Backdoor Attack is A Devil in Federated GAN-based Medical Image Synthesis

Ruinan Jin¹ and Xiaoxiao Li¹

¹ The University of British Columbia
ruinanjin@alumni.ubc.ca, xiaoxiao.li@ece.ubc.ca

Abstract. Deep Learning-based image synthesis techniques have been applied in healthcare research for generating medical images to support open research. Training generative adversarial neural networks (GAN) usually requires large amounts of training data. Federated learning (FL) provides a way of training a central model using distributed data from different medical institutions while keeping raw data locally. However, FL is vulnerable to backdoor attack, an adversarial by poisoning training data, given the central server cannot access the original data directly. Most backdoor attack strategies focus on classification models and centralized domains. In this study, we propose a way of attacking federated GAN (FedGAN) by treating the discriminator with a commonly used data poisoning strategy in backdoor attack classification models. We demonstrate that adding a small trigger with size less than 0.5% of the original image size can corrupt the FL-GAN model. Based on the proposed attack, we provide two effective defense strategies: global malicious detection and local training regularization. We show that combining the two defense strategies yields a robust medical image generation.

Keywords: GAN · Federated learning · Backdoor attack

1 Introduction

While deep learning (DL) has significantly impacted healthcare research, its impact has been undeniably slower and more limited in healthcare than in other application domains. A significant reason for this is the scarcity of patient data available to the broader machine learning research community, largely owing to patient privacy concerns. Furthermore, even if a researcher is able to obtain such data, ensuring proper data usage and protection is a lengthy process governed by stringent legal requirements. Therefore, synthetic datasets of high quality and realism can be used to accelerate methodological advancements in medicine [4, 7].

Like most DL-based tasks, limited data resources is always a challenge for the generative adversarial network (GAN)-based medical synthesis, and data

Code is available at https://github.com/Nanboy-Ronan/Backdoor-FedGAN
collaboration between different medical institutions makes effects to build a robust model. But this operation will cause data privacy problems which could be a risk of exposing patient information. Federated learning (FL) [15], a privacy-preserving tool, which keeps data on each client locally and exchanges model weights by the server during learning a global model collaboratively. Due to its property of privacy, it is a popular research option in healthcare [24].

However, FL is vulnerable to malicious participants and there are already studies deep dive into different kinds of attacks for classification models in federated scenarios, like gradient inversion attacks and backdoor attacks [2, 13]. In a backdoor attack for classification, the attacker adds a trigger signal, such as a small patch with random noise, to its training data and changes the correct label to a wrong one [25]. In FL training, malicious clients can poison training data using a backdoor attack and mislead the global to make incorrect predictions. It is possible for medical imaging backdoor triggers to be induced by (un)intentional artifacts occurring during the sensor acquisition and preparation processes. Recent work [2] observed that backdoor attack takes advantage of the classification model’s tendency to overfit the trigger rather than the actual image. This notion inspires us to think about how we can integrate it into generative models in FL.

Exiting backdoor attacks are specifically designed for the classification task or model training in centralized domain. In this work, we focus on backdoor attack on federated GAN (FedGAN) via data poisoning, which under-explored in existing literature. The success of this attack is subsequently determined to be the result of some local discriminators overfitting on the poisoned data and corrupting the local GAN equilibrium, which then further contaminates other clients when averaging the generator’s parameters during federated training and yields high generator loss. Based on the attack, we suggest two potential ways of defending it from global- and local-level of FL: detecting the client’s adversarial behavior on the server-side and blocking it from dispersing to further, and applying a robust training procedure locally for each client. In our experiment, we apply our adversarial and defense mechanisms to a widely used skin cancer dataset. We show that the adversarial strategy is able to corrupt FedGAN only by adding a trigger with 0.39% size of the original image in the malicious training set.

2 Methods

2.1 Federated Generative Adversarial Network

Fig. 1 depicts the framework of the FedGAN in our study. As discriminators in GAN have direct access to clients’ private data, exposing the risk of data leakage by inverting their gradients in FL training [13], our FedGAN framework only exchanges generator’s parameters with the server while keeping the whole discriminator locally. To this end, our FedGAN locally trains both discriminator and generator pairs and globally shares generators’ parameters, which is modified from [22].

Formally, we assume that a trusted central generator $G_{\text{server}}$ synthesizes images from a set of $N$ federated clients. Each client $C_i$, for $i \in [N]$ consists a
locally trained discriminator $D_i$, and a generator $G_i$. $G_i$ takes random Gaussian noise $z$ as input to generate synthetic images, and $D_i$ distinguish the synthetic image $\hat{x} = G(z)$ v.s. private image $x$. We adopt FedAvg [18] to aggregate $G_i$ to $G_{\text{server}}$, while keeping $D_i$ locally. At the end, our federated GAN generate synthetic medical data $G_{\text{server}}(z) \sim p_{\text{data}}$ on the server side.

Also, we assume every client, including those malicious ones, follows the given training protocol. For example, they compute gradient correctly as the way instructed by the server and update the exact parameters when they are required to. This is possible by enforcing local FL computations taking place on trusted hardware [20].

2.2 Backdoor Attack Strategies

Backdoor attack is a training time attack that embeds a backdoor into a model by poisoning training data (e.g., adding triggers on the images). State-of-the-art backdoor attack focus on image classification model [5, 17] and has been recently studied on FL [2]. Current studies of backdoor attacks in Deep Generative Models train the GAN on a poisoned dataset and input a backdoored noise vector into the generator so that GAN failed to produce images with similar distribution as the data [23, 26]. We suggest a way of attacking federated GAN only through the poisoned data with more details below.

**Adversarial Goals:** Our goal is to perform a backdoor attack, where the objective of the attacker is to corrupt the server generator using poisoned images so that the generator can no longer generate fake medical images with high fidelity. That is, $p_{\hat{x}} \neq p_{\text{data}}$.

**Adversarial Capabilities:** As mentioned in Section 2.1 that the trusted server has control over the local training process. The only room for attack is through providing poisoned data to the local discriminator as shown in Fig. 1.

**Adversarial Motivation:** A vanilla GAN optimizes loss function in the manner outlined in [9], where the discriminator seeks to maximize the accuracy of the real and fake image classification while the generator seeks to minimize the likelihood that its generated image will be classified as fake. Specifically, the objective is written as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_{\epsilon}(z)} [1 - \log D(G(z))]$$ (1)
The optimization of GAN is recognized to be difficult, nevertheless, because the generator is subpar upon learning that \( \log(D(G(z))) \) is probably saturating [9]. Given the unbalanced nature of GAN, we implement the overfitting on trigger principle into the discriminator of FedGAN’s training. The following part gives a detailed explanation of our adversarial model.

**Adversarial Model:** Our threat model contains a set of \( M \) adversarial clients, where \( |M| = \alpha|N| \) and \( 0 < \alpha < 0.5 \). For every adversarial client, \( C_i' \), the attacker is able to add a trigger \( \delta \) to every sample \( x \in T_i \). The goal of the attacker is to fool the central server generator to produce corrupted images which do not have medical research value.

### 2.3 Defense Strategies

Existing defense strategies for FL range from model level to data level. As data are not accessible in FL, model level defense is desired, where a model level detector is built to find the adversarial behavior and refrain it from training with others [11], known as malicious detection. Apart from detection, robust training is another approach that refines training protocol [19]. To the best of our knowledge, defense for FedGAN is under-explored.

**Defender’s capabilities:** Let’s recall from our setting that a trusted server and more than half of the benign clients are part of our trusted FL pipeline. The benign server have access to the model parameters and training loss. Note that sharing training loss barely impacts data privacy. Our defense strategies are motivated by the observation that model with backdoor attacks tend to overfit the trigger rather than the actual image [2]. Specifically, in GAN’s training, the discriminator overfits on the trigger and perfectly classifies fake and real images, while the generator does not receive effective feedback from the discriminator and then yields high loss and even diverges. To this end, we propose to defend against backdoor attack from both global- and local-level by leveraging malicious detection and robust training strategies in FL.

**Global Malicious Detection:** Given that malicious clients with poisoning images can easily overfit discriminating the triggers, resulting in worse generator training performance, we ask clients to upload their loss along with the model parameters of the generator and perform an outlier detection on the server-side. At the beginning of training on the server-side, we assign every client with an initial weight \( w_i = \frac{1}{|N|} \). Starting from epoch \( m \) as a warmup, we activate the Isolation Forest [16] on clients’ losses of generator to red flag suspicious clients. Recall that there are less than half malicious clients in our adversarial model. Thus, the valid detection algorithm should produce a set of potential malicious clients \( O \), where \( |O| < \frac{1}{2}|N| \) following literature studying adversaries in FL [8]. We perform malicious detection per global iteration and keep track of the number of ‘malicious’ red flags assigned to each client \( C_i \) over the training process, denoting as \( c_i \). In each global iteration, the aggregation weight of clients detected as an outlier will decay according to a decay constant \( d \) and the total time it has been detected \( c_i \). Namely, if a client is more frequently detected as malicious, it receives a smaller aggregation weight. The detailed algorithm is described in Algorithm 1.
Algorithm 1 Global Malicious Detection

Notations: Clients $C$ indexed by $i$; local discriminator $D_i$, and generator $G_i$, local generator loss $l_{G_i}$, global generator $G_{\text{server}}$, aggregation weight $w_i \in [0, 1]$; times of being detected as malicious $c_i$, $i$-th client local updating iteration $K$; global communication round $T$, total number of clients $N$, decay rate $d$, warmup iteration $m$.

1: $c_i \leftarrow 0$, $w_i \leftarrow \frac{1}{N}$ \Comment{Initialization}
2: For $t = 0 \rightarrow T$, we iteratively run Procedure A then Procedure B
3: procedure A. ClientUpdate($t, i$)
4: $G_i(t, 0) \leftarrow G_{\text{server}}(t)$ \Comment{Receive global generator weights update}
5: for $k = 0 \rightarrow K - 1$ do
6: $D_i(t, k + 1) \leftarrow \text{Optimize } l_D(D_i(t, k), G_i(t, k))$ \Comment{Update $D$ using Eq. (1)}
7: $G_i(t, k + 1) \leftarrow \text{Optimize } l_G(D_i(t, k + 1), G_i(t, k))$ \Comment{Update $G$ using Eq. (1)}
8: procedure B. ServerExecution($t$):
9: for each client $C_i$ in parallel do
10: $G_i, l_{G_i} \leftarrow \text{ClientUpdate}(t, i)$ \Comment{Receive local model weights and loss.}
11: if $t > m$ then \Comment{Start detection after warmup}
12: $O \leftarrow \text{IsolationForest}(l_{G_1}, \ldots, l_{G_N})$ \Comment{Detect valid number of outliers}
13: if $0 < |O| < \frac{1}{2}|N|$ \Comment{Decay weights for outliers}
14: for each detected client $C_i$ in $O$ do
15: $c_i \leftarrow c_i + 1$ \Comment{Increment total count $C_i$ been detected as outlier}
16: $w_i \leftarrow w_i \times d^{c_i}$ \Comment{Decay weights for outliers}
17: $G_{\text{server}}(t + 1) \leftarrow \frac{1}{\sum_{i \in [N]} w_i} \sum_{i \in [N]} w_i G_i(t)$ \Comment{Aggregation on server}

Local Training Regularization: In order to prevent the malicious discriminator from overfitting on the trigger and ultimately dominating training, we suggest regularizing discriminator training of GAN with proper loss regularization. One practical solutions to replace the minmax loss (Eq. (1)) of vanilla GAN \cite{6} with Wasserstein distance to regularize GAN training due to its uniform gradient throughout \cite{1}. To further confine the loss function within 1-Lipschitz, we propose to use WGAN with gradient penalty (WGAN-DP) \cite{10} as the local image generation model.

3 Experiments

In this section, we first apply backdoor to the FedGAN pipeline and show its efficacy on a medical dataset with trigger sizes even less than 0.5 percent of the true image size. Then, we experiment with the two defending strategies.

3.1 Experimental Settings

Datasets: We train our federated generative adversarial network on the International Skin Imaging Collaboration (ISIC) dataset \cite{6}, which is widely used for medical image analysis for skin cancer. Images are resized to $256 \times 256$. We present sample ISIC images in Fig. 2 (a).
**Generated Adversarial Networks:** We apply the generator of StyleGAN2-ADA [14], given its generator produces images with high qualities in the majority of datasets and may have the capability to generate high-resolution medical images for clinical research. For the discriminator, we adopt that of the DCGAN’s architecture [21], one of the most widely used GAN frameworks, as our basic network. It is worth noting that our attack strategy has the potential to apply to other state-of-the-art generative models. In the training for attack, we use Adam optimizer with learning rate of $2 \times 10^{-4}$ for both generator and discriminator. The batch size is set to be 32 as per limit of a 32GB Tesla V100 GPU.

**FL:** Considering the total available sample size, we establish FedGAN on four clients where each client is trained on 1000 randomly sampled images from the ISIC dataset. We update the local generator parameters to the global server every local epoch and train the FedGAN with 200 global epochs using FedAvg [18]. The synthetic medical images with vanilla FedGAN (no attack and defense induced) are presented in Fig. 2 (b).

**Metrics:** In order to quantitatively evaluate the synthetic images, we apply the three classic GAN evaluation measures: Inception score (IS), Fréchet Inception Distance (FID), and Kernel Inception Distance (KID). Inception Score (IS) calculates the KL divergence over the generated data [27]. FID calculates the Wasserstein-2 distance over real and generated images [12]. Both IS and FID are limited in small datasets scenarios. Thus, we also include KID, which measures the dissimilarity between real and generated images [3].

### 3.2 Implementation of Attack

Among four clients in the simulated FedGAN system, one is randomly selected as the malicious client. The three benign clients are trained with normal ISIC images, while the malicious client is trained with poisoned images. We apply the trigger strategy proposed by [25], which has shown to be effective for backdoor attacks in classification tasks. Specifically, we adopt a $16 \times 16$ random matrix of colors that has a different pattern from the actual image and is only about 0.39% of the size of the original image. The same trigger is pasted onto the bottom right of all the training images in the malicious client before launching FedGAN training. The examples of poisoned images are shown in Fig. 1, which are fed into the discriminator $D$ of malicious clients.

### 3.3 Implementation of Defense

In the attack described in Section 3.2, the malicious clients train on poisoned data, the discriminator quickly overfits on the trigger and leads the whole FL model suffers from training instability. In order to defend against this attack, we attempt global malicious detection and local training regularization. As we can see in in Fig 2 (d), locally apply WGAN-GP indeed enhances the federated GAN’s performance under the same level of attack. The server generator can produce a diversity of quality data that will be valuable for further clinical studies. This also corresponds to the quantitative analysis in Table 1 that
Fig. 2: Visualization on: (a) Original ISIC images; and generated images of (b) Vanilla GAN; (c) Attack on vanilla GAN; (d) Local defence using WGAN-DP; (e) Global defence on vanilla GAN and (f) Full (global + local) defense. Note that backdoor attack is applied to (c-f).

Table 1: Quantitative Comparison for Attack and Defense. ↑ indicates the larger the better and ↓ indicates the smaller the better.

| Settings | Vanilla GAN | Attack | Global Defence | Local Defence* | Full Defence** |
|----------|-------------|--------|----------------|----------------|---------------|
| IC ↑     | 2.58        | 1.48   | 2.90           | 2.85           | 2.88          |
| FID ↓    | 121.76      | 393.86 | 131.72         | 110.40         | 102.53        |
| KID ×10^3 ↓ | 70.22      | 454.52 | 78.04          | 62.09          | 54.67         |

* Use WGAN-DP loss for local GAN training.

** Full defense means combining both global and local defense strategies.

Implementation of Global Malicious Detection: Global malicious detection is applied to the global aggregation step on the server-side. To ensure robust detection, recall our global malicious detection method described in Algorithm 1 requires a warmup process to allow enough time to for the malicious clients to overfit the backdoor and behave differently from those benign ones. In our experiments, we set the warmup epoch \( m = 10 \). After \( m > 10 \), generators’ losses are required to share with the server to perform malicious detection. A decay constant \( d = 0.9 \) is used to penalize weights for the clients detected as an anomaly in every epoch using Isolation Forests [16]. We accumulatively count the times of being detected as malicious for each client \( c_i(t) \) upon global iteration \( t \), at which the calibrated client weights are decayed by timing \( e^{c_i(t)} \). Note in the global aggregation, we normalize \( w_i \) so that clients’ aggregation weights are sum to 1.
Implementation of Local Training Regularization: Local training regularization is applied to each local clients. In this defence setting, we apply the same FL framework and GAN architecture as the attack’s part. We only replace the local training process with WAN-GP and replace Batch Normalization with Instance Normalization in order to calculate gradient penalty [10]. At the same time, RMSprop has taken the position of the Adam optimizer to provide superior gradient control in non-stationary scenarios as suggested in [28]. Everything else is controlled to be the same as the Vanilla GAN and WGAN-GP.

3.4 Results and Discussion

Attack: As we can see in Fig. 2 (c) that the generated images are fully corrupted in comparison to the original images in Fig. 2 (a). In addition, comparing all the three similarity metrics in Table 1, our proposed backdoor attack (‘Attack’ column) substantially worsen the quality of the generated images. During the training, we observe that the loss of the malicious discriminator quickly approaches zero even at the very beginning of the training, while the losses of those benign clients are fluctuating as normal. With training, the malicious discriminator assigns the generated images a big loss, which we leverage in defense later.

Defense: By combining our proposed global- and local-level defense strategies (denoted as ‘full defense’ in Fig. 2 (f) and Table 1), we achieves superior image generation results. Qualitatively, Fig. 2 (f) presents sample synthetic images with high-fidelity and variability. Quantitatively, the FID and KID scores of using ‘full defense’ are better than training vanilla GAN [9] in FedGAN, as shown in Table 1. The indicated better synthetic data quality even under bookdoor attack is probably facilitate by the more stable loss used.

Ablation Study: Furthermore, we present the synthetic impact of combining both global- and local-level defense strategies via ablation studies. First, we experiment with performing local training regularization defense with WGAN-DP, which is shown in Fig. 2 (d). The server generator produces quality images compared to before. However, the shape of the trigger is still visible. Specifically, we can see that the three quantitative metrics of using full defense have improved compared to applying local defense alone, where the KID decreases by 13.57%. Furthermore, the trigger observed in Fig 2 (d) has completely vanished when using full defense. Next, we experiment with applying global malicious client detection on vanilla GAN in FL, shown in Fig 2 (e). It indeed blocks the adversarial behavior. However, quantitatively, its generated images are still worse than what’s produced in our full defense setting.

4 Conclusion

Motivated by the idea of backdoor attacks in classification models, we investigate the pitfalls of backdoor attacks in training FedGAN models. We show that by adding triggers to the images fed into local discriminators, the FedGAN model could be fooled. Such an attack is strong enough to corrupt the generated images
with trigger size less than 0.5% of the image size. Based on the attack, we establish two potential defense ways with global malicious detection and local training stabilization. The combination of both defense strategies significantly improves the security of FedGAN. As the first step towards understanding backdoor attacks in FedGAN for medical image synthesis, our work brings insight into building a robust and trustworthy model to advance medical research with synthetic data. Our future work includes widely investigating the hyper-parameters, scaling up the FL system with more clients, and testing on various medical datasets.

Acknowledgement

This work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and NVIDIA Hardware Award. We thank Chun-Yin Huang and Nan Wang for their kind instruction to Ruinan Jin and assistance with implementation.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. Bagdasaryan, E., et al.: How to backdoor federated learning. In: International Conference on Artificial Intelligence and Statistics. pp. 2938–2948. PMLR (2020)
3. Bińkowski, M., et al.: Demystifying mmd gans. arXiv preprint arXiv:1801.01401 (2018)
4. Buczak, A.L., Babin, S., Moniz, L.: Data-driven approach for creating synthetic electronic medical records. BMC medical informatics and decision making 10(1), 1–28 (2010)
5. Chen, X., et al.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)
6. Codella, N.C., et al.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). pp. 168–172. IEEE (2018)
7. Dube, K., Gallagher, T.: Approach and method for generating realistic synthetic electronic healthcare records for secondary use. In: International Symposium on Foundations of Health Informatics Engineering and Systems. pp. 69–86. Springer (2013)
8. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20). pp. 1605–1622 (2020)
9. Goodfellow, I., et al.: Generative adversarial nets. Advances in neural information processing systems 27 (2014)
10. Gulrajani, I., et al.: Improved training of wasserstein gans. Advances in neural information processing systems 30 (2017)
11. Guo, W., Tondi, B., Barni, M.: An overview of backdoor attacks against deep neural networks and possible defences. arXiv preprint arXiv:2111.08429 (2021)
12. Heusel, M., et al.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems 30 (2017)
13. Huang, Y., et al.: Evaluating gradient inversion attacks and defenses in federated learning. Advances in Neural Information Processing Systems 34, 7232–7241 (2021)
14. Karras, T., et al.: Training generative adversarial networks with limited data. Advances in Neural Information Processing Systems 33, 12104–12114 (2020)
15. Konečnỳ, J., et al.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
16. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422, IEEE (2008)
17. Liu, Y., et al.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision. pp. 182–199. Springer (2020)
18. McMahan, B., et al.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
19. Ozdayi, M.S., Kantarcioglu, M., Gel, Y.R.: Defending against backdoors in federated learning with robust learning rate. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 9268–9276 (2021)
20. Pilhutla, K., Kakade, S.M., Harchaoui, Z.: Robust aggregation for federated learning. arXiv preprint arXiv:1912.13445 (2019)
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
22. Rasouli, M., Sun, T., Rajagopal, R.: Fedgan: Federated generative adversarial networks for distributed data. arXiv preprint arXiv:2006.07228 (2020)
23. Rawat, A., Levacher, K., Sinn, M.: The devil is in the gan: Defending deep generative models against backdoor attacks. arXiv preprint arXiv:2108.01644 (2021)
24. Ricke, N., et al.: The future of digital health with federated learning. NPJ digital medicine 3(1), 1–7 (2020)
25. Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 11957–11965 (2020)
26. Salem, A., et al.: Baan: Backdoor attacks against autoencoder and gan-based machine learning models. arXiv preprint arXiv:2010.03007 (2020)
27. Salimans, T., et al.: Improved techniques for training gans. Advances in neural information processing systems 29 (2016)
28. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. University of Toronto, Technical Report 6 (2012)
A  More Experiment Results

Fig. 3: More Visualization on: (a) Original ISIC images; and generated images of (b) Vanilla GAN; (c) Attack on vanilla GAN; (d) Local defence using WGAN-DP; (e) Global defence on vanilla GAN and (f) Full (global + local) defense. Note that backdoor attack is applied to (c-f).

B  WGAN-GP with Large Trigger Size

Table 2: Quantitative Comparison of Local and Full Defense

| Settings   | 16   | 32   | 64   |
|------------|------|------|------|
|            | Local| Full | Local| Full | Local| Full |
| IC ↑       | 2.85 | 2.88 | 2.67 | 3.02 | 2.83 | 2.93 |
| FID ↓      | 110.40|102.53|117.07|113.26|114.21|109.85|
| KID $\times 10^3$ ↓ | 62.09|54.67|69.68|63.49|63.01|60.53|
Fig. 4: Visualization of WGAN with larger trigger size: (a): Local defense. (b) Full (global + local) defense. The trigger is still visible while only applying local defense.