2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) | 979-8-3503-0261-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSPW59220.2023.10193449

RESOURCE-EFFICIENT FEDERATED CLUSTERING WITH PAST NEGATIVES POOL

Runxuan Miao and Erdem Koyuncu

Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL

ABSTRACT

Federated learning (FL) provides a global model over data distributed to multiple clients. However, most recent work on FL focuses on supervised learning, and a fully unsupervised federated clustering scheme has remained an open problem. In this context, Contrastive learning (CL) trains distinguishable instance embeddings without labels. However, most CL techniques are restricted to centralized data. In this work, we consider the problem of clustering data that is distributed to multiple clients using FL and CL. We propose a federated clustering framework with a novel past negatives pool (PNP) for intelligently selecting positive and negative samples for CL. PNP benefits FL and CL simultaneously, specifically, alleviating class collision for CL and reducing client-drift in FL. PNP thus provides a higher accuracy for a given constraint on the communication rounds, which makes it suitable for networks with limited communication and computation resources. Numerical results show that the resulting FedPNP scheme achieves superior performance in solving federated clustering problems on benchmark datasets including CIFAR-10 and CIFAR-100, especially in non-iid settings.

***Index Terms*—** Federated learning, unsupervised clustering, contrastive learning, negative sample selection.

1. INTRODUCTION

Federated learning (FL) has shown tremendous success in training distributed data while preserving user privacy [1–3]. However, most previous work on FL focus on supervised learning. Meanwhile, there has been many studies on the classical clustering problem in the centralized setting [4–7]. Nevertheless, obtaining large centralized labeled data is not always practical as labeling data is expensive and some data is sensitive. Existing methods [8–10] explore FL with unlabeled data, but they are primarily designed for linear evaluation or semi-supervised learning with fully or partially labeled data, thus not suitable for clustering tasks. Although some algorithms are introduced for federated clustering, they are only effective on specific datasets such as medical datasets [11] and have difficulties with clustering complex data features [12–15]. Hence, training a more generic deep clustering model with FL is still an open research direction that has received little attention in the literature.

One major concern in FL is that heterogeneous data over multiple users may push the local models far from the global average, resulting in the so-called client drift problem during local training. To address this issue, many supervised FL algorithms are developed to minimize divergence between the local and global models [1, 2]. Besides, in recent years, self-supervised representation learning has been applied to FL [9, 10, 16]. The FedEMA [10] scheme shows that retaining local information is crucial for federated self-supervised learning (FedSSL) as well, and proposes a divergence-aware dynamic update during communication. In fact, reducing the effect of local-global model divergence and retaining more local knowledge during local client training are two contradictory goals, although both are crucial for the ultimate performance of the FL model. In this paper, we design a fully-unsupervised federated clustering framework that can be developed over limited communication resources based on contrastive learning (CL).

In centralized settings, contrastive learning (CL) has been well-studied in self-supervised representation learning [17–19] and deep clustering [5,20,21], but it suffers from class collision, where positive samples from the same class with the given input are still viewed as negatives in contrastive loss. Two main directions of study are designed to resolve this problem: Sampling positives [5, 20] and selecting negatives [22, 23]. For example, GCC [20] and WCL [21] expand positive samples by building graphs. MoCHi [23] chooses the hardest negatives by sorting the instance similarity with the given query based on the dot product similarity. However, sorting high dimensional instance embeddings and constructing graphs [20, 24] consume time and computational resources. Moreover, these strategies assume that the data is centrally available, and they are thus not suitable for IoT devices.

In this work, we propose a federated clustering scheme with a novel past negatives pool (PNP) to overcome the above limitations. We refer to the resulting framework as FedPNP. The PNP, which is generated via the past representations, determines the indices of the positive and negative samples for the current representations. Specifically, we calculate the Gaussian similarity between pseudo labels learned from the past local model in last communication round. We select negatives and positives by using a threshold for contrastive clustering. As mentioned before, local training causes weight divergence, but FedSSL requires local knowledge. The proposed PNP uses the previous cluster features to avoid the drift during current local updates and effectively incorporate more local information required by FedSSL. The negatives selection technique helps for class collision issue in CL by comparing the similarity between soft labels and removing positives from negatives when conducting the contrastive loss, which differs from existing strategies only handling instance representation with centralizing data.

In summary, FedPNP develops a deep clustering model that can be trained over distributed networks without sharing the

This work was supported in part by the Army Research Lab (ARL) under Grant W911NF-2120272 and in part by National Science Foundation (NSF) under Grant CNS-2148182.

private data. To emphasize our points, existing deep clustering methods can only be used under the assumption that the data is collected centrally. Our goal is to learn a clustering model in a federated manner, which is more challenging and practical when the data is distributed on edge computing devices. Our experiments show that the proposed FedPNP has significantly more stable and accurate performance on CIFAR-10 and CIFAR-100 compared to other methods in federated clustering in both IID and Non-IID settings. Since FedPNP can achieve superior performance in fewer rounds of training, it is especially suitable for edge devices that may be limited in terms of computation power and communication rates.

The rest of the paper is organized as follows: In Section 2, we start with stating the problem and then introduce the proposed FedPNP. In Section 3, we show our numerical results. Finally, we state our main conclusions in Section 4.

2. MODEL

In this part, we first define the problem, and then we present the proposed FedPNP to address the problem.

2.1. Contrastive Clustering (CC)

We begin with an overview of CC [6]; a detailed description can be found in [6] as well as Section 2.3. Two augmented views from the same sample pass through the same encoder to generate two representation vectors. An instance-level multilayer perceptron (MLP) and a cluster-level MLP process these two representation features simultaneously to provide a pair of instance-level projections and a pair of cluster-level projections, where the contrastive loss is applied jointly. Once well-trained, the cluster representations directly correspond to soft cluster assignments or labels. However, under the concerns of data privacy and limited computation resources, it is not practical to always assume that the data is centrally available. As a result, clients cannot send their local data to a central sever to train a model in federated learning.

2.2. The FedPNP Framework

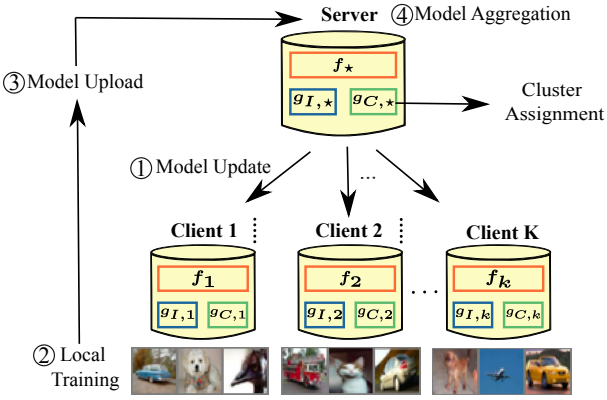

Fig. 1: The federated clustering scheme. At the client k, f_k, $g_{I,k}$, $g_{C,k}$ represent base encoder, instance-level projector, and cluster-level projector, respectively.

We aim at addressing the above issue by building on the CC scheme with a novel PNP in the federated setting. We aggregate several local models trained in a fully unsupervised federated way to obtain a global model that outputs the cluster information directly. In particular, suppose there are K clients, where client k has its local unlabeled data $\mathcal{D}_k$. Our goal is to learn a machine model over the dataset $\mathcal{D} \triangleq \bigcup_{k=1}^{K} \mathcal{D}_k$ on the central server that can cluster a group of unlabeled data automatically. The overall block diagram of the FedPNP architecture is shown in Fig. 1. The central server contains global networks $f_\star$, $g_{I,\star}$, and $g_{C,\star}$, representing a base encoder, an instance-level projector, and a cluster-level projector, respectively. Let $f_k, g_{C,k}, g_{I,k}$ denote the network elements at user k. In each communication round, the central server sends global networks to local clients. Each local device updates the local model using its own local data and sends the updated model to the sever. The server updates the global networks by weighted averaging. Finally, the clustering assignments can be obtained from the global cluster-level projector.

2.3. PNP and PNP-Contrastive Loss

The proposed FedPNP relies on contrastive representation learning [6, 17, 19], and specifically CC [6], as outlined in Section 2.1. Simply extending the CC scheme to federated setting results in poor performance we show in Section 3. This is because: 1) The Non-IID data over multiple users causes the client-drift during local training. 2) FedSSL needs to store more local information that may lose during fast aggregation in FL, which leads to poor representations. 3) Negative samples in contrastive loss causes class collision. Hence, we introduce the PNP for intelligently selecting negative samples in CL. Intuitively, in terms of the above issue 1) and 2), reducing client-drift contradicts with keeping more local data features. The PNP is designed to optimize the trade-off between these two demands. The idea is we remove the potential positive samples by comparing soft labels produced from the past local models. We utilize features learned from the past to remain more local knowledge and avoid the client-drift during the current local training. Moreover, we compare the similarity between soft labels to select potential positives with the given input image and remove it from the large negatives in contrastive loss to alleviate class collision issue.

In detail, let $\mathcal{D}_k \triangleq \{x_{1,k}, \ldots, x_{|\mathcal{D}_k|,k}\}$, $k = 1, \ldots, K$ represent the local datasets of the users. For each user $k \in \{1, \ldots, K\}$, given a local data $x_{i,k} \in \mathcal{D}_k$, two samples $x_{i,k}^a \triangleq t^a(x_{i,k})$ and $x_{i,k}^b \triangleq t^b(x_{i,k})$ are first created through transformations t^a and t^b, respectively. We use the variable $\sigma \in \{a, b\}$ to represent the sample index so that the transformations are succinctly expressed as $x_{i,k}^\sigma \triangleq t^\sigma(x_{i,k})$, $\sigma \in \{a, b\}$. The transformations are sampled uniformly at random from a family $\mathcal{T}$ of augmentations, which may include rotations, noise, etc. In FedPNP, two augmented samples pass through not only local models in current communication but also local models in the previous round as shown in Fig 2. From local networks consisting of a base encoder f_k, a instance-level MLP $g_{I,k}$, and a cluster-level MLP $g_{C,k}$, we can obtain instance-level and cluster-level representations learned from current local models in communication round r as

$$z_{i,k}^{\sigma,r} \triangleq g_{I,k}^r(f_k^r(t_k^\sigma(x_{i,k}))) \in \mathbb{R}^{d_1}, \tag{1}$$

$$y_{i,k}^{\sigma,r} \triangleq g_{C,k}^r(f_k^r(t_k^\sigma(x_{i,k}))) \in \mathbb{R}^{d_2}, \sigma \in \{a, b\}. \tag{2}$$

The output dimensionality d_2 of the cluster-level representations is chosen to be equal to the number of clusters one

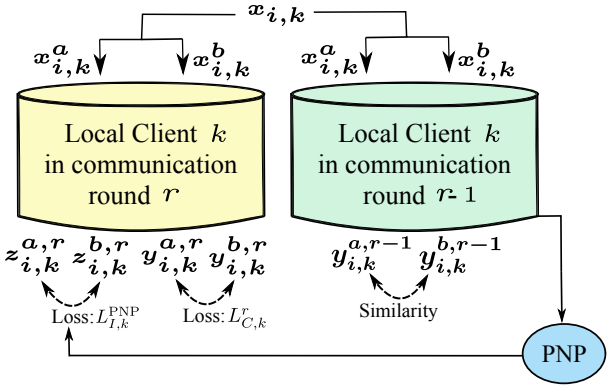

Fig. 2: The local training at client k for FedPNP

wishes to find in the dataset. In many cases, the instance-level output dimensionality d_1 is chosen to be much larger than d_2. Specifically, $z^{\sigma,r}_{i,k}$ represents the instance features of $x_{i,k}$, and $y^{\sigma,r}_{i,k}$ is considered as the soft labels of $x_{i,k}$. In a deterministic assignment of inputs to clusters, $y^{\sigma,r}_{i,k}$ would be one-hot encoded vectors. To construct the PNP, we compute the Gaussian similarity between the soft labels extracted from the past local model in communication round $r-1$. The Gaussian similarity measure is defined as $s(u,v) \triangleq \exp(u^\dagger v)$. Then, the PNP for a given augmented data $x^a_{i,k}$ is created by

$$P_i = \{j \neq i : s(y^{a,r-1}_{i,k}, y^{\sigma,r-1}_{j,k}) < \mu,\ \forall \sigma \in \{a,b\}\}, \quad (3)$$

where μ is the threshold to select negative and positive samples, $y^{\sigma,r-1}_{i,k}$ represents the soft label extracted from the cluster-level MLP in the previous round $r-1$. Hence, sufficiently close samples form positive pairs while far samples are negatives. The key idea is that the indices P_i for negative pairs are obtained from the past model in FedPNP. This allows preserving local information, a key requirement in FedSSL.

Given batch size n, matrices $\mathbf{u} = [u_1 \cdots u_n] \in \mathbb{R}^{d\times n}$ and $\mathbf{v} = [v_1 \cdots v_n] \in \mathbb{R}^{d\times n}$ constructed via the indicated column vectors, the ordinary contrastive loss [17] is defined by

$$L(\mathbf{u},\mathbf{v};\tau) \triangleq \frac{1}{n}\sum_{i=1}^{n} -\log \frac{s_\tau(u_i,v_i)}{\sum_{\substack{j=1\\ j\neq i}}^{n}[s_\tau(u_i,u_j)+s_\tau(u_i,v_j)]}, \quad (4)$$

where $s_\tau(u,v) \triangleq \exp(\frac{1}{\tau}u^\dagger v/(\|u\|\|v\|))$ is a normalized Gaussian similarity measure, and $\tau > 0$ is a temperature parameter. In (4), given an augmented input image $x^a_{i,k}$, we consider the $x^b_{i,k}$ as its positive pair, and all other augmented samples from the batch are negatives. In particular, the decision whether a given sample is positive or negative is done according to the current model weights. To preserve more local information, we use the idea of PNP, and choose the negative samples over the indices described by the set (3) instead. This results in the PNP-contrastive loss

$$L^{\text{PNP}}(\mathbf{u},\mathbf{v};\tau) \triangleq \frac{1}{n}\sum_{i=1}^{n} -\log \frac{s_\tau(u_i,v_i)}{\sum\limits_{j\in P_i}[s_\tau(u_i,u_j)+s_\tau(u_i,v_j)]}. \quad (5)$$

For FedSSL, the PNP selects negatives from the past local models, which alleviates the client-drift during the current local update and maintains more local knowledge that is forgotten during the model aggregation. For contrastive representation learning, the PNP computes similarity between soft labels extracted from the past and removes samples that may have the same class category with the given input, which alleviates the class collision issue in traditional contrastive loss. In short, the PNP is beneficial in three aspects. 1) It avoids the large divergence of Non-IID networks updated locally in current communication round. 2) It keeps more local knowledge, which can be lost during model aggregation, benefiting FedSSL. 3) It helps the class collision issue in traditional CL by removing potential positives from negative samples.

2.4. Local Training

We now describe the training procedure at each client. We minimize the PNP-contrastive loss (5) on instance representation and ordinary contrastive loss (4) on both past and current cluster features. Formally, given a batch size n, the instance-level PNP-contrastive loss at user k is defined via the instance-level representations $\mathbf{z}^{\sigma,r}_k \triangleq [z^{\sigma,r}_{1,k} \cdots z^{\sigma,r}_{n,k}] \in \mathbb{R}^{d_1\times n}$, $\sigma \in \{a,b\}$ as

$$L^{\text{PNP}}_{I,k} \triangleq L^{\text{PNP}}(\mathbf{z}^{a,r}_k, \mathbf{z}^{b,r}_k; \tau_I), \quad (6)$$

where $\tau_I > 0$ is the instance-level temperature. On the other hand, given $\mathbf{c}^{\sigma,r}_k \triangleq [y^{\sigma,r}_{1,k} \cdots y^{\sigma,r}_{n,k}]^\dagger \in \mathbb{R}^{n\times d_2}$, $\sigma \in \{a,b\}$, we define the cluster-level contrastive loss at round r via (4) as

$$L^r_{C,k} \triangleq L(\mathbf{c}^{a,r}_k, \mathbf{c}^{b,r}_k; \tau_C) + H(\mathbf{c}^{a,r}_k) + H(\mathbf{c}^{b,r}_k), \quad (7)$$

where $\tau_C > 0$ is the cluster-level temperature parameter, and for any matrix $\mathbf{u} = [u_1 \cdots u_d] \in \mathbb{R}^{n\times d}$, the entropy $H(u)$ is defined as $H(u) \triangleq -\sum_{i=1}^{d} \frac{\|u_i\|_1}{\|\mathbf{u}\|_1} \log \frac{\|u_i\|_1}{\|\mathbf{u}\|_1}$. As discussed in [6], entropy regularization helps avoid the trivial solution where all samples are assigned to the same cluster. The cluster-level contrastive loss differentiates clusters. Specifically, columns of $\mathbf{c}^{a,r}_k$ and $\mathbf{c}^{b,r}_k$ represents individual clusters.

We combine the different performance measures described above into the overall loss function

$$L_k \triangleq L^{\text{PNP}}_{I,k} + \alpha L^r_{C,k}, \quad (8)$$

where α is the hyperparameter used to control the weight of the loss. The dependencies between the different losses are illustrated in Fig. 2. Note that the first term in (8) depends on the parameters of both the current and the past network, while the second term depends only on the current parameters. To minimize the loss (8) in practice, we select the negatives from the current models only for $r = 1$ as there is no previous models in the first round. In the first round, the PNP is defined as $P_i = \{j \neq i : s(y^{a,r}_{i,k}, y^{\sigma,r}_{j,k}) < \mu,\ r = 1,\ \forall \sigma \in \{a,b\}\}$.

3. EXPERIMENTS

3.1. Experiment Setup

Datasets and Settings: For fair comparisons, we follow the recent clustering works [5,6,25] to evaluate our clustering performance in terms of clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI). We perform different federated clustering algorithms on datasets including CIFAR-10 and CIFAR-100. CIFAR-10 has 10 classes, and CIFAR-100 has 100 classes. We use the 50,000 training set for training and all the images for testing clustering performance on CIFAR-10 and CIFAR-100.

Table 1: Clustering accuracy (%) on CIFAR-10 and CIFAR-100 datasets. (In NCC, only 20 super-classes for CIFAT-100 is used.)

Dataset	IID						Non-IID ($\beta = 0.5$)						Non-IID ($\beta = 0.1$)					
	CIFAR-10			**CIFAR-100**			**CIFAR-10**			**CIFAR-100**			**CIFAR-10**			**CIFAR-100**		
Method	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
Single-Training	45.7	56.1	33.1	34.1	16.3	8.6	40.6	46.4	25.7	33.9	16.2	8.4	34.8	40.4	20.1	33.8	15.2	7.3
FedBYOLCC	53.8	62.7	44.9	33.2	15.7	7.9	41.3	45.7	28.3	33.9	16.3	8.3	35.5	40.4	22.9	31.9	14.9	7.2
FedCC	54.9	64.9	46.3	34.1	16.3	8.4	41.2	44.7	27.5	34.4	16.5	8.9	33.2	38.0	20.3	33.0	15.0	7.3
FedPNP (ours)	**56.8**	**66.5**	**47.1**	**34.7**	**17.1**	**9.0**	**42.7**	**49.5**	**30.5**	**34.5**	**17.0**	**8.9**	**36.4**	**43.7**	**24.2**	**34.1**	**16.0**	**7.9**
NCC [5] (Centralized)	88.6	94.3	88.4	-	-	-	88.6	94.3	88.4	-	-	-	88.6	94.3	88.4	-	-	-

For simulating Non-IID data in federated learning, we follow previous studies [8–10] to allocate the instances of class j to client k in a proportion of $p_{j,k}$ followed by the Dirichlet distribution $Dir_N(\beta)$ ($\beta = 0.5$ by default). A larger β causes more balanced distributions. For IID cases, each client has the same number of samples for all classes, which is also the same as the recent works [8–10, 26]

Implementation Details: In our experiments, we use ResNet-18 [27] as the base encoder. We use Adam optimizer [28] with an initial learning rate of 0.0003 and without weight decay. All input images are resized to 224×224, and the batch size n is set to 128. The output dimension of the instance-level MLP is set to 128, and the feature dimension of the cluster-level MLP is equal to the number of clusters. The instance-level temperature is $\tau_I = 0.5$, and the cluster-level temperature is $\tau_C = 1.0$. In FedPNP, μ is set to 0.999 for selecting negatives. We set the hyper-parameters $\alpha = 2$ for the first round $r = 1$ and $\alpha = 0.1$ starting from the second round.

Baselines: We compare the FedPNP with several existing representation learning methods for clustering training on distributed data. 1) Single-Training: each client is trained locally by performing the popular scheme SimCLR [17] without federated averaging. 2) FedBYOLCC: we combine BYOL [18] and CC [6] to train the model by minimizing MSE loss in BYOL and cluster-level contrastive loss in CC. 3) FedCC: we simply extend the CC to federated settings. Also, we consider the centralized clustering scheme NCC [5] as an upper bound.

Evaluation: For all experiments trained in federated learning, we train the model for 100 communication rounds for $K = 5$ clients. For each communication round, each client is trained for $E = 5$ local epochs. For the Single-Training experiment, we train each client 300 epochs and report the mean clustering accuracy among all 5 clients by K-means. For all other federated clustering methods, we show the performance based on the cluster assignment from the global cluster-level MLP.

3.2. Federated Clustering Results

Table 1 shows the proposed FedPNP constantly outperforms other methods and achieves the best clustering performance in all data distribution settings. In IID cases, we improve the clustering accuracy by 10.4%, 3.7%, and 2% when comparing with baselines on CIFAR-10. Compared to simply doing a CC framework in Non-IID setting, we improve the clustering accuracy by 4.6% and 5.7 % for $\beta = 0.5$ and $\beta = 0.1$, respectively. For CIFAR-100 with 100 clusters, FedPNP is still the best approach for dealing with such large number of classes. Figs. 3, 4 show the overall clustering performance during communication rounds for $\beta = 0.5$ and $\beta = 0.1$, respectively. FedPNP outperforms all other methods, achieving higher accuracy in fewer communication rounds. We note that fewer rounds translate to fewer amount of computations, which is an important gain for resource-limited edge devices. Another byproduct of fewer rounds is reduced communication latency, especially when the client-to-server communication rates are low [29,30]. FedPNP thus offers significant advantages for both power and communication-limited edge devices.

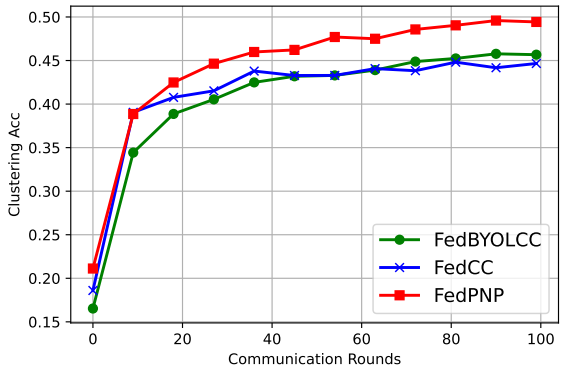

Fig. 3: The FedPNP for $\beta = 0.5$

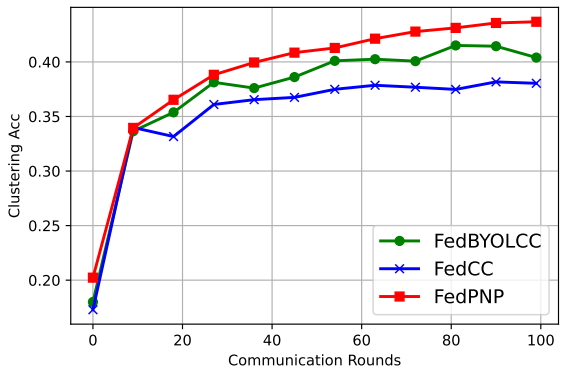

Fig. 4: The FedPNP for $\beta = 0.1$

4. CONCLUSION

In this work, we aim to solve the challenging task on clustering unlabeled data in federated learning. Our focus is on federated learning at the edge, which may be limited in terms of communication capabilities as well as computation power. We propose FedPNP, a fully unsupervised federated deep clustering model that is more suitable and robust for clustering in resources-limited devices. A novel past negatives pool is introduced in FedPNP to reduce client-drift in federated learning and alleviate class collision in contrastive learning.

5. REFERENCES

[1] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *ICML*, 2020.

[2] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *AISTATS*, 2017.

[4] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[5] Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan, "Exploring non-contrastive representation learning for deep clustering," vol. abs/2111.11821, 2021.

[6] Yunfan Li, Peng Hu, Jerry Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng, "Contrastive clustering," in *Thirty-Fifth AAAI*, 2021.

[7] Erdem Koyuncu, "Centroidal clustering of noisy observations by using rth power distortion measures," *accepted to IEEE Transactions on Neural Networks and Learning Systems*, June 2022.

[8] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li, "Federated unsupervised representation learning," *CoRR*, vol. abs/2010.08982, 2020.

[9] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi, "Collaborative unsupervised visual representation learning from decentralized data," *CoRR*, vol. abs/2108.06492, 2021.

[10] Weiming Zhuang, Yonggang Wen, and Shuai Zhang, "Divergence-aware federated self-supervised learning," in *ICLR*, 2022.

[11] Li Huang, Andrew L. Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, pp. 103291, 2019.

[12] Shuai Wang and Tsung-Hui Chang, "Federated clustering via matrix factorization models: From model averaging to gradient sharing," *CoRR*, vol. abs/2002.04930, 2020.

[13] Inderjit S Dhillon and Dharmendra S Modha, "A data-clustering algorithm on distributed memory multiprocessors," in *Large-scale parallel data mining*, pp. 245–260. Springer, 2002.

[14] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii, "Scalable k-means++," *arXiv preprint arXiv:1203.6402*, 2012.

[15] Tayfun Kucukyilmaz, University of Turkish Aeronautical Association, et al., "Parallel k-means algorithm for shared memory multiprocessors," *Journal of Computer and Communications*, vol. 2, no. 11, pp. 15, 2014.

[16] Runxuan Miao and Erdem Koyuncu, "Federated momentum contrastive clustering," *arXiv preprint arXiv:2206.05093*, 2022.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th ICML*, 2020.

[18] 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 *NeurIPS*, 2020.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, June 2020.

[20] Huasong Zhong, Jianlong Wu, Chong Chen, Jianqiang Huang, Minghua Deng, Liqiang Nie, Zhouchen Lin, and Xian-Sheng Hua, "Graph contrastive clustering," in *ICCV*, October 2021, pp. 9224–9233.

[21] Mingkai Zheng, Fei Wang, Shan You, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu, "Weakly supervised contrastive learning," in *ICCV*, October 2021, pp. 10042–10051.

[22] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka, "Contrastive learning with hard negative samples," in *ICLR*, 2021.

[23] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus, "Hard negative mixing for contrastive learning," in *NeurIPS*, 2020, vol. 33.

[24] Hongyi Pan, Diaa Badawi, Runxuan Miao, Erdem Koyuncu, and Ahmet Enis Cetin, "Multiplication-avoiding variant of power iteration with applications," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5608–5612.

[25] Yuming Shen, Ziyi Shen, Menghan Wang, Jie Qin, Philip H. S. Torr, and Ling Shao, "You never cluster alone," *CoRR*, vol. abs/2106.01908, 2021.

[26] Qinbin Li, Bingsheng He, and Dawn Song, "Model-contrastive federated learning," in *CVPR*, June 2021.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on CVPR, 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

[28] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] Pengzhen Li, Erdem Koyuncu, and Hulya Seferoglu, "Respipe: Resilient model-distributed dnn training at edge networks," in *ICASSP*. IEEE, 2021, pp. 3660–3664.

[30] Pengzhen Li, Hulya Seferoglu, Venkat R Dasari, and Erdem Koyuncu, "Model-distributed dnn training for memory-constrained edge computing devices," in *2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 2021, pp. 1–6.