BI-GAN: Batch Inversion Membership Inference Attack on Federated Learning

HIEP VO, University of Technology Sydney, Australia
MINGJIAN TANG, University of Technology Sydney, Australia
XI (JAMES) ZHENG, Macquarie University, Australia
SHUI YU, University of Technology Sydney, Australia

Federated Learning is a growing advanced collaborative machine learning framework that aims to preserve user-privacy data. However, multiple researchers have investigated attack methods from the server side via gradient inversion techniques and Generative Adversarial Networks (GAN) to reconstruct the raw data distributions from users. So far, the past researched attacks are limited to certain assumptions. For example, the attacker already has a small subset of true data, each client is limited to data of distinct labels from one another, and no local batch training. Furthermore, many GAN-based attacks can only achieve image class reconstruction instead of victim identification while hindering the global model performance. In this paper, we propose Batch Inversion GAN (BI-GAN), a novel membership inference attack that can recover user-level batch images from local updates, utilizing both gradient inversion techniques and GAN. Our attack is more stealthy since it only requires access to gradients and does not interfere with the global model performance and is more robust in terms of image batch recovery and victim classification. The experiments show that our attack recovers higher quality images of the victim with higher accuracy compared to other attacks.

Additional Key Words and Phrases: Membership Inference Attack, Gradient Inversion, GAN, Federated Learning

ACM Reference Format:
Hiep Vo, Mingjian Tang, Xi (James) Zheng, and Shui Yu. 2022. BI-GAN: Batch Inversion Membership Inference Attack on Federated Learning. *MobiArch* , 6 pages

1 Introduction

Federated Learning (FL) [5] is a novel distributed deep learning framework that allows a deep learning model to be collaboratively trained over a series of users. Each participant, as a data provider can locally train their model and submit the model updates to the global server instead of sending his/her raw private data. Recent researches have identified that federated learning can be subject to inference attacks that aim to learn the real training data attributes and to predict if a data sample is part of the training set [13].

Authors' addresses: Hiep Vo, University of Technology Sydney, 15 Broadway Ultimo, Sydney, New South Wales, Australia, hiep.k.vo@student.uts.edu.au; Mingjian Tang, University of Technology Sydney, 15 Broadway Ultimo, Sydney, New South Wales, Australia, mingjian.tang@student.uts.edu.au; Xi (James) Zheng, Macquarie University, Sydney, New South Wales, Australia, james.zheng@mq.edu.au; Shui Yu, University of Technology Sydney, 15 Broadway Ultimo, Sydney, New South Wales, Australia, shui.yu@uts.edu.au.

This work is partially supported by Australia ARC LP190100676.

XXXX-XXXX/2022/7-ART
https://doi.org/

The current recent attacks on FL system can be classified as Inversion attacks and GAN-based attacks. Inversion attacks can aim to steal the model's functionality (model inversion) [1] or reconstruct the exact private training data from the gradients update [16], [15]. Instead of reconstructing exact true training data, GAN-based attacks such as the ones proposed in [3] and [13] aim to generate fake data samples that best represent the training data distribution. Both Inversion attacks and GAN-based attacks can exploit the FL system's security weaknesses to duplicate the global model's function abilities and recover/replicate clients' private data.

Many of the current attacks targeting FL focus on a malicious client instead of a malicious server [1], [3], and [13]. These attacks have a weakness in requiring the attacker to have a subset of true training data with an unlimited amount of target model queries to perform the attack. Furthermore, attacks from a client are not as efficient as from a malicious server since the server can access the model's parameters for more efficient targeted attacks. Many of the GAN-based attacks on FL [3],[13] so far do not achieve targeted membership inference attack since they can only make fake replications of the whole training data instead of the victim's data.

To target the above weaknesses, our attack first utilizes a Gradient Inversion technique for a malicious server to constantly reconstruct client-wise private data representatives in batch without having to obtain a subset of true training data or unlimited model queries to perform the attack. Furthermore, our attack also implements a GAN-based attack model to generate fake client-wise private data. Our GAN attack model has an advantage over previous GAN-based attacks in FL because it can generate fake data of specific targeted victim instead of the overall private training data.

Our major contributions are listed as follows:

- We combine the benefits of Gradient Inversion techniques and GAN models to introduce a server side membership inference attack in Federated Learning. The attack can achieve a better client-wise targeted inference attack, eliminating the requirements of unlimited data queries with a subset of true data.
- The attack framework BI-GAN is introduced which combines the benefits of batch image gradient inversion and custom Auxiliary Classifier GAN to reconstruct and generate victim specific private data.
- Extensive experiments are conducted under both low and high quality image settings to compare BI-GAN's performance against other state of the art attack frameworks.

The rest of the article is organised as follows. Part II presents related work of this project. Part III reviews background of involved techniques. Part IV provides the threat model of out work. Part

V discusses the BI-GAN attack. Part VI presents the experimental results. Part VII concludes our work and projects future works.

2 Related Work

2.1 Membership Inference Attack in Federated Learning

2.1.1 *Inversion Attack*

Gradient Inversion is an approach for the server to extract user-level data from the submitted gradients from the clients.

In late 2019, early 2020, Ligeng Zhu and Bo Zhao have proposed algorithms that can construct private data from gradients leakage [16], [15]. These papers aim to revert the representations of single images from given leaked gradients by introducing optimization algorithms that match inputs and labels to their targeted gradients as well as enhancing the label restoration step. These researches, however, do not apply well to federated learning attacks since they do not support image restoration in batch which is usually the case for federated user data.

The most recent gradient inversion framework that support batch training in FL is proposed by Hongxu Yin in [11]. This research proposes an optimization algorithm that converts noise to adversarial images while controlling matching gradients under a group registration framework that aims to reconstruct images from the average of gradients. However, this approach has not been implemented to multiple local updates in federated learning.

2.1.2 *GAN-based attacks*

There are a couple of researches that implement GAN to reconstruct the distributions of true training data.

In [3], the authors assume that the adversary is a participant trying to learn a secret label of an image. The framework in this paper is quite similar to [13] with the difference that the adversary only tries to flip the label of an unknown generated image and learn how the global gradients shift to determine the actual class of that sample. This approach still needs white-box access to the model structure for gradients update which is not always available for participants in federated learning.

In [14], the authors use GAN to generate fake samples that have similar distribution as original dataset. The fake samples then get labeled by querying the target model to generated supervised attack training set for training attack model. Although this approach can get user-level privacy attack, it still relies on large amount of query of target model to get labels for fake samples which is usually very limited for a participant in federated learning.

2.2 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) have been proposed by Goodfellow in 2014 [2], and can be employed to generate images that similar to those in the real dataset or generate brand new one by itself. This training model consists of Generator G and Discriminator D. Generator creates fake data based on the random noise, and the results will be evaluated and identified by Discriminator that trained by the real dataset. The image been judged as fake will be trained again until it is similar to the real one and been identified as real by Discriminator. Training process of GAN model can be expressed as Eq. 1.

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \\ & \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G\left(z\right)\right)\right)\right],\end{aligned} \tag{1}$$

where p_{data} and $p_z(z)$ represent distribution of original images and random vector z respectively. This model will keep training until the Nash equilibrium has been achieved on the adversarial game.

3 Threat Model

3.1 Learning Scenario

Following the federated learning training protocols, we assume N (N ≥ 2) clients that train local models collaboratively for a common learning goal. The data of each client does not contain personal information and are nonIID distributed. For simplicity, we consider image classification as the common learning goal.

3.2 Attacker's Knowledge and Goals

We consider a malicious server instead of a malicious client as the attacker with the goal to reconstruct specific clients' private data. The attacker would have white-box access to the federated learning model structure as well as the gradient updates from each client. There are three main goals of the attacking server: (1) *Samples Reconstruction*: reconstruct the client-wise data representatives via batch-optimized gradient inversion, (2) *Samples Generation*: Train BI-GAN model to generate images similar to specific clients' data, and (3) *Prediction accuracy*: The Discriminator should has high accuracy in predicting the class label and the victim that the real sample image belongs to.

4 BI-GAN Attack

4.1 Overview of BI-GAN

For BI-GAN attack, we incorporated a batch gradient inversion technique to recover representatives of client data for training a custom GAN attack. Our GAN model is inspired by the capability of encoding more conditions into traditional GAN such as CGAN [7] and ACGAN [9] and improve it to further discriminate the client identifications.

Fig. 1 illustrates the high level view of the proposed BI-GAN attack. Assume the client population is N. Here the attack framework aims to discriminate all clients' data from one another instead of targeting a single client which makes the attack more robust and consumes less training time when targeting different clients. In a normal federated learning round, the malicious server distributes the global model M to N clients and receive the respected updates $u_1, u_2, ..., u_n$ after the clients have finished their local training where the clients are allowed to train the data in multiple batches of different labels. In order to reconstruct the private features from the gradients updates, a batch gradient inversion algorithm inspired by [11] is implemented to reconstruct representatives of true client data which will be used illustrated in Section 4.3. The representatives would then get labeled by querying the global model to construct the training data set for GAN. Then to generate new fake data that captures the clients' real data distributions while predicting the various memberships of a data sample, we propose a custom variant of ACGAN with extra embedded encoding of victim identification with a novel Discriminator that discriminate all clients' ids, realism, and categories from a single attack. Contrary to other attack frameworks that set the structure of the Discriminator similar to that of the global model, we implement custom Discriminator structure

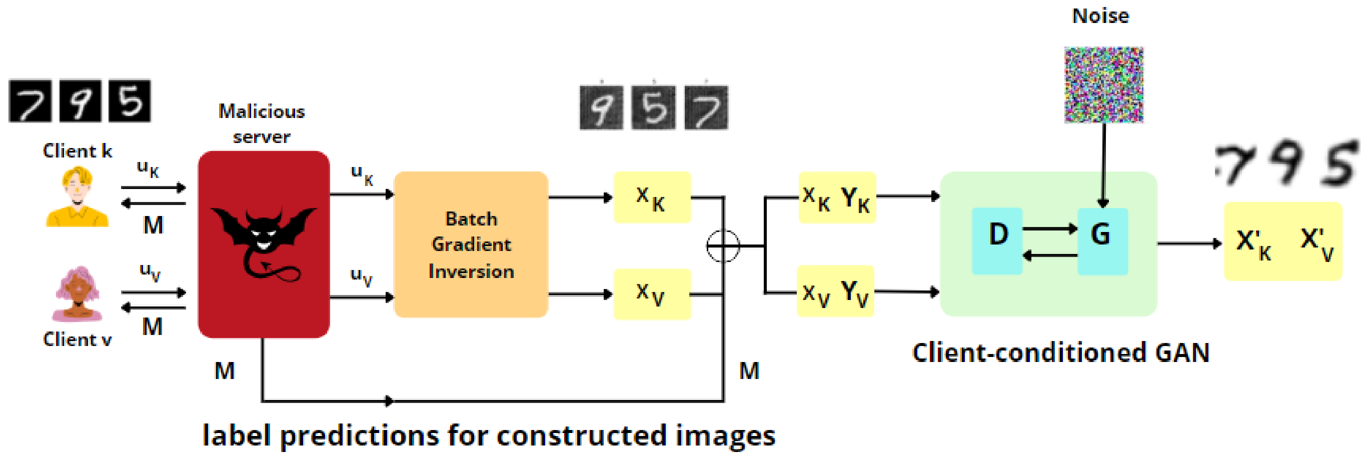


Fig. 1. Overview of proposed BI-GAN membership inference attack framework from malicious server in federated learning at a single iteration t. There are N clients and the server can attack/discriminate all clients from a single attack. The client k's update is denoted as u_k, the global shared model is denoted as M. After the batch gradient inversion step, the recovered features of clients k and v are denoted as x_k and x_v respectively with y_k and y_v are the predicted batch labels are used to query the global model M. The victim-conditioned GAN model is then trained with the reconstructed representatives $(x_k, y_k), (x_v, y_v)$ from the clients. The Discriminator D aims to have similar performance as the global model as well as discriminating client-wise data while Generator G aims to generate samples x'_k, x'_v that represents the clients' data

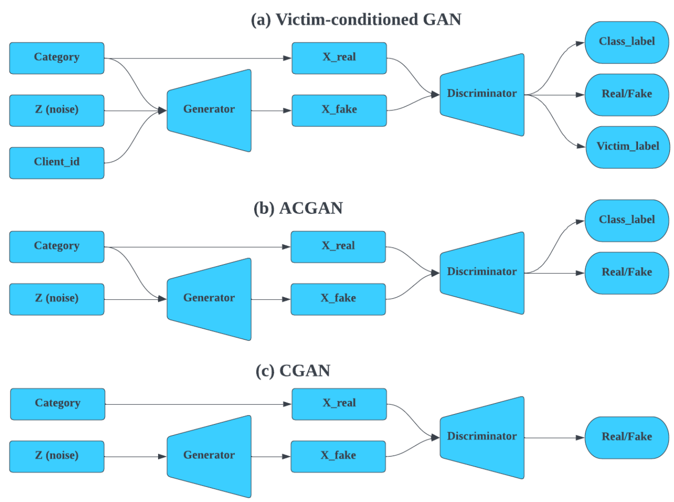


Fig. 2. Construction comparison between classic GAN models and our victim-conditioned GAN. Here CGAN has been used for membership inference attack in Federated Learning [14]

with fewer convolutional layers to further decrease training time. The Discriminator then would server as a shadow model which would reach similar performance as the global model while being able to discriminate data ownership and realism after training. in Section 4.2, we will detail the structure of our victim-conditioned GAN model.

4.2 Victim-Conditioned GAN

To be more specific, the roles of the discriminator of BI-GAN includes (1) discriminating real/fake image as a standard GAN, (2) correctly categorising the real label of the input, and (3) identifying the victim ownership of the input image by categorizing victim ids. The difference between our Victim-conditioned GAN to other GAN structures is shown in Fig. 2. In our model, Victim-Conditioned GAN trains the Discriminator as a shadow model to the global model with similar image category prediction and extra victim id prediction in the output layer. Compare to the membership inference attack in [14], our model encodes the victim identification to the Generator and does not require real sample data during training. The structure for each of the Dense layer from the Discriminator is shown below:

$$D_{real} = Sigmoid(FC_{real}(L_s))$$
$$D_{cat} = Softmax(FC_{cat}(L_s))$$
$$D_{id} = Softmax(FC_{id}(L_s))$$

Here, D_{real}, D_{cat}, and D_{id} are dense layers for predicting realism, image category, and client id respectively. We implement Softmax function for both categories and ids because victim-conditioned GAN model will have all client ids embedded instead that of a single victim. FC denotes the fully connected layers and L_s represents the layers from the shadow Discriminator model.

The Generator G takes three inputs: noise z sampled form Gaussian distribution, randomized category and client id. There will be three loss functions for detecting real/fake (l_r) , image category (l_c) and victim id (l_v) which are Binary Crossentropy, Sparse Categorial Crossentropy, and Sparse Categorial Crossentropy respectively. After sufficient training, the Discriminator D will try to minimize $l_r + l_v + l_c$ while Generator G will try to minimize $l_v - l_r + l_c$.

4.3 Batch Gradient Inversion

To obtain the training samples for victim-conditioned GAN model, the malicious server needs to reconstruct the clients' data base on their gradients updates. Hongxu Yin in [11] proposed a gradient inversion approach that can recover images of different labels in a batch up to a size of 48 images from gradients updates from noise. However, this method has not been implemented to perform a membership inference attack in Federated Learning. Motivated by [11], we are able to create samples that represent the victims' data when the clients are allowed to have multiple training labels in batch.

This approach introduces an image fidelity regulation and a group consistency regulation to the traditional gradient inversion optimization function mentioned in [16] as shown below

$$x'_k = \alpha_G \sum_l \arg\min_{x_k^l} ||\nabla u_k^l - \nabla u_{x_k}^l||_2 + R_{fidelity}(x_k) + R_{group}(x_k) \quad (2)$$

Here, the summation of ground truth gradients from all layers l is scaled on a fixed parameter α_G and the difference between the gradients of reconstructed images and the real images is minimized under l_2 distance. The fidelity regulation loss function is the combination of strong prior R_{BN} proposed in DeepInversion [12] and two classic image prior R_{TV} [6] and R_{l_2} [8]. R_{BN} penalizes x'_k according to the variation estimate and batch-wise mean of convolution layers' feature maps while R_{l_2} and R_{TV} penalizes x'_k according to total l_2 norm and total variance respectively as shown in Eq. 3 where α denotes different scaling factors.

$$R_{fidelity}(x_k) = \alpha_{l_2} R_{l_2} + \alpha_{TV} R_{TV} + \alpha_{BN} R_{BN} \quad (3)$$

To help enhances the quality of recovered images in batch, we incorporated the group consistency regulation $R_{group}(x_k)$ to our constructed images x_k as shown in Eq. 4. This regulation would first initiate multiple repeated optimizations with different seeds then optimizes the multiple seeds in parallel with a combined optimization goal.

$$R_{group}(x_k, x_{kg} \in G) = \alpha_{group} ||x_k - \hat{E}(x_{kg} \in G)||_2 \quad (4)$$

Here, $\hat{E}(x_{kg \in G})$ is considered to be the pixel-wise average of the image group. Optimizing multiple seeds under a joint optimization constraint is promised to output more robust and more realistic images.

To minimize the training cost, we omit the batch label restoration step in [11] and replaced it with labels predicted by querying the global model. Furthermore, to further ensure the quality of training data for victim-conditioned GAN, we implemented an extra noise filter by filtering out reconstructed images with high noise variance [4] as shown in Eq. 5. This method is only applicable to two dimensional images

$$\sigma_n^2 = \frac{1}{36(W-2)(H-2)} \sum_{x_k} (x_k(u,v) * N)^2 \quad (5)$$

where $N = 2(L_2 - L_1)$ is the mask operation over 2 Lapcacian masks L_1, L_2 of an image. In this case, we denote $N = \begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}$ and an image is considered noise if the noise variance σ_n^2 is greater than 0.5

Algorithm 1 BI-GAN attack

Input: Global model M and a set of client updates $(u_1, u_2, ..., u_n)$
Output: Discriminator D and Generator G
Initialize M, D, and G
trainImages = []
trainLabels = []
trainVictims = []
for k in range N **do**
  Construct client data images x_k from u_k via Eq. 2
  Filter noisy images from x_k via Eq. 5
  Add x_k to trainImages
  Get predicted labels y_k by querying x_k by M
  Add y_k to trainLabels
  Add k to trainVictims
end for
Get fake images X_{fi}, fake labels Y_{fc}, fake victim ids Y_{fv} from G with inputs: $latentdimension$, $trainLabels$, and $trainVictims$
Update D by minimizing $l_r + l_v + l_c$ with inputs: $trainImages$ and X_{fi}
Update G by minimizing $l_v - l_r + l_c$.

4.4 Attack Algorithm

This section details the BI-GAN attack algorithm in a specific federated training iteration which is depicted in algorithm 1. The model would first take the global model M and the list of client gradient updates $(u_1, u_2, ..., u_n)$ as inputs from $N > 2$ clients. Then, each of the client data representatives would be reconstructed via batch gradient inversion approach from Section 4.3, resulting in a set of x_k images for each victim. The noisy images from x_k is then filtered out by Eq. 5. Then, the high quality images from x_k will be labelled by global model M and the training data set for victim-conditioned GAN would include the reconstructed images, predicted labels, and the client ids.

After obtaining the training dataset, the predicted labels and victim ids are fed to the Generator G to generate fake images with fake ids and fake labels. The Discriminator D takes reconstructed images $trainImages$ as input and try to discriminate the reconstructed images from the fake images generated by G. Both G and D will be trained simultaneously while optimizing their respecting loss functions mentioned in section 4.2. The output of BI-GAN algorithm is not only the Generator G for creating fake client-like images but also the Discriminator D which has high accuracy in category prediction and is also able to predict the ownership of a data sample.

4.5 Dataset and Experiment Setup

4.5.1 Dataset

MNIST is a popular benchmark dataset for deep learning models. This dataset contains 70000 grayscale handwritten digits images with 10 labels from 0 to 9. The size of each image is 28x28 pixels and is divided into 60000 training records and 10000 testing records

4.5.2 Experiment Setup

As mentioned in Section 4.1, our attack model consists of a Batch Gradient Inversion model, a Discriminator, and a Generator, targeting a global Federated Learning model M. For both the attack model and the federated global model, we use Convolutional Neural

Table 1. Network Structure for MNIST

Federated Model	$28^2{\times}1 \xrightarrow{\text{Conv(stride=2, kernel=3)}} 32^2{\times}\,64 \xrightarrow{\text{Conv(stride=2, kernel=3)}} 64^2{\times}\,128 \xrightarrow{\text{Conv(stride=2, kernel=3)}} 128^2{\times}\,256 \xrightarrow{\text{FC}} 12544 \xrightarrow{\text{FC, Softmax()}} 10$
Discriminator	$28^2{\times}1 \xrightarrow{\text{Conv(stride=2, kernel=3)}} 14^2{\times}\,128 \xrightarrow{\text{Conv(stride=2, kernel=3)}} 7^2{\times}\,128 \xrightarrow{\text{FC}} 588 \xrightarrow{\text{FC, Softmax())}} 10$
Generator	$(100,1,1) \xrightarrow{\text{Embedding}} 100 \xrightarrow{\text{FC}} 7^2{\times}\,384 \xrightarrow{\text{Deconv}} 14^2{\times}\,192 \xrightarrow{\text{Deconv, tanh}} 28^2{\times}\,1$

Table 2. Gradient Inversion techniques

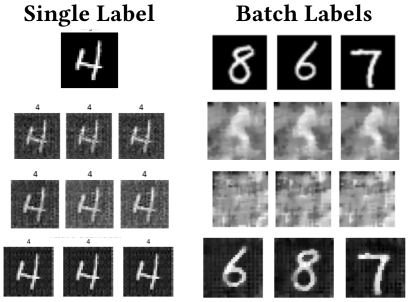
Single Label **Batch Labels**

Target

iDLG [15]

CPL [10]

BI-GAN



Network (CNN) based architecture with categorical output layers. Table 1 shows the network architectures for MNIST dataset.

The global Federated Learning model will have 4 convolution layers with a single Dense layer to predict image class. The size of (3x3) kernels are consistent for every convolution layers with strides 2 except for the last layer (no strides). There are also a Batch Normalization layer, a Leaky Rectified Linear Unit layer with negative slope 0.2, and a Dropout layer with parameter 0.5 after each of the last 3 convolution layers. The structure for the Discriminator is simpler with only 2 convolution layers with Leaky Rectified Linear Unit layer with negative slope 0.2 after both layers, and 3 Dense layers to predict realism, image category and client id respectively. For the generator, the kernel size is 5×5 with stride 2, taking random noise, number of categories, and number of clients as input. The generator G squeezes the input to size 100 and concanate the image category and client id embeddings. The output of G is generated image of size 28x28 pixels.

For setting up training dataset, we set the number of clients $N = 10$. Each of the client will have 200 samples randomly drawn from all available classes as personal private data.

4.6 Qualitative and Quantitative analysis

4.6.1 Batch Gradient Inversion Comparison

In this section, we compare the performance of our Batch Gradient Inversion algorithm to other existing gradient inversion algorithms. Table 2 shows the visual reconstructions under single label and batch labels scenarios. Here, all three methods are using l_2 distance metrics. iDLG and our method BI-GAN use label prediction instead of using a label-matching regulation while CPL has a label-matching regulation. Furthermore, the proposed BI-GAN includes reality and group consistency regulations.

Table 3. GAN techniques

CGAN

ACGAN

BI-GAN

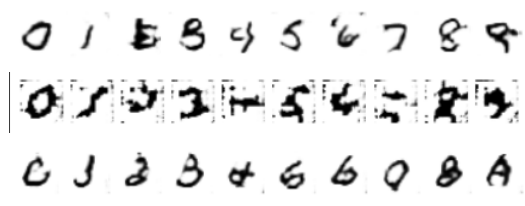


Table 4. GAN label prediction accuracy (%)

	Category	**Victim**
CGAN	~	~
ACGAN	91.1	~
BI-GAN	**92.4**	**62.9**

Table 5. BI-GAN accuracy per client (%)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Category	91	90	92	96	97	92	90	93	87	96
Victim	**70**	61	**85**	**74**	42	62	**72**	69	47	47

The results in Table 2 shows that BI-GAN manages to recover much higher quality image in both single label image setting and batch image setting (sample batch size 3) with more defined color details. Both iDLG and CPL fails to reconstruct images from the gradients of a batch training while BI-GAN manages to recover the features of individual images. This result proves the effectiveness of BI-GAN in recovering batch of images from gradients updates.

4.6.2 GAN frameworks comparison

This section compares the performance of GAN structures mentioned in Fig. 2 which includes CGAN from a membership inference attack [14], the traditional ACGAN structure and our Victim-conditioned GAN model.

A) Visual Comparison Table 3 shows the sample fake image constructions of 10 classes from MNIST data set. Since BI-GAN needs client identification as input, we show the results of BI-GAN generated images targeting one random victim while letting CGAN and ACGAN generate images that represent the whole training dataset. All three models are trained with the reconstructed images from Batch Gradient Inversion model for 5000 iterations. According to Table 3, CGAN appears to have good visual output of the generated images. That's because CGAN only has to focus on optimizing the reality of the image. ACGAN, on the other hand, has the most noisy image outputs. That's because predicting class label function create more noise during training. Our proposed BI-GAN model generates clear images similar to CGAN while also targeting a single client's training data. Our model is shown to perform better than traditional ACGAN model since the optimization functions are optimized to also highlight the ownership of the images.

The quantitative accuracy performance of BI-GAN is presented in tables 4 and 5. Table 4 evaluates how different GAN structures' Discriminators can be used to predict the Category label and the Victim owner of input image. Here, only BI-GAN is supported to predict both objectives while ACGAN can only predict item label and CGAN is incapable since CGAN is only supported to differentiate between real and fake categories. BI-GAN is proved to only be more accurate in predicting image label with 92.4% accuracy compared to 91.1% for ACGAN, but also able to predict client identification for client-wise membership inference attack with 62.9% accuracy. The accuracy of victim id is not as high as category accuracy for BI-GAN is because we allow each user to randomise there data from all given labels so the uniqueness among clients is not up to par with the uniqueness between image labels. This result also shows that adding embedded client information into GAN does not hinder its performance in predicting image category and the model can be served as a shadow model representing the global Federated Learning model. Furthermore, BI-GAN is proved to be a more robust attack because the malicious server only has to train the attack model once to attack all clients' data instead of retraining the model to target different clients at a time. Table 4 lists the accuracy of BI-GAN attacks performed on individual clients. This table shows that the model can get high client prediction accuracy (≥ 70%) for multiple clients after a single attack.

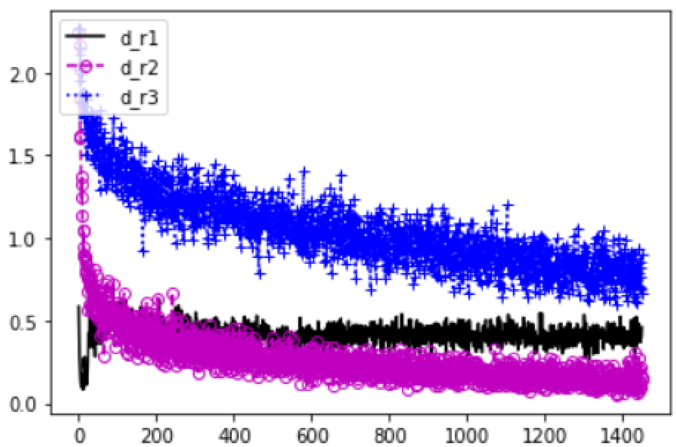


Fig. 3. BI-GAN's Discriminator loss progression on images recovered from Batch Gradient Inversion over 14500 iterations, recording the average loss of every 10 iterations. d_{r1}, d_{r2}, and d_{r3} are losses for realism, image labels, and victim labels respectively

4.6.3 BI-GAN loss functions

Fig. 3 illustrates the loss improvements of BI-GAN's Discriminator D. According to Section 4.2, D has three loss functions D_{real}, D_{cat}, and D_{id} for differentiating real and fake images, classifying image category and predicting client ownership. These three loss functions are represented as d_{r1}, d_{r2}, and d_{r3} in Fig. 3. Here, the model seems to converge from iteration 2000 with low consistent loss value (~ 0.5) for predicting realism and image category. In terms of optimizing the model for classifying client identification, d_{r3} loss gradually decreases over time reaching (~ 0.7) categorical entropy loss after training. This result does so that the BI-GAN model's Discriminator can further optimize the predictions clients for membership inference attack in Federated Learning.

5 Future Work

In this paper, we confirmed the feasibility of our BI-GAN membership inference attack model targeting client-wise private data in Federated Learning system. The attack framework implement an advanced batch gradient inversion algorithm that proves to recover higher image quality allowing clients to train local model in batches. Furthermore, BI-GAN includes a novel Victim-conditioned GAN model that is not only able to generate fake images comparable to existing GAN attack frameworks but is also able to predict the owner of an input image. In the future, we will research and improve the GAN attack framework together with other approaches.

References

[1] Xueluan Gong, Yanjiao Chen, Wenbin Yang, Guanghao Mei, and Qian Wang. 2021. InverseNet: Augmenting Model Extraction Attacks with Training Data Inversion. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2439–2447. Main Track.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[3] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 603–618. https://doi.org/10.1145/3133956.3134012

[4] John Immerkær. 1996. Fast Noise Variance Estimation. *Computer Vision and Image Understanding* 64, 2 (1996), 300–302. https://doi.org/10.1006/cviu.1996.0060

[5] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*. https://arxiv.org/abs/1610.05492

[6] Aravindh Mahendran and Andrea Vedaldi. 2014. Understanding Deep Image Representations by Inverting Them. *CoRR* abs/1412.0035 (2014). arXiv:1412.0035 http://arxiv.org/abs/1412.0035

[7] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

[8] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2014. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. https://doi.org/10.48550/ARXIV.1412.1897

[9] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2016. Conditional Image Synthesis With Auxiliary Classifier GANs. https://doi.org/10.48550/ARXIV.1610.09585

[10] Wenqi Wei, Ling Liu, Margaret Loper, Ka Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A Framework for Evaluating Gradient Leakage Attacks in Federated Learning. *CoRR* abs/2004.10397 (2020). arXiv:2004.10397 https://arxiv.org/abs/2004.10397

[11] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. 2021. See through Gradients: Image Batch Recovery via GradInversion. *CoRR* abs/2104.07586 (2021).

[12] Hongxu Yin, Pavlo Molchanov, Zhizhong Li, Jose M. Alvarez, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. 2019. Dreaming to Distill: Data-free Knowledge Transfer via DeepInversion. *CoRR* abs/1912.08795 (2019). arXiv:1912.08795 http://arxiv.org/abs/1912.08795

[13] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. 2019. Poisoning Attack in Federated Learning using Generative Adversarial Nets. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. 374–380. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00057

[14] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. 2020. GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC40277.2020.9148790

[15] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. iDLG: Improved Deep Leakage from Gradients. *CoRR* abs/2001.02610 (2020).

[16] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. *CoRR* abs/1906.08935 (2019).