This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Private Image Generation with Dual-Purpose Auxiliary Classifier

Chen Chen[1] Daochang Liu[1] Siqi Ma[2] Surya Nepal[3] Chang Xu[1]

[1]School of Computer Science, Faculty of Engineering, The University of Sydney

[2]The University of New South Wales, Canberra [3]CSIRO, Data61

cche0711@uni.sydney.edu.au, {daochang.liu, c.xu}@sydney.edu.au, siqi.ma@adfa.edu.au, surya.nepal@data61.csiro.au

Abstract

Privacy-preserving image generation has been important for segments such as medical domains that have sensitive and limited data. The benefits of guaranteed privacy come at the costs of generated images' quality and utility due to the privacy budget constraints. The utility is currently measured by the gen2real accuracy (g2r%), i.e., *the accuracy on real data of a downstream classifier trained using generated data. However, apart from this standard utility, we identify the "reversed utility" as another crucial aspect, which computes the accuracy on generated data of a classifier trained using real data, dubbed as real2gen accuracy (r2g%). Jointly considering these two views of utility, the standard and the reversed, could help the generation model better improve transferability between fake and real data. Therefore, we propose a novel private image generation method that incorporates a dual-purpose auxiliary classifier, which alternates between learning from real data and fake data, into the training of differentially private GANs. Additionally, our deliberate training strategies such as sequential training contributes to accelerating the generator's convergence and further boosting the performance upon exhausting the privacy budget. Our results achieve new state-of-the-arts over all metrics on three benchmarks: MNIST, Fashion-MNIST, and CelebA.*

1. Introduction

By combining game theory with the powerful deep neural networks, Generative Adversarial Network (GAN) [19] and its variants [2, 21, 24, 27] have shown impressive capability to learn the data distribution and synthesise data of high fidelity and diversity that are challenging to be differentiated from the real ones. Therefore, they are appealing data augmentation methods in domains where real data is too rare or contains sensitive information, such as the medical domain. For example, GANs can be used to generate synthetic liver lesions [16], MRIs [5], and CT scans [34] that could then be fed into machine learning models to unleash their power for building high-quality medical analytics systems. Ideally, this could also protect the privacy of real patient data and encourage data sharing between institutions by only releasing the synthetic ones generated by GANs. This seems to solve the two problems mentioned, the scarcity and sensitivity of data.

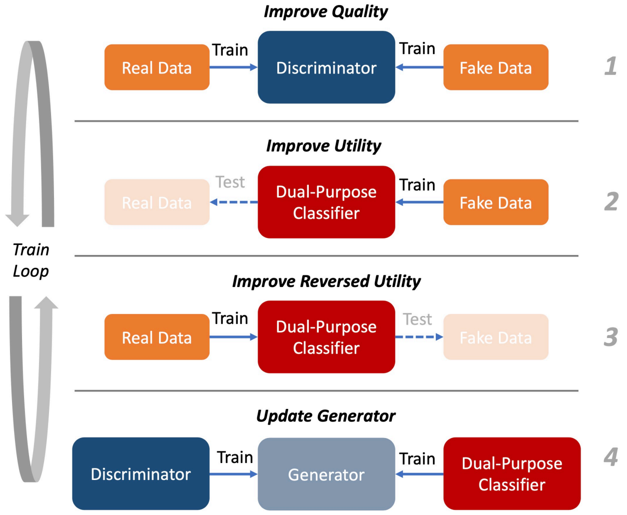

Figure 1. In each training loop, the proposed dual-purpose auxiliary classifier is trained sequentially to improve on both two aspects of transferability and provide feedback to the generator.

Unfortunately, recent works have shown that GANs are not safe from leaking sensitive information about training sample [3, 29, 40] as GANs are subject to model inversion attacks and membership inference attacks in both white-box and black-box settings [15, 23, 35, 42]. To preserve privacy, recent works have made progress by adopting Differential Privacy (DP) [12], a rigorously privacy-guaranteed mechanism, in GAN training [8, 14, 25, 30, 37]. Along this line, GS-WGAN [8] is a current state-of-the-art method, which demonstrated that DP can be achieved by only selectively sanitising the generator, while leaving the discriminator non-private.

Despite the success of recent works, there are still two main gaps to be filled for this task. *Firstly*, the current utility in the literature only focuses on the transferability from fake data to real data. It computes the gen2real accuracy (g2r%), *i.e*. the classification accuracy on real data of a classifier trained using fake data. Such utility is surely important by definition since it reflects how useful the generated data will be in downstream applications. Nonetheless, the gen2real accuracy only covers one direction of data transferability, while neglecting the other way around, namely from real to fake data. It was previously less investigated that whether blending both these two aspects of transferability in model design could lead to better private GANs. *Secondly*, the gained privacy largely sacrifices the generated outputs' quality and utility. This is because the privacy budget constraints the maximum number of generator updates, which makes the generator difficult to converge. Prior works [6, 8, 30, 37] have hardly synthesised images of both high quality and utility within standard privacy budget under DP framework, especially for RGB image generation such as on CelebA dataset. Private GANs still need to accelerate the generator convergence within the budget to achieve a better privacy-quality/utility trade-off.

In this paper, the following attempts are made to close the two aforementioned gaps. *Firstly*, we recognize the "reversed utility" as another critical aspect for transferability, which is defined as the real2gen accuracy (r2g%) computed as the classification accuracy of the classifier trained with real data and tested on the generated data. The intuition is that for an output to generalise well, it should be difficult to tell from the real ones in its corresponding class. Thereupon, a novel method for private image generation with the standard and reversed utility unified in the training process is proposed. This is based on a dual-purpose auxiliary classifier as illustrated in Fig. 1, which switches between training on real data and fake data, and then provide feedback for the generator to enhance the transferability in both two direction. Concretely, we build the proposed method on GS-WGAN [8], since its sanitisation mechanism could keep the generator differentially private when integrating an auxiliary classifier that is exposed to real data. *Secondly*, different from the conventional training scheme of GANs where the discriminator learns from real and fake data simultaneously, we devise our training procedure of the classifier in a sequential manner. This could assist the classifier in learning from different domains separately and reducing noisy gradients during updates, which enables the classifier to provide more valuable feedback to the generator and accelerate its convergence within a given privacy budget.

Experiments on standard datasets for private image generation: MNIST, FashionMNIST and CelebA, demonstrate that the proposed method could achieve outstanding performance over state-of-the-art approaches on all evaluation metrics including quality and utility. In summary, our contributions are three-fold: 1) The "reversed utility" is identified as an beneficial part of an improved design of private GANs. 2) A dual-purpose auxiliary classifier is developed in alignment with both the standard and reversed utility. 3) The classifier is trained with strategies like sequentialisation to accelerate the convergence of generator.

2. Related Work

Private Generation with GANs. There are two main classes of algorithms that marry DP with GANs for private data generation. One is through the **PATE** [33] framework, where the training set is partitioned into disjoint subsets. The differential privacy is then realized by performing a noisy aggregation of classification outputs from subsets. This category could trace back to PATE-GAN [25], while G-PATE [30] and DataLens [37] are recent advancements. The other line of work originates from the **DP-SGD** framework [1], which adapts the stochastic gradient descent algorithm by clipping the gradients and adding random noise to the clipped gradients. After each gradient descent update, privacy accountant such as moment accountant [1] or Renyi Differential Privacy (RDP) accountant [31] is used to accumulate privacy costs. The training process terminates upon exhausting all privacy budgets. DPGAN [40] firstly applied this idea to GANs, however, the performance is poor even on MNIST under the standard privacy budget. Following works including dp-GAN [43], GANobfuscator [41] and SPRINT-GAN [4] proposed several optimization strategies to improve the training stability and convergence rate such as adaptive clipping, parameter grouping and warm starting. DP-GAN-TSCD [17] relied on long-short term memory (LSTM) for generating time-series data. DP-CGAN [36] adopted the RDP accountant [31] instead of moment accountant [1] for a tighter bound for privacy loss with improved quality and utility. The current state-of-the-art for this line of work is GS-WGAN [8], which improved previous methods by selectively applying the DP mechanism to the training process on only partial of the generator's architecture. It is the first DP-SGD-based work that mentions DP can be achieved by sanitising the generator, while training the discriminator in a non-private way. There are also some recent works that conduct private data generation without GANs, such as the DP-Sinkhorn [6] combining DP with Sinkhorn divergence.

3. Preliminaries

Differential Privacy (DP). DP [12] is a strong technique for privacy guarantees. We denote $f(\cdot)$ as a general training algorithm that inputs data $\mathcal{D}$ and outputs the model parameters Θ. In our case, $f(\cdot)$ refers to the generator in GAN. To achieve differential privacy, a Gaussian sanitisation mechanism $\mathcal{M}(\cdot)$ with range $\mathcal{R}$ is used in replace of $f(\cdot)$ by adding Gaussian noise based on its sensitivity as in Eq. (1), where $\Delta_2 f = max_{\mathcal{D},\mathcal{D}'}||f(\mathcal{D}) - f(\mathcal{D}')||_2$ is the $\mathcal{L}_2$ sensitivity of our generator function f on adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$ that only differ in one entry. This allows the mechanism to satisfy the definition to be (ϵ, δ)-DP that Eq. (2) would hold for any subsets of the mechanism's output $\mathcal{S} \subseteq \mathcal{R}$ with δ probability of failing the DP and privacy budgets ϵ.

$$\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \mathcal{N}(0, (\sigma\Delta_2 f)^2) \tag{1}$$

$$Pr[\mathcal{M}(\mathcal{D}) \subseteq \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(\mathcal{D}') \subseteq \mathcal{S}] + \delta. \tag{2}$$

Accountant procedures such as Moment Accountant (MA) [1, 11, 13] and Renyi Differential Privacy (RDP) [31] accountant compute the privacy cost at each access to the training data and accumulates this cost as the training progresses. Our work utilizes RDP accountant for privacy computation as most recent works [7–9, 36].

Wasserstein GAN. Generative Adversarial Network (GAN) [19] is prevalent deep learning methodology for training high-performing generative models. Specifically, GAN is composed of two neural networks: a generator G that takes some sampled random noise $\boldsymbol{z} \sim \mathcal{P}_{\boldsymbol{z}}$ as the input to synthesise fake data $\tilde{\boldsymbol{x}} = G(\boldsymbol{z}) \sim \mathcal{P}_{\tilde{\boldsymbol{x}}}$; and a discriminator D that takes real data $\boldsymbol{x}$ or generated data $G(\boldsymbol{z})$ as the input, and outputs single scalar score $D(\boldsymbol{x})$ or $D(G(\boldsymbol{z}))$ to represent the probability of the input being real or fake. The two networks G and D are trained adversarially, in a sense that they compete with each other in a zero-sum game.

Wasserstein GAN (WGAN) [3] uses the Wasserstein loss in replace of the traditional Jensen-Shannon divergence in the plain GAN. This results in a better behaved discriminator gradient with respect to its input, thus facilitating the generator's optimisation. Then, under WGAN in particular, the generator G wishes to generate realistic fake data that can fool the discriminator D, *i.e.*, to maximise the discriminator's loss on fake data $\mathbb{E}[D(G(\boldsymbol{z}))]$; the discriminator D wishes to correctly distinguish the fake from real, thus to minimise its loss function on both real data $-\mathbb{E}[D(\boldsymbol{x})]$ and fake data $\mathbb{E}[D(G(\boldsymbol{z}))]$. As the discriminator needs to be 1-Lipschitz continuous for a stable training, gradient penalty [20] is further introduced to softly regularize the gradient norm. The objective function can be written as

$$\min_D \max_G V(G,D) = \mathbb{E}_{\tilde{x}\sim\mathcal{P}_{\tilde{x}}}[D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{x}}}[D(\boldsymbol{x})] + \lambda\mathbb{E}[(||\nabla D(\hat{\boldsymbol{x}})||_2 - 1)^2], \tag{3}$$

where $\mathbb{E}[(||\nabla D(\hat{\boldsymbol{x}})||_2 - 1)^2]$ would allow any gradient norms above one to be penalised, and λ is used to control the level of regularisation regarding the gradient penalty.

4. Methodology

4.1. Auxiliary Classifier in Private GANs

Our method builds on top of the recent developments of GANs, which have demonstrated their capabilities of generating outputs of high quality. Specifically, in our training process, the interactions between the discriminator D and the generator G are kept the same as in traditional GANs as discussed in Sec. 3. Differently, for privacy-preserving data generation, in addition to quality, it is also desirable for the outputs to be of high **utility** as measured by **gen2real** accuracy (g2r%), which is the accuracy on real data of a classifier that has been trained on generated data. However, to the best of our knowledge, no work so far has investigated that whether incorporating utility measure in the model design would result in better utility performance under the given privacy budget. Motivated by this, we propose a new private GAN method that blends the utility measure (*i.e.*, g2r%) into privacy-preserving GAN training. This is realised by introducing an auxiliary "gen2real" classifier C into the GAN training that specialises in classification tasks instead of discrimination tasks. The auxiliary classifier can then be used to mimic the utility evaluation process, where the classifier is firstly trained using fake data, then tested on real data. The learning objective of our auxiliary classifier is quite different from D, in that the interactions between G and C is now in collaboration instead of competition. They have the same goal of minimising the classifier's loss on fake data $-\mathbb{E}[C(G(\boldsymbol{z}, \boldsymbol{y}), \boldsymbol{y})]$, where $\boldsymbol{y}$ denotes the class label. Note that the generator G takes weighted average feedback from both source: D and C, where β is the proportion allocated to D and $1-\beta$ for C. Both weights are between 0 and 1. In summary, D, C and G play the following three-player minimax game with value function $V(G, D, C)$:

$$\begin{aligned}\min_G \max_D \min_C V(G, D, C) \\ = -\beta\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{P}_{\tilde{\boldsymbol{x}}}}[D(G(\boldsymbol{z}, \boldsymbol{y}))] + \beta\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{x}}}[D(\boldsymbol{x})] \\ - (1-\beta)\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{P}_{\tilde{\boldsymbol{x}}}}[C(G(\boldsymbol{z}, \boldsymbol{y}), \boldsymbol{y})]\end{aligned} \tag{4}$$

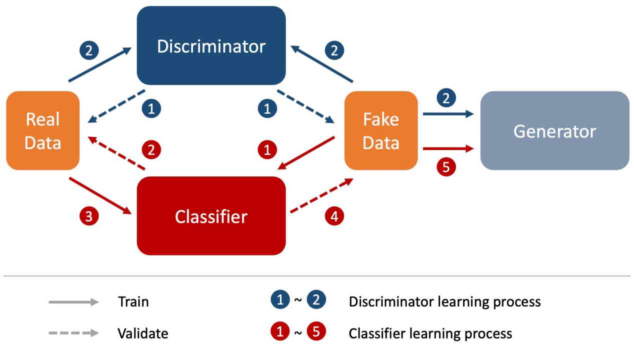

Figure 2. Architecture design. The discriminator receives feedback from both real and fake data simultaneously, while the classifier is trained in a sequential manner.

Remark on privacy-preservation. We adopt the powerful privacy protection technique: Differential Privacy (DP) in our GAN training. Recall in Eq. (1) from Sec. 3 that for any general training algorithm $f(\cdot)$, we can make it differentially private by using a satinitisation mechanism $\mathcal{M}$ that adds Gaussian noise $\mathcal{N}(0, (\sigma\Delta_2 f)^2)$ to the algorithm $f(\cdot)$, where the variance of the noise is determined by the function's sensitivity value $\Delta_2 f$. In our case, the function $f(\cdot)$ is the generator function $G(\cdot)$, since it is the generator that we wish it to achieve differential privacy. A common way to bound its sensitivity is by clipping its gradient $\boldsymbol{g}_G$ to have an $\mathcal{L}_2$-norm within a fixed clipping bound ζ. This would then allow us to analytically determine the distribution of the aforementioned added noise to be $\mathcal{N}(0, \sigma^2\zeta^2\boldsymbol{I}^2)$. To sum up, by applying the gradient sanitisation mechanism as follows would enable the generator to be differentially private:

$$\mathcal{M}_{\sigma,\zeta}(\boldsymbol{g}) = \text{Clip}(\boldsymbol{g}, \zeta) + \mathcal{N}(0, \sigma^2\zeta^2\boldsymbol{I}^2). \tag{5}$$

In addition, to minimise the loss on quality after the introduction of the sanitisation mechanism, we maximally preserve the gradient information by identifying all sub-components that can skip sanitisation. Firstly, we identify that D and C can be trained non-privately as Eq. (6) and Eq. (7), and eventually dropped since it is only the generator G that would be released after training. Secondly, according to the chain rule, the generator gradient can be decomposed into two sub-components: $\boldsymbol{g}_G = \nabla_{\boldsymbol{\theta}_G}\mathcal{L}_G(\boldsymbol{\theta}_G) = \nabla_{G(\boldsymbol{z};\boldsymbol{\theta}_G)}\mathcal{L}_G(\boldsymbol{\theta}_G) \cdot J_{\boldsymbol{\theta}_G}G(\boldsymbol{z};\boldsymbol{\theta}_G)$, where $J_{\boldsymbol{\theta}_G}G(\boldsymbol{z};\boldsymbol{\theta}_G)$ is the local generator jacobian computed independent of training data, hence can skip sanitisation as in Eq. (9). The other term $\nabla_{G(\boldsymbol{z};\boldsymbol{\theta}_G)}\mathcal{L}_G(\boldsymbol{\theta}_G)$ is the derivative of generator loss function with respect to its synthesised output. This is computed with the use of sensitive information since the generator's loss $\mathcal{L}_G(\boldsymbol{\theta}_G)$ is defined with regard to the discriminator or classifier, both of which is trained non-privately. Thus, to conclude, as shown in Eq. (9), this is the only sub-component that sanitisation mechanism $\mathcal{M}$ is applied to. This selective fashion of sanitisation allows reducing the number of parameters to be sanitised, and also training the non-sanitised components D and C more reliably.

$$\boldsymbol{\theta}_D := \boldsymbol{\theta}_D - \eta_D \cdot \boldsymbol{g}_D, \tag{6}$$

$$\boldsymbol{\theta}_C := \boldsymbol{\theta}_C - \eta_C \cdot \boldsymbol{g}_C. \tag{7}$$

$$\boldsymbol{\theta}_G := \boldsymbol{\theta}_G - \eta_G \cdot \tilde{\boldsymbol{g}}_G \tag{8}$$

$$\tilde{\boldsymbol{g}}_G = \mathcal{M}_{\sigma,\zeta}(\nabla_{G(\boldsymbol{z};\boldsymbol{\theta}_G)}\mathcal{L}_G(\boldsymbol{\theta}_G)) \cdot J_{\boldsymbol{\theta}_G}G(\boldsymbol{z};\boldsymbol{\theta}_G) \tag{9}$$

In above equations, η_G, η_D, η_C represent the learning rates for corresponding model components. Besides, gradient penalty is used for updating D and C to satisfy the 1-Lipschitz continuity condition for using W-loss. This also brings an additional benefit of precise estimation and analytical determination of the clipping bound ζ to be one thus saves the computational search.

Remark on evaluating the utility. Same as the evaluation process of g2r%, after training the classifier C using fake data, we freeze the classifier weights and validate it on real data. This validation result can be used to monitor the training progress of C, and facilitate decisions such as early stopping and hyper-parameter tuning. Finally, the classifier with optimized validation performance would be capable of providing the generator with higher quality feedback.

Remark on relevant GAN variants. A relevant GAN variant to our method is the auxiliary classifier GAN (AC-GAN) [32], which lets the discriminator to provide two separate feedback to the generator, a probability distribution over sources and a probability distribution over the class labels. This is still a two-player GAN, but the discriminator is multi-tasking. Although AC-GAN loosely shares some common concepts with our proposed method, they are very distinct in terms of motivation, model design and training process. Another loosely related line of work that uses classifier in GAN include Triple GAN [28] and Triangle GAN [18], both of which are designed for semi-supervised learning tasks, in that another generative model is designed to generate labels $\boldsymbol{y}$ using real data $\boldsymbol{x}$ to supplement the unsupervised features. Another 3-player GAN that includes a classifier is ALI [10], which leverages the classifier to improve the training of the discriminator by learning mappings from $\boldsymbol{y}$ to $\boldsymbol{x}$. In comparison, our method incorporates the classifier for completely different purposes, which are to improve on both standard and reversed utility for private data generations given a fixed privacy budget.

4.2. Real2Gen As a Reversed Utility

In addition to improving the generated outputs' standard utility, we identify that it is also beneficial to improve on their **reversed utility** as measured by real2gen accuracy (r2g%), *i.e.*, the accuracy on fake data of a classifier that has been trained on the real data. This aims for improving data transferability in both directions and also data generalisability. We have also discovered that combining this new direction of transferability in our model design would result in better private GANs.

This is realised by introducing an auxiliary "real2gen" classifier C into GAN training. As in Eq. (10), the learning objective for G and D are identical to the "gen2real" counterpart shown in Eq. (4). However, C now takes only real data as inputs for its updates hence minimises $-\mathbb{E}[C(\boldsymbol{x}, \boldsymbol{y})]$, to mimic the r2g% evaluation process, where it is trained on the real data, then tested on the generated data. In summary, the value function is as follows:

$$\begin{aligned} &\min_G \max_D \min_C V(G, D, C) \\ &\quad = -\beta\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{P}_{\tilde{\boldsymbol{x}}}}[D(G(\boldsymbol{z}, \boldsymbol{y}))] + \beta\mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{x}}}[D(\boldsymbol{x})] \\ &- (1-\beta)\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{P}_{\tilde{\boldsymbol{x}}}}[C'(G(\boldsymbol{z}, \boldsymbol{y}), \boldsymbol{y})] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{P}_{\boldsymbol{x}}}[C(\boldsymbol{x}, \boldsymbol{y})] \end{aligned} \tag{10}$$

Note that in the above equation, $C'(\cdot)$ directly copies the weight from C, and $C'(\cdot)$ would be detached from any gradient computations.

Algorithm 1 Differentially Private Generative Adversarial Network with Dual-Purpose Auxiliary Classifier (DP-GAN-DPAC) Training

```
Input: Dataset $\mathcal{D}$, subsampling rate $\gamma$, noise scale $\sigma$, training iterations $T$, learning rates $\eta_D$, $\eta_C$, $\eta_G$, the number of iterations per generator iteration for the discriminator $n_{dis}$, for the classifier using fake data and real data respectively $n_f$ and $n_r$, batch size $B$, class label $y$
Output: Private generator $\boldsymbol{\theta}_G$ and total privacy cost $\epsilon$
  Load non-private discriminators $\boldsymbol{\theta}_D^k$ for $k = 1, 2, ..., K$ ($K = 1/\gamma$); initialise private generator $\boldsymbol{\theta}_G$;
  for $step$ in $\{1, ..., T\}$ do
    Sample subset index $k \sim U[1, K]$ and subset $d_k$;
    Initialise classifier $\boldsymbol{\theta}_C^k$
    for $t$ in $\{1, ..., n_{dis}\}$ do
      Sample batch $\{\boldsymbol{x}_i\}_{i=1}^B \subseteq \mathcal{D}_k$;
      Sample batch $\{\boldsymbol{z}_i\}_{i=1}^B$ with $\boldsymbol{z}_i \sim \mathcal{P}_{\boldsymbol{z}}$;
      Compute mean discriminator gradient $\boldsymbol{g}_D$;
      $\boldsymbol{\theta}_D^k \leftarrow \boldsymbol{\theta}_D^k - \eta_D \cdot \boldsymbol{g}_D$
    end for
    for $t$ in $\{1, ..., n_f\}$ do
      Sample batch $\{\boldsymbol{z}_i\}_{i=1}^B$ with $\boldsymbol{z}_i \sim \mathcal{P}_{\boldsymbol{z}}$;
      Compute mean classifier gradient $\boldsymbol{g}_C$;
      $\boldsymbol{\theta}_C^k \leftarrow \boldsymbol{\theta}_C^k - \eta_C \cdot \boldsymbol{g}_C$
    end for
    for $t$ in $\{1, ..., n_r\}$ do
      Sample batch $\{\boldsymbol{x}_i\}_{i=1}^B \subseteq \mathcal{D}_k$;
      Compute mean classifier gradient $\boldsymbol{g}_{C'}$;
      $\boldsymbol{\theta}_C^k \leftarrow \boldsymbol{\theta}_C^k - \eta_C \cdot \boldsymbol{g}_{C'}$
    end for
    Compute mean satinised generator gradient $\tilde{\boldsymbol{g}}_G$;
    $\boldsymbol{\theta}_G \leftarrow \boldsymbol{\theta}_G - \eta_G \cdot \tilde{\boldsymbol{g}}_G$
    Accumulate privacy cost $\epsilon$;
  end for
  return Generator $\boldsymbol{\theta}_G$, privacy cost $\epsilon$
```

Remark on evaluating the reversed utility. Similarly, the real2gen classifier mimic the evaluation process of r2g%. After training the C on real data, it is then validated on the fake data. However, different from the design of gen2real classifier, this time we use the validation loss directly as the feedback for updating G, instead of a facilitator for monitoring purposes.

4.3. Training Strategies

Dual-purpose auxiliary classifier. For the auxiliary classifier, learning from the fake data mimics the evaluation process of standard utility by construction, making it a single-purpose classifier that improves on g2r%; while learning from the real data mimics the evaluation process of reversed utility, making it a single-purpose classifier that improves on r2g%. We further find that learning from both sources would enable us to feedback the generator with learning signals about the bilateral transferability (g2r% and r2g%) during training. This would further improve on its performance. The auxiliary classifier now kills two birds with one stone, hence named as "dual-purpose". However, this brings additional complexity to the training of the classifier, as the learning sources are from different data domains and distributions. Hence, we apply sequential training to the classifier, which would be discussed as follows.

Sequential training. Recall that D's task is to discriminate real from fake, thus it is necessary to batch the data in a way that corresponds each real data with its fake counterpart. As a result, D learns from both data domains simultaneously as in Fig. 2. In comparison, the task of C is instead to discriminate between classes, therefore the correspondence should be shift from real-and-fake labels to class labels. Moreover, we find that for classification task, mixing the losses from real and fake sources into the same equation would result in noisy gradients that confuse C during its updates. This is because real and fake data are of different distributions and have quite distinct features for each class label. Hence, we train the classifier sequentially and for two separate updates, by alternatively learning from fake data then real data (instead of simultaneously). This also enables C to be designed in a dual-purpose way (Fig. 2) that incorporates the evaluation of both g2r% and r2g% into the training process, which guarantees the improvements on both measures. Intuitively, this auxiliary classifier is essentially acting as the intermediary facilitating the communication between G and the real data, telling the generator: "I was entirely trained by you (step 1), I then go and see what real data look like (step 2) and have got slighted improved (step 3), now I am back and show you what I am like now (step 4), to let you know how you should have trained me back then if you wish to let me perform well on the real data that you can never see (step 5)." The detailed algorithm is shown in Algorithm 1. Implementational details are discussed in Sec. 5.1.

5. Experiments

5.1. Experimental setup

We compare our generated data with several state-of-the-art differentially private generative model baselines on three image datasets over quality and utility evaluation metrics.

Datasets. We conducted experiments on image datasets to demonstrate the superiority of our method generating high dimensional differentially private data. MNIST [26] and FashionMNIST (F-MNIST) [39] have been the standard datasets for this line of work, however, they are both grayscale images. We have also experimented on even higher dimensional celebrity face image dataset CelebA, to validate the applicability of our method on image datasets with RGB colour channels. Specifically, MNIST and F-

MNIST datasets both contains 60000 training examples and 10000 validation examples of 28 × 28 grey-scale images consist of 10 labels. The CelebA dataset contains 202599 colour images of celebrity faces, each with 40 attribute annotations, among which we take the binary "gender" attribute as the label. We used the official pre-processed version and further resized the images to 32 × 32 × 3. The data is partitioned into three subset: 162770 training, 19867 validation, and 19962 test. For all datasets, we use only the training set for training purposes, and use the validation set for evaluating our conditional GAN.

Evaluation metrics. To evaluate output quality, we use Inception Score (IS) and Frechet Inception Distance (FID), we have also included the generated images as Fig. 3 and Fig. 4 for visual inspections. To evaluate utility, the standard is to use the g2r%, which computes the accuracy on real data of a downstream classifier trained using generated data, where for the classifier, we used Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). However, we find g2r% only considers the transferability from fake data to real data, while neglecting the other way around. This gives a readily exploitable loophole for shortcut outputs to trick the metric. Take gender classification on a hypothetical face image dataset as an example, if in which 90% females are white, and 90% males are black, we can easily trick the g2r% metric by generating only purely black images for male, and purely white images for female. Then, the learned classifier from our fake data would overfit the only feature it could extract - the degree of blackness, and still get 90% accuracy on the real data. This overfitting issue arises more frequently when the available real data for training are very limited, which is exactly the venue where this line of work comes in - remember that we wish to generate data for domains that have sensitive and **limited** data. To prevent overfitting and generate high-performing classifiers that are not easily being tricked, ideally, the generated images should be more generalisable to the true features that distinguish between semantic classes of male and female: *e.g.*, female faces should have smaller size, larger cheeks, and smaller and less prominent brows, noses, and chins. Therefore, we propose to also measure the "reversed utility" that computes the accuracy on fake data of a downstream classifier trained using real data, dubbed as real2gen accuracy (r2g%). This other direction of transferability from real to fake allows the evaluation of output generalisability, as intuitively, for a fake output to generalise well, it should be difficult to tell from its real class by a real-world classifier. In our evaluations, MLP and CNN are selected as the classifiers. We recommend future works in this domain to consider both g2r% and r2g% as utility metrics.

Implementational details. For the generator and discriminator, we use DCGAN for the discriminator and classifier, and ResNet (adapted from BigGAN) for the generator. Same as previous works, we have also used the subsampling technique for enhancing privacy protection, and pre-trained the discriminators before training. This brings better discriminator convergence without costing any additional privacy budgets since it is only the generator that we would release after training. Our models are implemented in PyTorch. We implement the gradient sanitisation mechanism by registering a backward hook to the selected portion of generator gradient $\nabla_{G(\boldsymbol{z})}\mathcal{L}_G(\theta_G)$ that we wish to sanitise. We adopt the official implementation of [38] for accumulating the privacy costs after each generator iteration, where *the smaller the* σ, *the larger the privacy cost* ϵ. The value choice of $\delta = 1e-5$, $\sigma = 1.07$ were both kept the same as in [8]. These correspond to line 23 of Algorithm 1. After 20000 iterations, the total privacy costs are within the budget of $\epsilon = 10$.

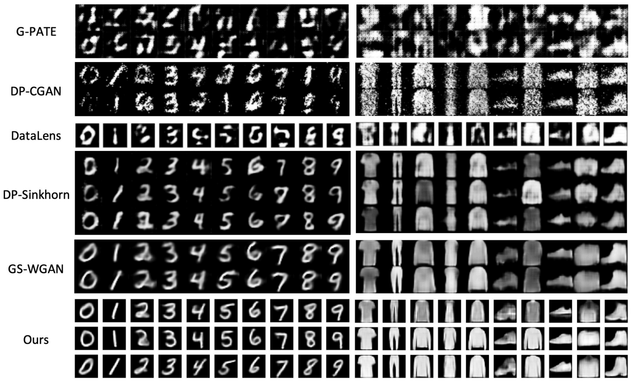

Figure 3. Image generated at privacy budget $\epsilon = 10$ for MNIST (Left) and F-MNIST (Right) by various methods.

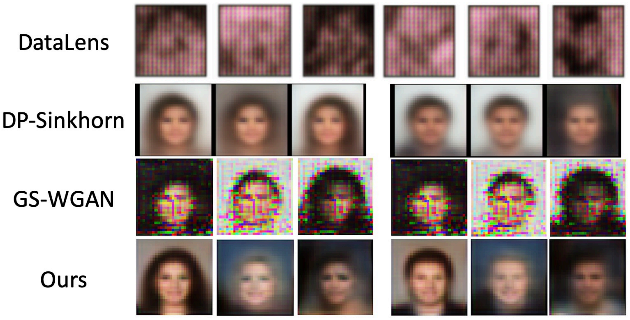

Figure 4. Image generated at privacy budget $\epsilon = 10$ for CelebA by various methods conditioned on gender. Left: Female. Right: Male.

5.2. Comparison with baselines

Quality. Qualitatively, as shown in Fig. 3 and Fig. 4, our method produced the most visually appealing results on all tested datasets compared to several state-of-the-art baselines. On MNIST and F-MNIST, our method successfully generates privacy-preserving images that are hard to

	MNIST		F-MNIST		CelebA	
Method	IS	FID	IS	FID	IS	FID
PATE-GAN [25]	1.46	253.55	2.35	229.25	-	-
DP-CGAN [36]	-	179.20	-	243.80	-	-
G-PATE [30]	5.16	150.62	4.33	171.90	1.37	350.92
DataLens [37]	5.78	137.50	4.58	167.70	1.42	320.84
DP-MERF [22]	-	121.40	-	110.40	-	-
GS-WGAN [8]	9.23	61.34	5.32	131.34	1.85	297.35
DPSinkhorn [6]	-	55.56	-	129.40	-	168.40
Ours	**9.71**	**54.06**	**6.60**	**90.77**	**1.90**	**139.99**

Table 1. Comparing IS ↑ and FID ↓ on various datasets.

	MNIST		F-MNIST		CelebA	
Method	MLP	CNN	MLP	CNN	MLP	CNN
DP-CGAN [36]	0.60	0.63	0.50	0.46	-	-
G-PATE [30]	-	0.81	-	0.69	-	0.71
DataLens [37]	-	0.81	-	0.71	-	0.73
DP-MERF [22]	0.81	0.82	0.71	**0.73**	-	-
GS-WGAN [8]	0.79	0.80	0.65	0.65	0.68	0.66
DPSinkhorn [6]	0.80	0.83	0.73	0.71	0.76	0.76
Ours	**0.85**	**0.88**	**0.75**	**0.73**	**0.80**	**0.85**

Table 2. Comparing gen2real accuracy ↑ on various datasets.

	MNIST		F-MNIST		CelebA	
Method ↑	MLP	CNN	MLP	CNN	MLP	CNN
GS-WGAN [8]	0.99	0.99	0.85	0.85	0.66	0.60
Ours	**1.00**	**1.00**	**0.97**	**0.98**	**0.99**	**0.98**

Table 3. Comparing real2gen accuracy ↑ on various datasets.

tell from the real ones. On the more challenging CelebA dataset, DataLens [37] can hardly generate meaningful outputs, GS-WGAN can resemble faces, but with a lot of masaic on the faces and hard to visualise gender differences. DP-Sinkhorn [6] can show clear signs of gender, however, the outputs are much blurrier compared to ours. Quantitatively, as in Tab. 1, our method results in the best-performing IS and FID on all tested datasets. The advantage is much more distinct for the more challenging F-MNIST and CelebA datasets. These demonstrate that our deliberate training design of sequential training of the auxiliary classifier has resulted in improved training stability and better output quality.

Utility. Results in Tab. 2 and Tab. 3 shows that on all three datasets, our method consistently improve on baselines no matter we choose MLP or CNN as the classifier. Specifically, the improvements on GS-WGAN over both utility metrics are very significant, which clearly illustrate the effectiveness of our dual-purpose design of auxiliary classifier on realising both of its purposes.

More experimental results and analyses that demonstrate our method's better privacy-quality and privacy-utility trade-offs are provided in the supplementary material.

			gen2real ↑		real2gen ↑	
Method	IS ↑	FID ↓	MLP	CNN	MLP	CNN
Baseline	5.32	131.24	0.65	0.65	0.85	0.85
w/o g2r	6.33	88.17	0.73	0.68	0.94	0.95
w/o r2g	6.47	**86.91**	0.74	0.71	0.92	0.92
w/o seq	4.91	128.25	0.65	0.64	0.88	0.77
w/o init	6.56	101.69	0.72	0.65	**0.97**	0.95
Full	**6.60**	90.77	**0.75**	**0.73**	**0.97**	**0.98**

Table 4. Ablation studies.

5.3. Ablation studies

(1) Single-purpose vs. dual-purpose auxiliary classifier. With the two purposes of improving on standard and reversed utility, we incorporate both gen2real and real2gen components into the algorithm design, by introducing a dual-purpose auxiliary classifier. We experiment on removing each purpose from the design, and compare their performances with the dual-purpose version, and also the baseline method that does not use the classifier. As in Tab. 4, results show that in terms of g2r% and r2g% performances, the single-purpose versions (*i.e.*, without gen2real or real2gen component) are inferior to the dual-purpose one, but much superior to the no-purpose baseline. Additionally, the one without real2gen has worse r2g% performance, but better g2r% performance compared to the one without gen2real. These all prove the effectiveness of our design. The study also shows that under our deliberate training design, the single-purpose and dual-purpose versions would have comparable output quality, with the dual-purpose one having slightly better IS, but slightly worse FID. This is because the variate in number of purposes does not change the key mechanism of our training design that contributes to quality improvement, which is the separation of real and fake inputs in the training of C. The dual-purpose version benefits from our sequential training design, while the single-purpose ones would only use either real or fake data as the input. Thus, they have all avoided inputting the real and fake data simultaneously. This study also shows the auxiliary classifier's consistent improvement on quality over the baseline by a large margin, regardless of the change in number of purposes.

(2) Sequential vs. parallel training of auxiliary classifier. We design the classifier to learn from fake and real inputs separately and sequentially. What if we did not go the extra mile and simply uses the conventional training scheme of GANs for updating the discriminator where it learns from real and fake data simultaneously? We then modify the loss equation for updating C to be the simple average of its loss from real and fake inputs, and the results in Tab. 4 show a significant downgrade compared to the sequential opponent in all aspects. This proves the contribution of the sequential training strategy for classifier in preventing noisy gradients and improving training stability.

	Hyperparameters					gen2real ↑		real2gen ↑	
	t_c	β	$i_f : i_r$	IS ↑	FID ↓	MLP	CNN	MLP	CNN
Baseline	-	1.0	-	5.32	131.24	0.65	0.65	0.85	0.85
Different t_c	0	0.8	10:10	6.09	93.27	**0.75**	0.71	**0.97**	0.92
	16000	0.8	10:10	6.12	105.35	0.68	0.65	0.96	0.92
Different β	6000	0.5	10:10	6.55	105.11	0.69	0.65	0.95	0.97
	6000	0.95	10:10	6.36	**90.43**	0.74	0.70	**0.97**	**0.98**
Different $n_f : n_r$	6000	0.8	60:10	6.26	106.50	0.70	0.66	0.95	0.92
	6000	0.8	10:60	6.49	90.93	0.73	0.70	**0.97**	0.96
Ours	6000	0.8	10:10	**6.60**	90.77	**0.75**	**0.73**	**0.97**	**0.98**

Table 5. Hyperparameter analyses on F-MNIST. t_c refers the generator iteration that we introduce the auxiliary classifier; β is the weights given to the discriminator compared to the classifier in the loss function; $i_f : i_r$ means the number of classifier iteration per generator iteration using fake data and real data respectively. Our proposed method is relatively insensitive to the hyperparameter selection.

(3) The re-initialisation of classifier during each generator iteration. The auxiliary classifier is re-initialised after each generator iteration, to closely mimic the gen2real evaluation process, where a classifier is trained from scratch using the fake data, before testing on real data. Thus, this strategy aims to improve on g2r% performance. What if we save and re-use the classifier output after every generator iteration? The results show that if the strategy is removed, r2g% would have comparable performance, the quality measures would have slightly worse performance, and the g2r% would have a distinct downgrade in performance, especially the g2r% using CNN would decrease from 73% to 65%. These have demonstrated the effectiveness of the re-initilisation design of classifier on improving output performance, especially g2r%. It indeed allows better feedback to be passed to the generator during training, as the strategy makes the auxiliary classifier more representative of the generator during each specific iteration.

5.4. Hyper-parameter analyses

We have extensively tuned hyper-parameters that relates to the proposed auxiliary classifier: the iteration t_c to introduce auxiliary classifier into the training pipeline, the proportion (β) of generator's feedback allocated to D as opposed to C, and the number of classifier iterations n_f and n_r per generator update using fake data and real data respectively. In Tab. 5, we present two representative examples for each hyper-parameter to discuss the general patterns.

We find late-launching the auxiliary classifier has better performance compared to introducing it into the pipeline from start ($t_c = 0$). During the 20000 total iterations of generator, the best results are achieved when the auxiliary classifier comes into effect after $t_c = 6000$ iterations for MNIST and F-MNIST, and $t_c = 1000$ iterations for CelebA. This is because the generator finds challenging to generate meaningful outputs during initialisations. Thus, the auxiliary classifier trained using the generated output is of less quality, and in return gives lower-quality feedback to the generator during the first hundreds of training iterations. Also, as the clipping step in gradient sanitisation scheme is constraining the amount of gradient flow to the generator during each iteration, the convergence would be faster if all budgets were allocated to discriminator during initial stages. We also find the performance is insensitive to t_c when it takes values within the range of $500 - 8000$. However, setting t_c too large (*e.g.*,$t_c = 16000$ as in Tab. 5) would prevent the classifier from being fully leveraged and result in worse performance compared to smaller settings of t_c, but better performance compared to the baseline that does not incorporate the classifier. Also, the performance is not very sensitive to β, i_f, and i_r, comparable performances with the best setting were achieved after testing on a wide range of values.

6. Conclusion

In this paper, we identify the "reversed utility" as a crucial supplement of the standard utility to better evaluate output generalisability. We also find both utility capable of improving on transferability between real and fake data if they are incorporated into our model design. To implement this unification, we propose a novel and effective differentially private GAN with dual-purpose auxiliary classifier (DP-GAN-DPAC). Align with our deliberate sequential training strategies, the method is also entitled to train more stably, hence generate higher quality outputs. Extensive experiments show that DP-GAN-DPAC significantly outperforms the current state-of-the-art baselines on both greyscale and RGB image datasets. We conclude that incorporating a dual-purpose auxiliary classifier into private GAN trainings could become a standard for this line of work.

7. Acknowledgments

This work was supported in part by the Australian Research Council under Project DP210101859 and the University of Sydney Research Accelerator (SOAR) Prize. The AI training platform supporting this work were provided by High-Flyer AI. (Hangzhou High-Flyer AI Fundamental Research Co., Ltd.)

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 2, 3

[2] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *ICIP*, pages 2089–2093, 2017. 1

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017. 1, 3

[4] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019. 2

[5] Christopher Bowles, Roger Gunn, Alexander Hammers, and Daniel Rueckert. GANsfer learning: Combining labelled and unlabelled data for GAN based data augmentation. *arXiv preprint arXiv:1811.10669*, 2018. 1

[6] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with Sinkhorn divergence. *NeurIPS*, pages 12480–12492, 2021. 2, 7

[7] Dongjie Chen, Sen-ching Samson Cheung, Chen-Nee Chuah, and Sally Ozonoff. Differentially private generative adversarial networks with model inversion. In *2021 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2021. 3

[8] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. GS-WGAN: A gradient-sanitized approach for learning differentially private generators. *NeurIPS*, pages 12673–12684, 2020. 1, 2, 3, 6, 7

[9] Victoria Cheng, Vinith M Suriyakumar, Natalie Dullerud, Shalmali Joshi, and Marzyeh Ghassemi. Can you fake it until you make it? Impacts of differentially private synthetic data on downstream classification fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 149–160, 2021. 3

[10] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *arXiv*, 2016. 4

[11] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009. 3

[12] 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. 1, 2

[13] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, 2010. 3

[14] Liyue Fan. A survey of differentially private generative adversarial networks. In *The AAAI Workshop on Privacy-Preserving Artificial Intelligence*, page 8, 2020. 1

[15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 1

[16] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, 2018. 1

[17] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 151–164. Springer, 2019. 2

[18] Zhe Gan, Liqun Chen, Weiyao Wang, Yunchen Pu, Yizhe Zhang, Hao Liu, Chunyuan Li, and Lawrence Carin. Triangle generative adversarial networks. In *NeurIPS*, 2017. 4

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 3

[20] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 3

[21] Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *CVPR*, pages 8385–8393, 2020. 1

[22] Frederik Harder, Kamil Adamczewski, and Mijung Park. DP-MERF: Differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*, pages 1819–1827, 2021. 7

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

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 1

[25] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2018. 1, 2, 7

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[27] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017. 1

[28] Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *NeurIPS*, 2017. 4

[29] Yuezun Li and Siwei Lyu. De-identification without losing faces. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pages 83–88, 2019. 1

[30] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-PATE: Scalable differentially private data generator via private aggregation of teacher discriminators. *NeurIPS*, pages 2965–2977, 2021. 1, 2, 7

[31] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275, 2017. 2, 3

[32] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *ICML*, pages 2642–2651, 2017. 4

[33] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016. 2

[34] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific reports*, 9(1):1–9, 2019. 1

[35] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18, 2017. 1

[36] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-CGAN: Differentially private synthetic data and label generation. In *CVPR Workshops*, 2019. 2, 3, 7

[37] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. DataLens: Scalable privacy preserving training via gradient compression and aggregation. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2146–2168, 2021. 1, 2, 7

[38] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled renyi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. 6

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[40] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018. 1, 2

[41] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. GANobfuscator: Mitigating information leakage under GAN via differential privacy. *IEEE Transactions on Information Forensics and Security*, 14(9):2358–2371, 2019. 2

[42] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1

[43] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model (technical report). *arXiv preprint arXiv:1801.01594*, 2018. 2