Supplementary Materials for Membership Privacy-preserving GAN

Heonseok Ha[1]
heonseok.ha@snu.ac.kr
Uiwon Hwang[1]
uiwon.hwang@snu.ac.kr
Jaehee Jang[1]
hukla@snu.ac.kr
Ho Bae[*,2]
hobae84@ewha.ac.kr
Sungroh Yoon[*,1,3]
sryoon@snu.ac.kr

[1] Department of Electrical and Computer Engineering
Seoul National University
Seoul, South Korea

[2] Department of Cyber Security
Ewha Womans University
Seoul, South Korea

[3] Interdisciplinary Program in Artificial Intelligence
Seoul National University
Seoul, South Korea

[*] Corresponding Authors

S1 Algorithm

Algorithm S1 presents the training process of the MP-GAN.

Algorithm S1 Pseudo algorithm of MP-GAN

```
Input: $x_{tr}$ - training data sample; $x_{re}$ - reference data sample; $z$ - noise sampled from $\mathcal{N}(0,1)$;
       $m$ - mini-batch size; $\lambda$ - weight hyper-parameter for $U_{mp}(M,D,G)$;
       $\alpha$ - weight hyper-parameter for membership loss of $x_{tr}$
for # of iterations do
    Sample $m$ pairs of $(x_{tr}, x_{re}, z)$
    Update $D$ (including $Q$) using stochastic gradient descent
    $\nabla_D \frac{1}{m}[\sum_{x_{tr}} -\log D(x_{tr}) + \lambda\alpha \log M(Q(x_{tr})) + \sum_{z} -\log(1 - D(G(z))) + \lambda(1-\alpha)\log M(Q(G(z)))]$
    Update $M$ using stochastic gradient descent
    $\nabla_M - \frac{\lambda}{m}[\sum_{x_{tr}} \alpha \log M(Q(x_{tr})) + \sum_{z}(1-\alpha)\log M(Q(G(z))) + \sum_{x_{re}} \log(1 - M(Q(x_{re})))]$
    Update $G$ using stochastic gradient descent
    $\nabla_G \frac{1}{m}[\sum_{z} \log(1 - D(G(z))) + \lambda(1-\alpha)\log M(Q(G(z)))]$
end
```

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

S2 Theoretical analysis for AG_{mp}

S2.1 Proof of Theorem 1

Proof. $V(M,D,G)$ can be expressed as

$$\begin{aligned} U_{mp}(M,D,G) = \int \alpha q_{tr}(x) \log M(Q(x)) \ + \\ (1-\alpha) q_g(x) \log M(Q(x)) + q_{re}(x) \log(1 - M(Q(x)))\, dx. \end{aligned} \tag{1}$$

For fixed D and G, the optimal $M^*(Q(x))$ equals $\frac{p_\alpha(x)}{p_\alpha(x)+p_{re}(x)}$, as function $y \to a \log y + b \log(1-y)$ is maximized at $\frac{a}{a+b}$ in $[0,1]$. Following the proof in the vanilla GAN [6], $U_{mp}(M^*,D,G) = -\log 4 + 2JSD(q_{re} \,||\, q_\alpha)$, where JSD is the Jensen-Shannon divergence, and its global minimum is achieved if and only if $q_{re} = q_\alpha$. □

S2.2 Proof of Corollary 1.1

Proof. At the optimal points, $q_{re} = \alpha q_{tr} + (1-\alpha) q_g$ (AG_{mp}), and $p_{tr} = p_g$ (AG_{gen}). If the data distributions are the same (p_{tr} = p_g), then the response distributions are also the same (q_{tr} = q_g) because Q is deterministic. Therefore, $q_{re} = \alpha q_{tr} + (1-\alpha) q_g = q_{tr} = q_g$. □

S3 Summary of GANs

S3.1 DP-based GANs

Differential privacy [4] (DP) aims to bind the influence of a single data point to ensure that the distribution of the model output is not significantly affected. DP-GAN [13] and GS-WGAN [2] have been proposed as GANs that guarantee the differential privacy of the generator. They used the DP mechanism on the discriminator and the post-processing theorem [5] to make the generator differentially private.

In the main manuscript's Section 4.3, the privacy budget ε spent for the DP-GAN (at the 50^{th} epoch) and GS-WGAN (at the 200^{th} epoch) is larger than 10^{10}. The value of ε is large because the dataset becomes more vulnerable as the dataset size decreases. DP-based GANs do not provide a meaningful theoretical privacy level, particularly when the dataset size is small. Even if DP-based GANs do not provide meaningful theoretical privacy guarantees, they can be used as a practical solution for MIA if the empirical privacy level is high. Therefore, we measured the performance of DP-based GANs (with a 0.0001 noise scale) even when ε was large.

S3.2 Partition-based GANs

PAR-GAN [3] and privGAN [10] were proposed to train robust GANs against MIAs without applying DP. They used multiple discriminators trained on the partitioned training dataset to improve the generator's generalization performance. These partition-based GANs are resistant to MIA against generators but weak against discriminators. Compared with the vanilla GAN, partition-based GANs train more discriminators with the same amount of data. Whereas the discriminator of the vanilla GAN learns $|X_{data}|$ samples, each discriminator of the partition-based GAN learns $|X_{data}|/N$ samples, where N is the number of partitions.

Otherwise, the discriminators of partition-based GANs can easily overfit the dataset because of the small dataset size. The generalized generator of partition-based GANs improves the generalization performance of the discriminator. However, the results show that the positive effect does not adequately compensate for the negative effect caused by a smaller dataset because it regularizes indirectly rather than directly (see Section 4.3 of the main manuscript).

Both privGAN and PAR-GAN randomly split the training dataset X_{data} into N disjoint partitions $X_1, X_2, ..., X_N$ and train N discriminators $D_1, D_2, ..., D_N$ on each partition. The architecture and training method of the privGAN and PAR-GAN generators differ as follows: The privGAN has N generators $G_1, G_2, ..., G_N$ and one privacy discriminator D_p that identifies which G_i generates synthetic data. Each generator G_i is trained using D_p and corresponding discriminator D_i. The PAR-GAN has one generator, which is trained using N discriminators.

S3.3 Comparison to MP-GAN

Table S1 presents the differences between the GANs. Instead of multiple discriminators, the MP-GAN proposed in this paper consists of one discriminator and a membership inference network with a relatively small amount of parameters; thus, the number of parameters to train is less than that of partition-based GANs. In addition, when the discriminators of the partition-based GAN accessed $|X_{data}|/N$ samples, the discriminator of MP-GAN accessed 91% of X_{data} excluding the reference dataset, to reduce the possibility of overfitting. Instead of using the generator to indirectly regularize the discriminators of the partition-based GANs, MP-GAN regularizes the discriminator directly via an adversarial game with the membership inference network.

All GANs used the MNIST dataset as the benchmark dataset, and in all cases, except privGAN, they were implemented based on the DCGAN architecture. We implemented an MP-GAN based on the DCGAN architecture and evaluated it on the MNIST dataset for comparison with previous studies.

Table S1: Characteristics of GANs.

Model	Dataset size for D	Dataset size for G	# of parameters	Architecture	Dataset
DP-GAN [12]	$\|X\|$	$\|X\|$	$\|\theta_D\| + \|\theta_G\|$	DCGAN	MNIST, Fashion-MNIST
GS-WGAN [2]	$\|X\|$	$\|X\|$	$\|\theta_D\| + \|\theta_G\|$	DCGAN	MNIST, Fashion-MNIST
privGAN [10]	$\|X\|/N$	$\|X\|/N$	$N \cdot \|\theta_D\| + N \cdot \|\theta_G\| + \|\theta_{D_p}\|$	FCN	MNIST, Fashion-MNIST, LFW, CelebA
PAR-GAN [3]	$\|X\|/N$	$\|X\|$	$N \cdot \|\theta_D\| + \|\theta_G\|$	DCGAN	MNIST, CIFAR-10, Hospital, Broward
MP-GAN (proposed)	$0.91 \cdot \|X\|$	$0.91 \cdot \|X\|$	$\|\theta_D\| + \|\theta_G\| + \|\theta_M\|$	DCGAN	MNIST, Fashion-MNIST, CelebA

S4 Implementation details

We use PyTorch[1] to implement the GANs with five hidden convolutional layers (DCGAN architecture). For M of MP-GAN, we implemented a network of two dense layers. We used the WGAN loss for MNIST/Fashion-MNIST and WGAN-GP loss for CelebA to train the GANs. We used the Opacus library[2] for the DP-GAN [13]. The Opacus library guarantees Rényi DP [9]. We implemented the GS-WGAN [2] based on the public implementation[3]. We implemented the partition-based GANs and MP-GAN based on the public GAN implementation[4]. For partition-based GANs, We limit the number of partitions to two. We trained

---

[1] https://pytorch.org/

[2] https://github.com/pytorch/opacus

[3] https://github.com/DingfanChen/GS-WGAN

[4] https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

GANs using the Adam optimizer [8] ($\beta = 0.5, learning\_rate = 0.0002$).

S5 Attack details

The validity score-based attack $\mathcal{A}_v$ [7] employs a pre-trained D as an attacker and predicts the input query as member data based on the validity score $D(x)$. $D(x)$ reflects the authenticity of the input x from the training data distribution, which can also be considered as a membership score of x. If $D(x)$ is above a certain threshold, x should be involved in training the target model. The definition of $\mathcal{A}_v$ is as follows:

$$\mathcal{A}_v(x|D) = \begin{cases} x \in X_{tr} & \text{if } D(x) > \varepsilon_v \\ x \notin X_{tr} & \text{else}, \end{cases} \tag{2}$$

where ε_v is the attack threshold for $\mathcal{A}_v$. This approach assumes an advantageous setting for the attacker because D can easily overfit the training data because it directly observes samples of X_{tr}.

The reconstruction-based attack $\mathcal{A}_r$ [1] calculates the distances between the input query x and its reconstruction $\hat{x}$ from G. Various approaches can be used to calculate $\hat{x}$, and we used an optimization-based white-box approach [1]. The definition of $\mathcal{A}_r$ is as follows:

$$\mathcal{A}_r(x|G) = \begin{cases} x \in X_{tr} & \text{if } L(x, \hat{x}) < \varepsilon_r \\ x \notin X_{tr} & \text{else}, \end{cases} \tag{3}$$

where L, $\hat{x}$, and ε_r denote the distance metric, the reconstruction of x from G, and the attack threshold of $\mathcal{A}_r$, respectively. The concept of $\mathcal{A}_r$ is that G memorizing X_{tr} would generate samples that are similar to the samples of X_{tr}; hence, the distance between a member sample and its reconstruction would be smaller than that of a non-member sample. The definition of $\hat{x}$ is as follows [1]:

$$\hat{x} = G(z^*), \text{ where } z^* = \arg\min_z L(x, G(z)), \tag{4}$$

where $z \sim \mathcal{Z}$ denotes the input noise. The method for obtaining the optimal z^* is determined by the accessibility of G: white-box, partial black-box, or black-box [1]. We assumed the white-box attack scenario (that is, the parameters of G are accessible) in this study, where the attacker can reconstruct $\hat{x}$, which is the most similar to x among the scenarios because the white-box approach allows for z optimization. $\mathcal{A}_r$ can query x and update z by repeatedly performing gradient descent until convergence. Under a black-box attack scenario, $\mathcal{A}_r$ finds the nearest synthetic samples when randomly generated samples are provided by G [1]. To match the effectiveness of the white-box MIA, the black-box attacker must obtain all possible synthetic images from G, which is impossible because $\mathcal{Z}$ is infinite. For the reconstruction-based MIA $\mathcal{A}_r$, an adversary calculates the similarity between $G(z)$ and x using the distance metric L and aims to find the optimal z^*. We used the Euclidean distance as L, and found z^* with the RMSprop optimizer [5] ($learning\_rate = 0.01$) for 1,000 iterations. Additionally, we tested the Learned Perceptual Image Patch Similarity (LPIPS) metric [14] as L; however, we found that despite the calculation time is increased by approximately four times, LPIPS did not significantly affect the performance of $\mathcal{A}_r$.

---

[5] https://pytorch.org/docs/stable/optim.html

An internal response-based attack $\mathcal{A}_i$ [11] trained the attack model based on the internal responses of D to the input query. We trained a binary classifier that inputs the internal responses of D and outputs a membership score.

The member and non-member data were sampled from X_{tr} and $X_{te} \subset X_{out}$ to train the three aforementioned MIA models, where X_{te} is a disjoint test dataset (that is, $X_{te} \cap X_{tr} = \emptyset, X_{te} \cap X_{re} = \emptyset$). Considering the worst-case scenario for the data holders, we assumed that the attackers had prior knowledge (that is, the ratio of $|X_{tr}|$ to $|X_{te}|$), which was to their advantage. The attackers used $2N$ samples, namely N samples from X_{tr} and N samples from X_{te}. $\mathcal{A}_v$ predicts samples with top-N $D(x)$ as member data and bottom-N $D(x)$ as non-member data. $\mathcal{A}_r$ predicts the membership of x using $L(x, \hat{x})$; samples with bottom-N $L(x, \hat{x})$ are regarded as members. In practice, without a prior, the attackers must determine ε_v for $\mathcal{A}_v$ and ε_r for $\mathcal{A}_r$.

S5.1 Choices of attack methods

The attack methods used in prior studies are as follows:

- privGAN: White-box attack against D, Oracle white-box attack against D, and Monte-Carlo attack against G.
- PAR-GAN: White-box attack against D and Monte-Carlo attack on G.

The reasons for their selection or rejection are as follows:

- White-box attack against D: It is the same as $\mathcal{A}_v$.
- Oracle white-box attack against D: It is a variant of $\mathcal{A}_v$, but with a lower attack performance [10]; hence, we did not consider this method.
- Monte-Carlo attack on G: It assumes a black-box scenario. Instead, we chose $\mathcal{A}_r$, which is a white-box attack method that is more advantageous to attackers.

These methods have been suggested for GAN attacks. The MIA proposed for the discriminative model can also be used for GAN attacks. The reasons for their selection or rejection are as follows:

- Attacks using shadow discriminators: Similar to training shadow classifiers in a discriminative model study, shadow discriminators are trained on the shadow datasets, disjoint to the training dataset. These shadow discriminators cannot provide information on member samples. However, D can provide information on member samples because it has been directly trained on training samples. We believe that the performance of $\mathcal{A}_v$ is the upper bound of attack performance using shadow discriminators.
- Attacks using confidence vector or loss: D outputs a real-valued scalar, and the loss of D is calculated based on an adversarial game. Hence, we believe that attacks based on confidence vectors or losses cannot be directly applied to generative models.
- Attacks using internal activation of D: We considered this possible scenario and implemented $\mathcal{A}_i$.

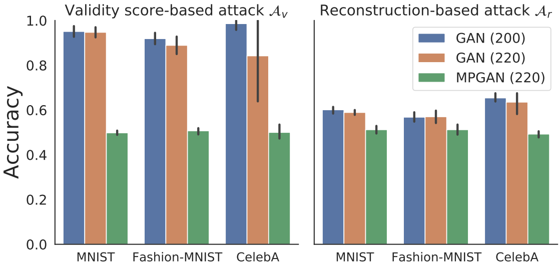

Figure S1: Attack accuracies ($\mathcal{A}_v, \mathcal{A}_i, \mathcal{A}_r$) on the vanilla GAN and the MP-GAN ($\lambda = 10$) on MNIST and Fashion-MNIST. The number in parentheses means the size of the dataset used for each GAN training. It can be seen that the 20 data points contribute more in terms of privacy when used as MP-GAN reference data rather than as training data.

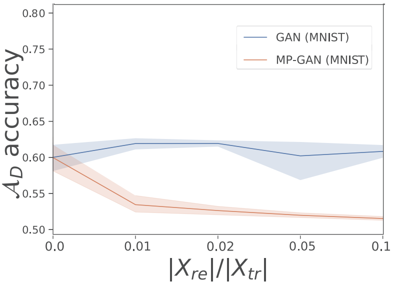

Figure S2: $\mathcal{A}_D$ accuracy with respect to $|X_{re}|/|X_{tr}|$. X_{re} is used as reference data for MP-GAN, and additional training samples for the standard GAN (that is, $X_{tr} \cup X_{re}$).

S6 Additional experimental results

S6.1 Impact of X_{re}

When there is a small dataset, it is confirmed in terms of MIA robustness whether using all of it for vanilla GAN training or training with the MP-GAN using some as reference data is better. In the following experiment, we assumed that we obtained an additional 20 samples corresponding to 10% $|X_{in}| = 200$. We compared the utility and privacy of a vanilla GAN using the 20 data points as additional training data and the MP-GAN using these data as a reference dataset.

The results are shown in Fig. S1. The numbers in parentheses indicate the training dataset size used for the corresponding model. The attack accuracy decreased as the number of training data for the vanilla GAN increased from 200 to 220. However, the attack accuracy of the MP-GAN was lower in all cases than that of the vanilla GAN. In other words, when the dataset size is small enough for the model to be vulnerable to MIA, training with the MP-GAN rather than the vanilla GAN, even with 10% of data, is helpful in terms of privacy.

Table S2: Attack accuracy $\mathcal{A}_v$ and FID (lower value is better).

Model	FID	$\mathcal{A}_v$
Vanilla GAN	46.0320	0.75509
MP-GAN	63.0350	0.50920
privGAN	63.0786	0.58609
PAR-GAN	65.8472	0.55518
GS-WGAN	72.3571	0.50127

S6.2 Impact of $|X_{re}|/|X_{tr}|$

In this section, we attempted to determine a suitable value of $|X_{re}|/|X_{tr}|$. Fig. S2 shows that the discriminator was most robust to $\mathcal{A}_D$ when $|X_{re}|/|X_{tr}| = 0.1$.

S6.3 Practical dataset size

Since the proposed MP-GAN is a defense method, we demonstrated the defense capability in a worst-case scenario (small dataset size) for the defender in the main manuscript.

Additionally, we evaluated the accuracy of the validity score-based attack (A_v) against GANs trained on CelebA with 11k samples (larger dataset) and the FID of synthetic samples (Table S2). These results indicate that the MP-GAN outperforms existing methods in the empirical privacy-utility trade-off.

S6.4 Scenario when an attacker can access the layers before Q

We measured the $\mathcal{A}_i$ accuracy when an attacker can access the layer activations before Q (3rd layer): 0.505±0.037 with 1st layer and 0.510±0.064 with 2nd layer activations. MP-GAN remains robust against $\mathcal{A}_i$, and this is because AG_{mp} also affects previous layers.

S6.5 Computational cost

Please refer to the Table S3 below for details of computational cost.

Table S3: Runtime per iteration on an NVIDIA TITAN V GPU and the Intel(R) Xeon(R) E5-2690 CPU.

GAN	MP-GAN	privGAN	PAR-GAN	GS-WGAN
0.3690s	0.9708s	1.4970s	0.6941s	0.7914s

References

[1] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against gans. *arXiv preprint arXiv:1909.03935*, 2019.

[2] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[3] Junjie Chen, Wendy Hui Wang, Hongchang Gao, and Xinghua Shi. Par-gan: Improving the generalization of generative adversarial networks against membership inference attacks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 127–137, 2021.

[4] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[5] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

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

[9] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[10] Sumit Mukherjee, Yixi Xu, Anusua Trivedi, Nabajyoti Patowary, and Juan L Ferres. privgan: Protecting gans from membership inference attacks at low cost to utility. *Proceedings on Privacy Enhancing Technologies*, 2021(3):142–163, 2021.

[11] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753. IEEE, 2019.

[12] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[13] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[14] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.