UFed-GAN: A Secure Federated Learning Framework with Constrained Computation and Unlabeled Data

Achintha Wijesinghe, Songyang Zhang, *Member, IEEE*, Siyu Qi and Zhi Ding *Fellow, IEEE,*

***Abstract*—To satisfy the broad applications and insatiable hunger for deploying low latency multimedia data classification and data privacy in a cloud-based setting, federated learning (FL) has emerged as an important learning paradigm. For the practical cases involving limited computational power and only unlabeled data in many wireless communications applications, this work investigates FL paradigm in a resource-constrained and label-missing environment. Specifically, we propose a novel framework of UFed-GAN: Unsupervised Federated Generative Adversarial Network, which can capture user-side data distribution without local classification training. We also analyze the convergence and privacy of the proposed UFed-GAN. Our experimental results demonstrate the strong potential of UFed-GAN in addressing limited computational resources and unlabeled data while preserving privacy.**

***Index Terms*—Federated learning, unlabeled data, data privacy, generative adversarial networks.**

I. Introduction

THE burgeoning rise of deep learning has shown remarkable achievements in learning, based on the often voluminous amounts of data for centralized training. However, in many cases of learning-based wireless connections for collaboration, decentralized learning is vital to handle the heterogeneous data distribution among nodes (users). Importantly, privacy concerns and resource limitations also prevent direct data sharing. To ensure data privacy and communication efficiency, federated learning (FL) [1] has emerged as an important framework to disengage data collection and model training via local computation and global model aggregation.

Despite reported successes, existing FL frameworks have certain limitations. One major obstacle of FL in practice is the heterogeneity of data distribution among participating FL users. It is known [2] that the accuracy of classic FL frameworks such as FedAvg [1] could drop by 55% for some datasets showing non-IID, i.e., not identically and independently distributed, data distributions. To combat performance loss against non-IID datasets, the more general approach of FedProx [3] may depend on certain unrealistic dissimilarity assumptions of local functions [4]. Alternatively, generative adversarial network (GAN) [5] provides another approach to address data heterogeneity. In GAN-based FL, GAN models are used as a proxy to share user updates without training the global model on the user side. For example, in [6], a global classifier is trained using a user-shared generator of the user-end-trained conditional GANs (cGANs). Another example is [7], the authors proposed to share the full user-end GAN for generating a synthetic dataset. However, the high communication cost and the potential privacy leakage hinder the performance gain of these known GAN-based FL frameworks as discussed in [8]. Moreover, the training of the entire GAN and other learning models sometimes can be impractical at the user end, especially for nodes with limited computation resources, such as computation-constrained sensors and devices [9]. How to develop a more efficient GAN-sharing strategy to preserve privacy and handle limited computation remains an open question in generative FL.

In addition to data heterogeneity and limited computation resources, most of the existing works focus on supervised FL, where the performance depends heavily on the availability of labeled training data. In practical applications such as user clustering and video segmentation [10], the computational and privacy limitations may prevent user-side data labeling. Therefore, learning from unlabeled data is equally important for FL to reach its full potential. Presently, only limited research works have specifically addressed FL with unlabeled data, primarily due to inherent challenges. A typical category of FL dealing with unlabeled data [11], [12] focuses on clustering tasks, which may limit its generalization to other deep learning tasks. Another line of FL focuses on unsupervised representation learning [13], where knowledge distillation and contrast learning are applied to address heterogeneous user data distributions. Other FL works on unlabeled data [14]–[16] leverage the efficiency of latent space and may lack a general description of the original data. As aforementioned, GAN-based approaches can be intuitive solutions to capture the data distributions and assist further applications, even without annotated labels.

In this work, we develop a novel FL framework, the Unsupervised Federated Generative Adversarial Network (UFed-GAN), for resource-limited distributed users without labeled data. The novelty is an innovative GAN-based FL and data-sharing strategy to significantly reduce the computational cost at the user end and to preserve privacy. Note that, instead of focusing on one specific unsupervised learning task, we provide a general FL scheme to learn the data distributions without labels. Our framework can be easily adapted to handle specific learning tasks, including unsupervised representation learning, user clustering, and semi-supervised classification.

Our contributions can be summarized as follows:

- We propose a novel UFed-GAN as an FL framework to

A. Wijesinghe, S. Qi, and Z. Ding are with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616. (E-mail: achwijesinghe@ucdavis.edu, syqi@ucdavis.edu, and zding@ucdavis.edu).

S. Zhang was with the University of California at Davis, Davis, CA, 95616 and now is with the Department of Electrical and Computer Engineering, University of Louisiana at Lafayette, Lafayette, LA 70504 (E-mail: songyang.zhang@louisiana.edu).

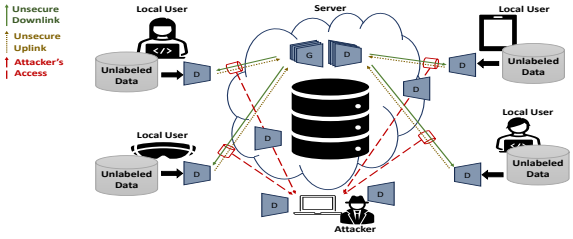

Fig. 1. UFed-GAN in a distributed learning setup in an untrustworthy communication scenario, where users are assumed with less computational power. An attacker may eavesdrop to understand the user data.

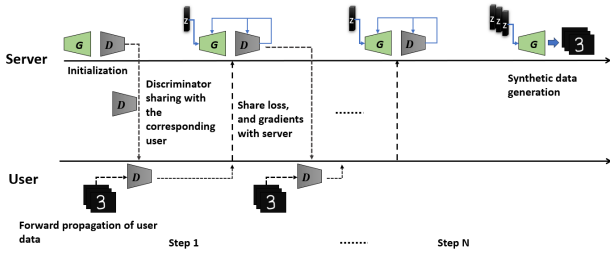

Fig. 2. Communication rounds till the convergence of UFed-GAN. We use the inception score (IS) as the measure of convergence.

learn and characterize the non-IID user data distributions from unlabeled user data. Our UFed-GAN captures the underlying user data distributions without explicitly training a local GAN model for each user, thereby significantly lowering the computational cost on the user side. To our best knowledge, this is the first work to address such constrained computation in GAN-based FL.

- We analyze the convergence of UFed-GAN and prove that privacy leakage can be prevented by our UFed-GAN, in comparison to traditional GAN-based FL.
- Our experimental results in several benchmark datasets demonstrate the performance of UFed-GAN in a semi-supervised classification setup.

In terms of organization, we first introduce the architecture of UFed-GAN and a training strategy in Section II. Following the study on model convergence and the privacy analysis in Section III. We present the experimental results of UFed-GAN on several well-known datasets in Section IV. We provide concluding remarks in Section V.

II. Method and Architecture

A. Problem Setup

As a typical example of a resource-limited FL setup in Fig. 1, a server aims to learn from user nodes, each with limited computational resources whereas attackers may attempt to eavesdrop on the network links. Users may not be able to annotate their raw data. Moreover, since the target tasks may be different among users and the data may also be skewed, a non-IID data distribution shall be considered in this scenario. Different from local users, the server has sufficient computational resources to obtain a model that learns a global data distribution from all the local data, without initial training data. Such a setup is applicable in many distributed learning scenarios. For example, a distributed camera/sensor system placed for object detection can benefit from the collaboration of different cameras for better feature extraction, where each digital camera or sensor may have limited computation power.

To demonstrate the privacy protection offered by UFed-GAN, we consider an attacker that has access to the vulnerable communication links between distributed users and the central server. Such attacks try to gain users' data features based on the information shared through the channels.

Note that, we aim to develop a novel data/model-sharing strategy for FL. The strategy could handle unlabeled data and capture user data distributions in the scenario with limited local computation resources. The proposed framework should offer flexible integration with various unsupervised and semi-supervised learning tasks, such as latent representation learning, user clustering, and semi-supervised classification [13].

B. UFed-GAN

We now explain the framework and training process of UFed-GAN, which allows private and secure learning from unlabeled data in a distributed learning setup. Our proposed UFed-GAN aims to train a GAN model on the server to capture the underlying user data distributions without implementing complex GAN training on the user side.

First, for each user u, we initiate a GAN model on the server, including a generator (G_u) and a discriminator (D_u). Due to label inaccessibility, we select deep convolutional GANs (DCGANs) [17] as the backbone. The details of choices for GAN architecture will be elaborated in Section II-D.

The GAN training comes in three steps, which contains two steps of D training and one step of G training. We split the dual-step D training into the server side and the user side. On one side, the server initiates a discriminator D_u and then shares the initiated model with the corresponding user u. Subsequently, on the other end, the user performs a forward pass (FP) on D_u using a single batch of real local data I_u. The gradients and loss are calculated and then shared with the server. Compared with training a full GAN on the user side, this step can significantly reduce the computational cost and be easily deployed in computation-constrained devices. Upon receiving the updated information of D_u from the user u, the server completes the training of D_u. It generates a noise vector $\mathbf{z}$ to pass through G_u. Finally, synthetic data $G_u(\mathbf{z})$ is generated from the generator. $G_u(\mathbf{z})$ is then sent to D_u to calculate the corresponding gradients and loss. These received gradient updates are combined with the gradient updates of the previous step and then back-propagated through D_u to update the parameters.

Similar to conventional GAN training, starting with a noise vector $\mathbf{z}$', we train G_u with forward- and backward-propagation to update its parameters. In each communication round, the above process repeats until GAN convergence to a favorable point. We illustrate this training process further in Fig. 2. For model convergence monitoring and the stopping criterion, we use inception score (IS) [18], which measures the characteristics of the generated images. After convergence, we create a synthetic dataset using the trained D_u. With

the generated synthetic dataset, we can design corresponding unsupervised or semi-supervised algorithms to implement specific learning tasks. For example, in a semi-supervised classification task, we could apply the MoCo [19], which uses dictionary lookups in contrastive learning to obtain the global classifier. The pseudocode of the proposed training strategy is presented in Algorithm 1.

Algorithm 1 UFed-GAN: Training Algorithm

for each user u **do**
  Server initialization of G_u and D_u
end for
for each communication round r, until the GAN convergence **do**
  for each step t **do**
    Share D_u with user u.
    Perform FP in D_u with user data $\mathcal{I}_u$ and share the gradient updates ∇W_u.
    Perform FP in D_u with fake data $G(z_t)$ where z_t is a random noise vector and get the gradient updates ∇W_z.
    Update D_u with $(\nabla W_u + \nabla W_z)$.
    Train G_u with trained D_u and random noise vectors.
  end for
  Generate an unlabeled dataset using each G_u.
end for

C. Attacker Model

We now introduce the attacker model to quantify the privacy leakage of UFed-GAN. It is highly challenging, if not impossible, to obtain access to the global generator G given a secure server or to guess the exact architecture of G, regardless of the computation prowess of the attacker. Therefore, as suggested in [8], we focus on a reconstruction attack [20] in this work, where an attacker attempts to reconstruct the training data. Let θ be the parameters released to the communication channel by the user. We denote the releasing mechanism by $\mathcal{M}$ and the information that an attacker $\mathcal{A}$ could obtain by $\mathcal{M}(\theta)$. According to the UFed-GAN framework, $\mathcal{M}(\theta)$ is the discriminator gradients and the loss values. Let $\mathcal{I}$ represent the reconstructed data, we have

$$\mathcal{A} : \mathcal{M}(\theta) \mapsto \mathcal{I}. \tag{1}$$

Suppose that the generator $G_{\mathcal{A}}$ of attacker $\mathcal{A}$ has the same architecture as G but with initial weights $W_{\mathcal{A}}$ different from those of G, represented by W_G. In parallel to the server training, we train $G_{\mathcal{A}}$ at the attacker's end.

D. GAN Architecture

Due to label inaccessibility and resource constraints, we adopt the unsupervised DCGANs [17] over cGANs [8] which saves extra computation needs for any pseudo-labeling. The generator architecture follows five transposed convolutional layers. The first layer takes an input with 100 channels and maps it to 1024 channels. Every subsequent layer reduces the number of channels by half. Every layer uses ReLU activation except for the Tanh activation for the final layer. All the layers in both the generator and the discriminator use 4×4 kernels and batch normalization for each layer before the final layer. For the discriminator model, we use four convolutional layers. The first layer accepts similar channel sizes of the data samples and maps to 256 channels. Every subsequent layer doubles the number of channels except the final layer, which outputs a single channel. The final layer uses a Sigmoid activation whereas all other layers use LeakyReLU activation.

III. Convergece and Privacy Analysis of UFed-GAN

In this section, we present the convergence and privacy analysis of the proposed UFed-GAN. We introduce the major proof steps and refer those interested to our corresponding references for more details.

A. Convergence of the discriminator

Let $\mathcal{G}$ and $\mathcal{D}$ be the generator and the discriminator of a GAN, respectively. Assume $\mathcal{G}$ is capable of capturing a distribution p_{GS} on the server and we are interested in learning a user distribution $p_{data}(x)$.

Proposition 1. *Any $\mathcal{D}$ initiated on a server and trained in accordance to Algorithm 1 with $\mathcal{G}$ and $p_{data}(x)$ converges to a unique $\mathcal{D}^*$ for the given $\mathcal{G}$ as presented in [5], i.e.,*

$$\mathcal{D}^* = \frac{p_{data(x)}}{p_{data(x)} + p_{GS}} \tag{2}$$

Since UFed-GAN merely splits the GAN training, the proof of Proposition 1 shall directly follow that in [5]. This proposition serves as a guarantee of the convergence of the server-side discriminator. It shows that the discriminator is still capable of capturing the user side's data distribution. This helps the generator on the server-side to generate user-like data as illustrated in the following proposition.

B. Convergence of the generator

Proposition 2. *Any $\mathcal{G}$ initiated on a server and trained in accordance to Algorithm 1 with $\mathcal{D}$ and $p_{data}(x)$ converges to a unique $\mathcal{G}^*$ which captures $p_{data}(x)$. i.e. $p_{GS} = p_{data}(x)$.*

Since UFed-GAN does not alter the training of $\mathcal{G}$, we can imitate and adopt the proof steps in [5]. Proposition 2 suggests that the server-side generator converges to the same generator that could have been trained locally. Therefore, on the server, we are able to regenerate synthetic samples which resemble the user data in terms of data distribution.

C. Divergence of any discriminator other than $\mathcal{G}$

Proposition 3. *Any generator G, other than $\mathcal{G}$ trained in accordance to the Algorithm 1 with $\mathcal{D}$ diverges from unique $\mathcal{G}^*$. i.e. $p_{GS} \neq p_{data}(x)$.*

To prove Proposition 3, we adopt a similar proof process as provided in [8]. Following Proposition 1 and Proposition 2, the pair of $\mathcal{G}$ and $\mathcal{D}$ is unique. Therefore, at each training step $\mathcal{G}^{'} = G^{'}$ must be asserted. Hence, any G difference from $\mathcal{G}$ at any step fails to capture $p_{data}(x)$.

TABLE I
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT FL APPROACHES OVER THREE DATASETS. PART OF THE RESULTS IN THIS TABLE ARE REPORTED FROM [13].

Method	CIFAR 10	SVHN	FashionMNIST
FedSimCLR	52.88	76.50	79.44
+ FedX	57.95	77.70	82.47
FedMoCo	57.82	70.99	83.58
+ FedX	59.43	73.92	84.65
FedBYOL	53.14	67.32	82.37
+ FedX	57.79	69.05	84.30
FedProtoCL	52.12	50.19	83.57
+ FedX	56.76	69.75	83.34
FedU	50.79	66.22	82.03
+ FedX	57.26	68.39	84.12
Full GAN	68.77	80.17	86.25
UFed-GAN	67.0	80.109	86.33

IV. RESULTS AND DISCUSSION

In this section, we present the experimental results on both utility and privacy.

A. Evaluation of the Utility

In a common semi-supervised setting as [13], let N be the number of users, β be the concentration parameter of Dirichlet distribution ($Dir_N(\beta)$), and s_{ij} be a sample taken from $Dir_N(\beta)$. We assign s_{ij} in proportion to the i-th class size of the user j. We pick $N = 10$ and $\beta = 0.5$ in accordance with [13]. All model comparisons are based on the linear evaluation protocol, which trains a linear classifier on top of representations or regenerated fake data [21].

We consider three well-known datasets: CIFAR10 [22], SVHN [23] and FashionMNIST [24]. Comparative results against five other FL algorithms are presented in Table IV-A. These algorithms are: FedSimCLR [25], FedMoco [19], FedBYOL [26], FedProtoCL [27] and FedU [28]. For each method, we present their accuracy on the respective dataset, together with their accuracy when further applying FedX [13]. We also compare the results with "full GAN" sharing.

From the results, UFed-GAN outperforms all other FL methods in the benchmark group, with an improvement of around 8% in CIFAR10, 3% in SVHN, and 2% in FashionMNIST. The accuracy gain arises from the power of GANs to understand the underlying data distribution of users and to generate synthetic data by preserving essential features. Another observation is that UFed-GAN is dataset-agnostic in terms of performance, delivering the best outputs in all tested datasets. This observation promotes the generalizability of the proposed method. In fact, UFed-GAN achieves similar performance as with full GAN sharing. However, full GAN sharing is prone to severe privacy leakage as shown in [8] and additionally requires heavy computation at each user node, which is in conflict with the FL objective of privacy preservation and conserving computation resources for users.

B. Evaluation of Privacy

We now evaluate the privacy leakage of the proposed UFed-GAN with respect to the attacker $\mathcal{A}$ as described in Section II-C. Suppose that $\mathcal{A}$ has access to each communication round.

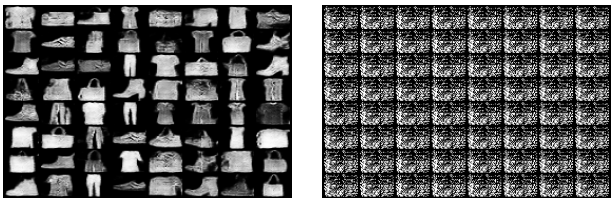

Fig. 3. Generated images from (a) cloud-server's model. (b) attacker's model.

TABLE II
FID SCORE, IS SCORE, AND SSIM OF THE $\mathcal{A}$ AND THE CLOUD SERVER AFTER 100 COMMUNICATION ROUNDS ON THE FASHIONMNIST DATASET.

Metric	Attaker	Cloud Server
FID	566.83	172.06
IS	1.01	3.15
SSIM	0.0067	0.8191

We initialize the generator of $\mathcal{A}$ as $G_{\mathcal{A}}$, with some random weights and eavesdrop on user uplink and access $\mathcal{M}(\theta)$. $\mathcal{A}$ trains $G_{\mathcal{A}}$ similarly as with the server-side training. The design of $G_{\mathcal{A}}$ is constrained by two parameters, the exact generator architecture and the initial weights W_G of the generator at the server. Therefore, any attacker selecting the accurate generator architecture and initial weights is practically unachievable. However, in our experiments, we assume $\mathcal{A}$ knows the exact architecture of G, but with different random initial weights $W_{\mathcal{A}} \neq W_G$. We compare the generated images of the G and $G_{\mathcal{A}}$ trained on the FashionMNIST dataset in Fig. 3. It can be clearly seen that the generator G on the cloud server captures the user's underlying data distribution, whereas $G_{\mathcal{A}}$ converges to a trivial point. Moreover, almost no useful visualization information is gained by the attacker as shown in Fig. 3. The main reason is the uniqueness of the generator and the discriminator pair as presented in Proposition 1.

To examine privacy leakage quantitatively, we use the Frechet Inception Distance (FID) score [29], IS, and structural similarity index measure (SSIM) [30]. As discussed in the paper [31], the similarity between the generated images and real data quantifies the privacy leakage. In table IV-B, we record average FID, IS, and SSIM scores for the data generated by $\mathcal{A}$ and the cloud server. Lower FID values, higher SSIM, and larger IS values represent better reconstruction quality. The FID score for $\mathcal{A}$ is higher than the cloud server by a great margin. This shows that, compared to the cloud server, $\mathcal{A}$ generated data carries less information about the training data. We further corroborate this observation by comparing the SSIM and IS values as well. The experimental results demonstrate the privacy preservation of UFed-GAN against full-GAN sharing.

V. CONCLUSION

In this work, we develop a novel framework of UFed-GAN to address the challenges imposed by the lack of labeled data and by limited local computation resources in federated learning. Moreover, we propose a separate training strategy and a sharing scheme based on DCGANs. We provide an analysis of the convergence and privacy leakage of the proposed

framework. Our empirical results demonstrate the superior performance of UFed-GAN in comparison against benchmark FL methods. We plan to investigate the communication overhead reduction for GAN-based FL and the application of semantic learning in distributed learning in future works.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[2] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[3] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[4] X. Yuan and P. Li, "On convergence of fedprox: Local dissimilarity invariant bounds, non-smoothness and beyond," in *Advances in Neural Information Processing Systems*, New Orleans, LA, USA, Jul. 2022, pp. 10 752–10 765.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, Montreal, Canada, Dec. 2014.

[6] H. Zhang, Z. Zhang, A. Odena, and H. Lee, "Consistency regularization for generative adversarial networks," in *International Conference on Learning Representations*, Addis Ababa, Ethiopia, Apr. 2020.

[7] X. Cao, G. Sun, H. Yu, and M. Guizani, "Perfed-gan: personalized federated learning via generative adversarial networks," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3749–3762, Mar. 2023.

[8] A. Wijesinghe, S. Zhang, and Z. Ding, "Ps-fedgan: an efficient federated learning framework based on partially shared generative adversarial networks for data privacy," *arXiv preprint arXiv:2305.11437*, 2023.

[9] J. Mu, D. Xie, H. Huang, and X. Jing, "Computation-constrained spectrum sensing in iot-based scenarios," *IET Communications*, vol. 14, no. 20, pp. 3631–3638, Nov. 2020.

[10] J. Bekker and J. Davis, "Learning from positive and unlabeled data: a survey," *Machine Learning*, vol. 109, pp. 719–760, Apr. 2020.

[11] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the win: One-shot federated clustering," in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 2611–2620.

[12] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Advances in Neural Information Processing Systems*, Dec. 2020, pp. 19 586–19 597.

[13] S. Han, S. Park, F. Wu, S. Kim, C. Wu, X. Xie, and M. Cha, "Fedx: Unsupervised federated learning with cross knowledge distillation," in *European Conference on Computer Vision*, Tel Aviv, Israel, Nov. 2022, pp. 691–707.

[14] M. Servetnyk, C. C. Fung, and Z. Han, "Unsupervised federated learning for unbalanced data," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[15] N. Lu, Z. Wang, X. Li, G. Niu, Q. Dou, and M. Sugiyama, "Federated learning from only unlabeled data with class-conditional-sharing clients," in *International Conference on Learning Representations*, Apr. 2022.

[16] E. S. Lubana, C. I. Tang, F. Kawsar, R. P. Dick, and A. Mathur, "Orchestra: unsupervised federated learning via globally consistent clustering," *arXiv preprint arXiv:2205.11506*, 2022.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, Barcelona, Spain, Dec. 2016.

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

[20] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: evaluating privacy leakage of generative models using generative adversarial networks," *arXiv preprint arXiv:1705.07663*, pp. 506–519, 2017.

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

[22] A. Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[23] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, Granada, Spain, Dec. 2011.

[24] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, Vienna, Austria, Jul. 2020, pp. 1597–1607.

[26] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, Dec. 2020.

[27] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[28] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp. 4912–4921.

[29] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, Dec. 2017.

[30] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[31] D. Dangwal, V. T. Lee, H. J. Kim, T. Shen, M. Cowan, R. Shah, C. Trippel, B. Reagen, T. Sherwood, V. Balntas *et al.*, "Analysis and mitigations of reverse engineering attacks on local feature descriptors," *arXiv preprint arXiv:2105.03812*, 2021.