m}^{\alpha,\nu})$ denote the hard clustering decisions of the online and target networks with different augmentations. Ties are broken in favor of the smallest index.

The data is chosen to contribute to the crossed KL divergence loss only when the predicted label from the online and the target networks are the same. We thus modify the loss in (7) to work with

$$\begin{aligned} L^{\text{CKL}} \triangleq \tfrac{1}{N}\textstyle\sum\{\mathrm{KL}(p_i^{b,T}||q_i^{a,O}) : l_i^{b,T} = l_i^{a,O}\}+ \\ \tfrac{1}{N}\textstyle\sum\{\mathrm{KL}(p_i^{a,T}||q_i^{b,O}) : l_i^{a,T} = l_i^{b,O}\}. \quad (8) \end{aligned}$$

As shown in Figure 1, we jointly optimize the cluster centers and the online/target networks during local training. Therefore, the overall loss function is given by

$$L_k = L^{\text{CKL}} + L^{\text{INS}}, \quad (9)$$

where L^{INS} recalls the classical instance-level non-contrastive loss defined in (1). Usually, a hyperparameter can be incorporated to the loss function to control the relative weight of the losses L^{CKL} and L^{INS}. In our experiments, equal weights on the losses already provided a good performance. We thus leave a detailed study on hyperparameter tuning as future work.

3.2. Updates After Server-to-Client Communications

In this part, we describe the cluster center and online network update mechanisms during the server-to-client communications. We use subscript $\star$ to denote the global models, the subscript k to be the local model, and the superscript O to be the online networks. Let r represent the current training round. During the communication update, only the cluster centers and the online network are updated. We now introduce a novel rule to update the centers.

Specifically, given centers $\{\mu_{m,k}^{r-1} \in \mathbb{R}^d\}_{m=1}^M$ in local user k with local data $\mathcal{D}_k$ at round $r-1$, global centers $\mu_{m,\star}^r$ at round r, the centers of Client k at round r are updated according to

$$\mu_{m,k}^r = \tfrac{\epsilon}{1+\epsilon}\mu_{m,k}^{r-1} + \big(1 - \tfrac{\epsilon}{1+\epsilon}\big)\mu_{m,\star}^r, \quad (10)$$

where ϵ is updated progressively by the KL divergence between the probability generated from the local and global centers. Specifically, letting $f_\star^r$ and $f_k^{O,r-1}$ denote the global encoder in round r and the local encoder in round $r-1$ at Client k, respectively, we define $z_{\star,i,k} \triangleq \frac{1}{2}(f_\star^r(x_{i,k}^a) + f_\star^r(x_{i,k}^b))$, $z_{i,k} \triangleq \frac{1}{2}(f_k^{O,r-1}(x_{i,k}^a) + f_k^{O,r-1}(x_{i,k}^b))$ as the mean representations of data $x_{i,k}$ under different augmentations and with global and local networks. We now evaluate the soft class probabilities for the data of Client k according to the global model at Round r as $q_{\star,i,m,k} \triangleq \Delta_m(z_{\star,i,k}, \{\mu_{n,\star}^r\}_{n=1}^M)$. Likewise, we can evaluate the class probabilities according the local model at Round $r-1$ as $q_{i,m,k} \triangleq \Delta_m(z_{i,k}, \{\mu_{n,k}^{r-1}\}_{n=1}^M)$. We can now compute the momentum parameter ϵ via

$$\epsilon = \tfrac{1}{|\mathcal{D}_k|}\textstyle\sum_{i=1}^{|\mathcal{D}_k|} \mathrm{KL}\big(\{q_{\star,i,m,k}\}_{m=1}^M \| \{q_{i,m,k}\}_{m=1}^M\big). \quad (11)$$

When ϵ is large, the divergence between probabilities generated from global and local networks is large, so that the cluster centers inherit more local knowledge. Otherwise, a smaller ϵ gathers more information from global cluster centers.

Finally, we discuss how to update the online networks of the client. For this purpose, we follow the EMA scheme (Zhuang et al., 2022). Specifically, the online networks at Round r are updated as

$$\begin{aligned} (f_k^{O,r}, g_k^{O,r}) \leftarrow \gamma(f_k^{O,r-1}, g_k^{O,r-1})+ \\ (1-\gamma)(f_\star^{O,r}, g_\star c^{O,r}). \quad (12) \end{aligned}$$

In (12), the parameter γ is used to control the weight between the global model and the local model. An explicit formula for γ is given by (Zhuang et al., 2022) $\gamma = \min(\lambda_k||f_\star^r - f_k^{O,r-1}||, 1)$ where $\lambda_k = \frac{\tau}{||f_\star^1 - f_k^0||}$ is a customized magnitude, τ is a tuned hyperparameter, and f is the encoder. In EMA (Zhuang et al., 2022), λ_k is only measured once at the first round. Algorithm 1 in appendix B shows the overall FedRLC scheme.

4. Experiments

Baselines We evaluate FedRLC on linear evaluation and semi-supervised learning tasks. Our baselines include FedU (Zhuang et al., 2021) and FedEMA (Zhuang et al., 2022), which are current state-of-the-art FedSSL methods. We also evaluate FedBYOL, which refers to combining BYOL (Grill et al., 2020) with federated averaging as in (3). Single-Training refers to training each client independently, and the accuracy is calculated by the average of all clients.

Settings: The IID and Non-IID data distribution are exactly followed as FedU and FedEMA, where Non-IID scenarios are splitting the data by class. CIFAR-10 and CIFAR-100 are used in our experiments. To simulate more Non-IID cases, we consider Dirichlet distribution with a parameter β to allocate the data and provide details in appendix C.1. More implementation details are given by appendix C.2.

Linear Evaluation: To validate the quality of learned representations, a linear classifier is trained on top of the frozen representations learned from different FedSSL methods. The results are shown in Tables 1 and 2. FedRLC constantly outperforms other methods, especially for CIFAR-100 with a large number of classes, where it improves by 2.77% and 1.62% on IID and non-IID data, respectively.

Semi-supervised Learning: We compare our model with state-of-the-art works on semi-supervised learning tasks. A new MLP is added on the top of the encoder in semi-supervised learning, and we fine-tune the entire model with 10% labeled data. We compare different federated representation learning methods under IID and Non-IID setting for CIFAR-10 and CIFAR-100 datasets. Tables 3 and 4 demonstrate that our scheme achieves the best results in all cases. In particular, FedRLC improves the performance of CIFAR-100 by 1.68% under a highly heterogeneous scenario.

Table 1. Linear Evaluation: IID & Data-Split Non-IID.

Dataset	CIFAR-10		CIFAR-100	
Method	IID	Non-IID	IID	Non-IID
Single-Training	82.42	74.95	53.88	52.37
FedBYOL	84.29	79.44	54.24	57.51
FedU	83.96	80.52	54.82	57.21
FedEMA	86.26	83.34	58.55	61.78
FedRLC	**87.06**	**84.08**	**61.32**	**63.40**
BYOL (Centralized)	90.46		65.54	

Table 2. Linear Evaluation: Dirichlet Non-IID.

Dataset	CIFAR-10		CIFAR-100	
β	0.5	0.1	0.5	0.1
Single-Training	83.42	83.08	58.45	57.20
FedBYOL	85.44	84.69	59.14	59.93
FedU	85.62	85.33	59.10	58.06
FedEMA	86.12	86.00	60.26	61.46
FedRLC	**86.89**	**86.69**	**62.39**	**63.21**

Table 3. Semi-Supervised Learning: IID & Data-Split Non-IID.

Dataset	CIFAR-10		CIFAR-100	
Method	IID	Non-IID	IID	Non-IID
Single-Training	78.08	69.06	43.50	39.99
FedBYOL	83.24	76.95	49.20	47.07
FedU	82.61	77.06	47.64	46.67
FedEMA	83.38	79.49	49.26	50.48
FedRLC	**83.99**	**79.52**	**49.67**	**52.16**

Table 4. Semi-Supervised Learning: Dirichlet Non-IID.

Dataset	CIFAR-10		CIFAR-100	
β	0.5	0.1	0.5	0.1
Single-Training	81.72	79.89	48.53	49.41
FedBYOL	82.84	82.20	50.00	50.12
FedU	81.33	81.66	49.25	49.31
FedEMA	83.18	82.06	50.11	51.07
FedRLC	**83.41**	**82.73**	**50.41**	**51.19**

5. Conclusions

We have proposed FedRLC, a federated self-supervised representation learning scheme. A key idea of FedRLC is to achieve good representations through the guide and aid of clustering. In particular, FedRLC optimized a crossed KL divergence loss between two augmented data with a data selection mechanism and updated several cluster centers dynamically during communication. Evaluation on the learned image features demonstrated that our approach learned better semantic knowledge of the data compared with other existing FedSSL methods. Moreover, FedRLC has achieved state-of-the-art results on benchmark downstream tasks including linear evaluation and semi-supervised learning.

Acknowledgements

This work was supported in part by the Army Research Lab (ARL) under Grant W911NF-21-2-0272, and in part by the National Science Foundation (NSF) under Grant CNS-2148182.

References

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Huang, Z., Chen, J., Zhang, J., and Shan, H. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Koyuncu, E. Centroidal clustering of noisy observations by using r th power distortion measures. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Le-Khac, P. H., Healy, G., and Smeaton, A. F. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.

Li, Q., He, B., and Song, D. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021a.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8547–8555, 2021b.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Miao, R. and Koyuncu, E. Federated momentum contrastive clustering. *arXiv preprint arXiv:2206.05093*, 2022.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020a.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Wang, L., Zhang, K., Li, Y., Tian, Y., and Tedrake, R. Does learning from decentralized non-iid unlabeled data benefit from self supervision? In *ICLR*, 2023.

Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Zhang, F., Kuang, K., You, Z., Shen, T., Xiao, J., Zhang, Y., Wu, C., Zhuang, Y., and Li, X. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020.

Zhuang, W., Gan, X., Wen, Y., Zhang, S., and Yi, S. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4912–4921, 2021.

Zhuang, W., Wen, Y., and Zhang, S. Divergence-aware federated self-supervised learning. In *ICLR*, 2022.

A. Related Works

A.1. Federated learning

FL has been extensively studied and implemented in various algorithms such as FedAvg (McMahan et al., 2017). FL trains distributed data across different machines due to concerns under security challenges and low-resource computing devices. The main challenge in FL is the inconsistency between local and global objectives caused by the heterogeneity of local data. Numerous FL algorithms (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020; Li et al., 2021a; Wang et al., 2020a; He et al., 2020) try to solve this issue and can be divided into two parts. There are works (Li et al., 2021a; 2020) focusing on the local training phase, and some (Wang et al., 2020a;b) methods study the aggregation stage during communication. For example, FedProx(Li et al., 2020) adds a proximal term to the local training to improve the performance of FedAvg, while FedMA(Wang et al., 2020a) averages weights in a layer-wise fashion based on Bayesian non-parametric model during the aggregation phase. However, these approaches rely on labeled data and are developed for supervised learning.

A.2. Self-supervised representation learning

In the past few years, SSL deals with unlabeled data and archives competitive performance in computer vision tasks (Grill et al., 2020; Chen et al., 2020; Chen & He, 2021; Zbontar et al., 2021; Caron et al., 2020; Le-Khac et al., 2020; Li et al., 2021b; Huang et al., 2022; Caron et al., 2021). Contrastive learning (Chen et al., 2020; Li et al., 2021b) and non-contrastive learning (Grill et al., 2020; Chen & He, 2021) are two popular methods in the community. BYOL-like methods are typically referred to as non-contrastive methods as learning is accomplished through two augmented versions, or positive samples, of the same input. Contrastive methods (Chen et al., 2020), which rely on comparing the input with many negative samples have also been utilized for SSL. Nevertheless, BYOL suffers from learning non-uniform representation (Huang et al., 2022), where data embeddings are not uniformly distributed over various class and not able to be separated well. Contrasitve methods cause class collision issue where negative samples are not truly negative and require large memory. Besides, most popular SSL paradigms (Chen et al., 2020; Grill et al., 2020; He et al., 2020) assume that the data is gathered centrally in a server, unfortunately, which is not suitable when the data is distributed.

A.3. Federated clustering

A recent notable example is FeatARC (Wang et al., 2023), which also utilizes the novel idea of clustering clients for "FL in groups." FeatARC provides experiments in the same setup as in Table 1 for the CIFAR-10 dataset. It achieves an accuracy of %86.74 and %84.63 for the IID and non-IID settings, respectively. FedRLC outperforms FeatARC in the IID case, while it is worse in the non-IID case. We note that FeatARC relies on the significantly more memory and computationally-intensive contrastive-learning methods, and do not provide numerical results for the CIFAR-100 dataset with a large number of classes. Also, the data clustering methodology of FedRLC can be combined with the client clustering method of FeatARC to potentially improve the performance of either method, as will be discussed in a future work. We note that there have been several works on clustering centralized data (Caron et al., 2018; Xie et al., 2016; Li et al., 2021b; Huang et al., 2022; Koyuncu, 2022), which are not immediately applicable to our distributed setting.

A.4. Federated self-supervised representation learning

FedSSL (Zhuang et al., 2022; 2021; Miao & Koyuncu, 2022; Zhang et al., 2020; Wang et al., 2020b) has been developed effectively to take advantage of both FL and SSL by learning useful representation with unlabeled data and training across several local machines with decentralized data. For example, recent state-of-the-art approaches FedU (Zhuang et al., 2021) and FedEMA (Zhuang et al., 2022) directly adapt the centralized scheme BYOL (Grill et al., 2020) to FL. We have showed the comparison between our method FedRLC and these works (Zhuang et al., 2022; 2021) in the experimental section 4. As we have mentioned, training an SSL directly in local machines of FL leads to a performance drop due to the non-uniformity of representations caused by BYOL and non-identity distribution from heterogeneous system.

In this paper, we have demonstrated the proposed framework FedRLC, which is a clustering-based self-supervised learning algorithm specifically designed in FL. In FedRLC, several cluster centers are optimized to learn a more differentiated embedding in local training. During aggregation, we dynamically update cluster centers to balance the divergence between local and global models. The experiments in section 4 have showed that FedRLC outperforms the existing FedSSL methods and achieves state-of-the-art results.

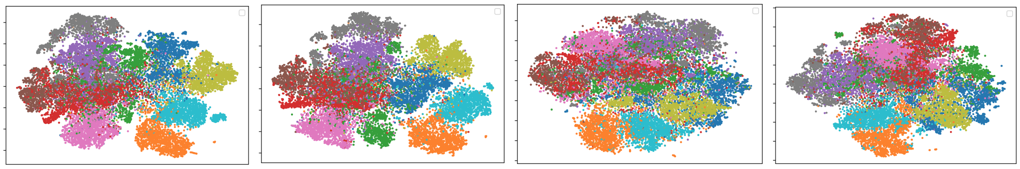
B. FedRLC Algorithm

We summarize the overall FedRLC pipeline in Algorithm1.

Algorithm 1 FedRLC

```
Input: Number of communication rounds $R$, Number of clients $K$, Number of local epochs $E$.
Output: Global encoder $f_\star$ and predictor $g_\star$.
Server executes: Initialize server's network parameters $f_\star$, $g_\star$, and $\mu_{\star,m}$. Have the clients initialize local parameters $f_k^O$, $g_k^O$, and $\mu_{k,m}$
for $r = 1, \ldots, R$ do
    for $k = 1, 2, \ldots, K$ in parallel do
        Send global encoder $f_\star$, predictor $g_\star$, and cluster centers $\mu_{\star,m}$ to client $k$.
        $f_k^O$, $g_k^O$, $\mu_{m,k} \leftarrow$ ClientTraining($f_\star$, $g_\star$, $\mu_{\star,m}$).
    end for
    FedAvg: $(f_\star^O, g_\star^O, \mu_{\star,m}) \leftarrow \sum_k \frac{|\mathcal{D}_k|}{|\mathcal{D}|}(f_k^O, g_k^O, \mu_{m,k})$.
end for
Return global encoder $f_\star$ and predictor $g_\star$.
ClientTraining($f_k^O$, $g_k^O$, $\mu_{k,m}$)
Update the online networks and cluster centers via global parameters by (12) and (10), respectively.
for epochs $= 1, \ldots, E$ and size-$N$ batch learning within each epoch over dataset $\mathcal{D}_k$ do
    Update online networks and cluster centers via global parameters by descending the gradient of the local cost function in (9).
    Update the target network parameters $f_k^T$ via (2).
end for
Return the online networks $f_k^O$ and $g_k^O$.
```

C. Experiment Details



(a) IID FedBYOL (b) IID FedRLC (c) Non-IID FedBYOL (d) Non-IID FedRLC

Figure 2. t-SNE data visualization on CIFAR-10.

C.1. Data heterogeneity

We follow the exact settings of FedU (Zhuang et al., 2021) and FedEMA (Zhuang et al., 2022) for a fair comparison. Namely, to simulate data heterogeneity in federated learning, each user only consists of samples from M/K classes, where M is the number of classes, and K is the number of clients. This is referred to as the data-split scenario. For independent and identically distributed (IID) data, each user has the same number of samples from M classes. In addition to the data-split non-IID scenario, to evaluate on different non-IID scenarios, we sample a specific proportion of the data from class m to client k, where the proportion is followed by the Dirichlet distribution with parameter β, which is also a widely-used method to simulate non-IID data distribution. A smaller β indicates a more heterogeneous distribution. The results have been shown in Tables1 2 3 4 in section 4.

C.2. Implementation details

For federated training, we adopt the SGD optimizer with a 0.032 initial learning rate. The learning rate is decayed by cosine annealing. The batch size is 128, and the input size is 32×32. We use ResNet18 to be the encoder, and the predictor is a two-layer multiplayer perceptron (MLP) with the output dimension 2048. The σ of EMA is 0.99, and the $\tau = 0.7$ is directly followed by (Zhuang et al., 2022) without tuning. We set both the number of local clients and the number of local epochs to 5. The total communication rounds are 100 in federated learning, which are the same as recent FedSSL approaches (Zhuang et al., 2021; 2022) for a fair comparison. For linear evaluation, the AdamW optimizer is adopted with a learning rate of 0.022. The batch size of linear evaluation is 512, and we train the linear classifier for 200 epochs. For semi-supervised evaluation, Adam optimizer is used with a learning rate of 0.001. We fine-tune the entire network for 100 epochs with 10% labeled data.

C.3. Visualization of representations

To analyze the data features visually, we plot the t-SNE visualization of the CIFAR-10 learned from FedBYOL and FedRLC in Figure 2, where different colors indicate different classes. From the comparison between FedBYOL and FedRLC, we observe that the data representations obtained from FedRLC are separated more clearly. The linear and semi-supervised evaluations in section 4 further verify the effectiveness of FedRLC.