PFGUARD: A GENERATIVE FRAMEWORK WITH PRIVACY AND FAIRNESS SAFEGUARDS

Soyeon Kim, Yuji Roh, Geon Heo, and Steven Euijong Whang
KAIST, {purplehibird,yuji.roh,geon.heo,swhang}@kaist.ac.kr

ABSTRACT

Generative models must ensure both privacy and fairness for Trustworthy AI. While these goals have been pursued separately, recent studies propose to combine existing privacy and fairness techniques to achieve both goals. However, naïvely combining these techniques can be insufficient due to privacy-fairness conflicts, where a sample in a minority group may be amplified for fairness, only to be suppressed for privacy. We demonstrate how these conflicts lead to adverse effects, such as privacy violations and unexpected fairness-utility tradeoffs. To mitigate these risks, we propose PFGuard, a generative framework with privacy and fairness safeguards, which simultaneously addresses privacy, fairness, and utility. By using an ensemble of multiple teacher models, PFGuard balances privacy-fairness conflicts between fair and private training stages and achieves high utility based on ensemble learning. Extensive experiments show that PFGuard successfully generates synthetic data on high-dimensional data while providing both DP guarantees and convergence in fair generative modeling – the first of its kind to our knowledge.

1 INTRODUCTION

Recently, generative models have shown remarkable performance in various applications including vision (Wang et al., 2021b) and language tasks (Brown et al., 2020) – while also raising significant ethical concerns. In particular, *privacy* and *fairness* concerns have emerged due to generative models mimicking their training data. On the privacy side, *specific training data can be memorized*, allowing the leakage of personal sensitive information (Hilprecht et al., 2019; Sun et al., 2021). On the fairness side, *any bias in the training data can be learned*, resulting in biased synthetic data and unfair downstream performances across demographic groups (Zhao et al., 2018; Tan et al., 2020).

Although privacy and fairness are both essential for generative models, previous research has primarily tackled them separately. Differential Privacy (DP) techniques (Dwork et al., 2014), which provide rigorous privacy guarantees, have been developed for *private generative models* (Xie et al., 2018; Jordon et al., 2018); various fair training techniques, which remove data bias and generate more balanced synthetic data, have been proposed for *fair generative models* (Xu et al., 2018; Choi et al., 2020). To achieve both objectives, harnessing these techniques has emerged as a promising direction. For example, Xu et al. (2021) combine a fair pre-processing technique (Celis et al., 2020) with a private generative model (Chanyaswad et al., 2019) to train both fair and private generative models.

However, we contend that naïvely combining developed techniques for privacy and fairness can lead to a *worse privacy-fairness-utility tradeoff*, where utility is a model's ability to generate realistic synthetic data. We first illustrate how *privacy and fairness can conflict* in Fig. 1. Given the data samples M_1, M_2, M_3, and m_1 where M and m denote the majority and minority data groups, respectively, DP and fairness techniques play a tug-of-war regarding the use of minority data point m_1; DP techniques *limit* its use to prevent pri-

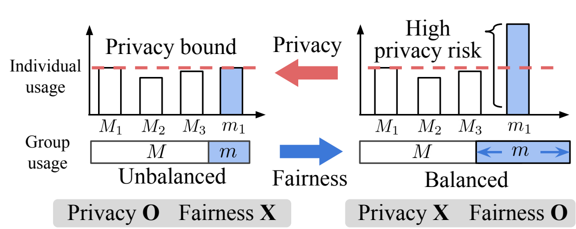

Figure 1: Privacy-fairness conflict. Privacy techniques prefer the left-hand scenario to prevent privacy risk of a certain data sample, while fairness techniques prefer the right-hand scenario to balance learning w.r.t. groups. Related empirical results are shown in Table 3.

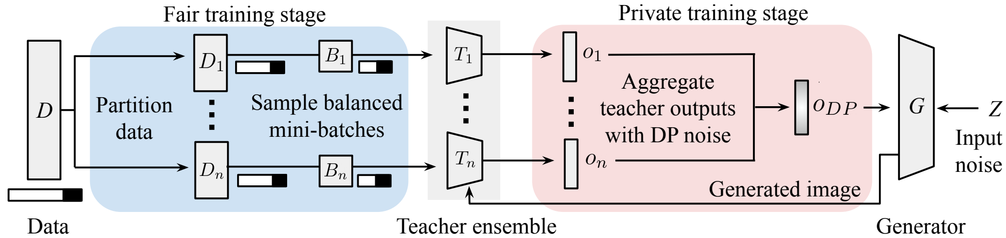

Figure 2: Overview of PFGuard. PFGuard integrates fairness and privacy in generative models through a two-stage process. In the fair training stage (blue), we train fair teacher models by sampling balanced mini-batches from biased data subsets, disjointly partitioned from the original dataset. In the private training stage (red), we aggregate the teacher outputs to supervise the generator with DP noise to ensure privacy. With these training stages to achieve fairness and privacy, PFGuard also achieves high utility based on ensemble learning of teacher models, resulting in high-quality, unbiased, and private synthetic data. More details on the framework including DP guarantee and the training algorithm are in Sec. C.

vacy risks such as memorization, while fairness techniques *increase* its use to promote more balanced learning w.r.t. groups given the biased data. As a result, fairness techniques may undermine privacy by overusing m_1, while DP techniques may undermine fairness by limiting m_1's usage. Moreover, combining different techniques can *introduce new technical constraints*, reducing the effectiveness of original methods. For instance, the fair preprocessing technique used by Xu et al. (2021) hinders the utility of the DP generative model by requiring data binarization, which incurs significant information loss on high-dimensional data such as images – restricting their overall framework only applicable to low-dimensional structural data.

Therefore, we design a generative framework that *simultaneously* addresses fairness and privacy while achieving utility for high-dimensional synthetic data generation. To this end, we propose PFGuard: a generative framework with **P**rivacy and **F**airness Safe**guard**s. As illustrated in Fig. 2, the key component is an *ensemble of intermediate teacher models*, which balances privacy-fairness conflicts between fair training and private training stages. In the *fair training* stage, we design a new sampling technique to train fair teachers, which provides a theoretical convergence guarantee to the fair generative modeling. In the *private training* stage, we employ the Private Teacher Ensemble Learning (PTEL) approach (Papernot et al., 2016; 2018), which aggregates each teacher's knowledge with random DP noise (e.g., noisy voting), to privatize the knowledge transfer to the generator. As a result, PFGuard provides a unified solution to train both fair and private generative models by transferring the teachers' fair knowledge in a privacy-preserving manner.

Compared to simple sequential approaches, PFGuard is carefully designed to address privacy-fairness conflicts. Recall that fairness techniques can incur privacy breaches by overusing minority data; in contrast, PFGuard *prevents privacy breaches* by decoupling fairness and privacy with intermediate teacher models. Although fair sampling can still compromise privacy in teacher models by potentially overusing minority data, PFGuard ensures privacy in the generator – our target model – by training it solely with the privatized teacher output, as shown in Fig. 2. Also, recall that privacy techniques can lead to fairness cancellation by suppressing the use of minority data; in contrast, PFGuard *avoids fairness cancellation* through teacher-level privacy bounding using PTEL approaches. Compared to sample-level privacy bounding methods like gradient clipping (Abadi et al., 2016), teacher-level bounding leaves room for teachers to effectively learn balanced knowledge via fair training. As a result, PFGuard provides strict DP guarantees for the generator and better preserves fairness compared to the combination of fairness-only and privacy-only techniques – see more analyses in Sec. 3.

Moreover, PFGuard is compatible with a wide range of existing private generative models and preserves their utility. PTEL is widely adopted in private generative models as it provides prominent privacy-utility tradeoff (Jordon et al., 2018; Chen et al., 2020; Long et al., 2021; Wang et al., 2021a). PFGuard can extend any of these models with a fair training stage as shown in Fig. 2, which requires a *simple modification in the minibatch sampling process*. Since additional fair sampling can be advantageous in maintaining optimization complexity compared to say adding a loss term for fairness, PFGuard preserves the privacy-utility tradeoff of PTEL as well while improving fairness. We also provide guidelines to control the fairness-privacy-utility tradeoff – see more details in Sec. 4.

Experiments show that PFGuard successfully generates high-dimensional synthetic data while ensuring both privacy and fairness; to our knowledge, PFGuard is the *first* framework that works on high-dimensional data including images. Our results also reveal two key findings: (1) existing private generative models can produce highly-biased synthetic data in real-world scenarios even with simple bias settings, and (2) a naïve combination of individual techniques may fail to achieve either privacy or fairness even with simple datasets – highlighting PFGuard's effectiveness and the need for a better integration of fair and private generative models.

Summary of Contributions **1)** We identify how privacy and fairness conflict with each other, which complicates the development of responsible generative models. **2)** We propose PFGuard, which is to our knowledge the first generative framework that supports privacy and fairness for high-dimensional data. **3)** Through extensive experiments, we show the value of integrated solutions to address the privacy-fairness-utility-tradeoff compared to simple combinations of individual techniques.

2 PRELIMINARIES

Generative Models We focus on Generative Adversarial Networks (Goodfellow et al., 2014), which are widely-used generative models that leverage adversarial training of two networks to generate realistic synthetic data: 1) a generator that learns the underlying training data distribution and generates new samples and 2) a discriminator that distinguishes between real and generated data. The discriminator can be considered as the teacher model of the generator, as the generator does not have access to the real data and only learns from the discriminator via the GAN loss function.

Differential Privacy To privatize generative models, we use Differential Privacy (DP) (Dwork et al., 2014), a gold standard privacy framework that provides quantified privacy analysis. DP measures how much an adversary can infer about one data sample based on differences in two outputs of an algorithm, using two *adjacent datasets* that differs by one sample. Two parameters (ε, δ) are used:

Definition 2.1. ((ε, δ)-Differential Privacy (Dwork et al., 2006a)) A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two adjacent datasets $\mathcal{D}, \mathcal{D}'$ and for any subset of outputs $\mathcal{O} \subseteq \mathcal{R}$, the following holds:

$$\Pr(\mathcal{M}(\mathcal{D}) \in \mathcal{O} \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathcal{D}') \in \mathcal{O}) + \delta,$$

where ε is the upper bound of privacy loss, and δ is the probability of breaching DP constraints.

We can enforce DP in an algorithm in two steps (Dwork et al., 2014). Given an algorithm f to ensure DP and a dataset $\mathcal{D}$, we first bound *sensitivity* (Def. 2.2), which captures the maximum influence of a single data sample on the output of f. We then add random noise with a scale proportional to the sensitivity value. A common way to ensure DP is to use a Gaussian mechanism (Dwork et al., 2014) (Thm. 2.1), which utilizes Gaussian random noise with a scale proportional to l_2-sensitivity.

Definition 2.2. (Sensitivity (Dwork et al., 2014)) The l_p-sensitivity for a d-dimensional function $f : X \to \mathbb{R}^d$ is defined as $\Delta_f^p = \max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_p$ over all adjacent datasets $\mathcal{D}, \mathcal{D}'$.

Theorem 2.1. (Gaussian mechanism (Dwork et al., 2014; Mironov, 2017)) Let $f : X \to \mathbb{R}^d$ be an arbitrary d-dimensional function with l_2-sensitivity Δ_f^2. The Gaussian mechanism $\mathcal{M}_\sigma$, parameterized by σ, adds Gaussian noise into the output, i.e., $\mathcal{M}_\sigma(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, and satisfies (ε, δ)-DP for $\sigma \geq \sqrt{2\ln(1.25/\delta)}\Delta_f^2/\varepsilon$.

Fairness We consider a generative model to be fair if two criteria are satisfied: 1) the model generates similar amounts of data for different demographic groups with similar quality, and 2) the generated data can be used to train a fair downstream model w.r.t. traditional group fairness measures. For 1), we measure the size and image quality disparities between the groups using the Fréchet Inception Distance (FID) score (Heusel et al., 2017; Choi et al., 2020) to assess image quality. For 2), we use two prominent group fairness measures: equalized odds (Hardt et al., 2016) where the groups should have the same label-wise accuracies; and demographic parity (Feldman et al., 2015) where the groups should have similar positive prediction rates.

3 CHALLENGES OF SATISFYING BOTH PRIVACY AND FAIRNESS

In this section, we examine the practical challenges of integrating privacy-only and fairness-only techniques to train both private and fair generative models. Based on Fig. 1's intuition on how privacy and fairness conflict, we analyze how existing approaches for DP generative models and fair generative models can technically conflict with each other using standard DP and fairness techniques: DP-SGD (Abadi et al., 2016) and reweighting (Choi et al., 2020). Let $\boldsymbol{g}(\mathbf{x})$ denote the gradient of the data sample $\mathbf{x}$, and p_{bal} and p_{bias} denote balanced and biased data distributions, respectively.

- *DP-SGD* is a standard DP technique Chen et al. (2020) for converting non-DP algorithms to DP algorithms by modifying traditional stochastic gradient descent (SGD). Compared to SGD, DP-SGD 1) applies *gradient clipping* to limit the individual data point's contribution, where $\boldsymbol{g}(\mathbf{x})$ is clipped to $\boldsymbol{g}(\mathbf{x})/\max(1, ||\boldsymbol{g}(\mathbf{x})||_2/C)$ (the sensitivity becomes the clipping threshold C), and 2) uses a *Gaussian mechanism* (Thm. 2.1) to add sufficient noise to ensure DP.
- *Reweighting* is a traditional fairness method (Horvitz & Thompson, 1952) widely used in generative modeling (Choi et al., 2020; Kim et al., 2024), which assigns *greater weights to minority groups* for a "balanced" loss during SGD. In particular, setting the sample weight to the likelihood ratio $h(\mathbf{x}_i){=}p_{\text{bal}}(\mathbf{x}_i)/p_{\text{bias}}(\mathbf{x}_i)$ produces an unbiased estimate of $\mathbb{E}_{\mathbf{x}\sim p_{\text{bal}}}[\boldsymbol{g}(\mathbf{x})]$ as follows:

$$\mathbb{E}_{\mathbf{x}\sim p_{\text{bias}}}[\boldsymbol{g}(\mathbf{x})\cdot h(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim p_{\text{bias}}}\left[\boldsymbol{g}(\mathbf{x})\frac{p_{\text{bal}}(\mathbf{x})}{p_{\text{bias}}(\mathbf{x})}\right] = \mathbb{E}_{\mathbf{x}\sim p_{\text{bal}}}[\boldsymbol{g}(\mathbf{x})]. \tag{1}$$

Adding Fairness Can Worsen Privacy Ensuring fairness in DP generative models can significantly *increase sensitivity* (Def. 2.2), leading to invalid DP guarantees. Sensitivity, which measures a data sample's maximum impact on an algorithm, is crucial in DP generative models because the noise amount required for DP guarantees is often proportional to this sensitivity value. However, integrating fairness techniques in DP generative models can invalidate their sensitivity analyses by adjusting model outputs for fairness purposes. One example is the aforemetioned reweighting technique. While the reweighting technique amplifies the impact of certain data samples to balance model training across groups (Choi et al., 2020), performing reweighting after DP-SGD can incur privacy breaches by amplifying sample gradients beyond C, invalidating the sensitivity derived from gradient clipping. Other examples include directly feeding data attributes such as class labels or sensitive attributes (e.g., race, gender) to a generator for more balanced synthetic data (Xu et al., 2018; Sattigeri et al., 2019; Yu et al., 2020), which can cause large fluctuations in the generator output and similarly end up increasing sensitivity. This increased sensitivity by fairness techniques require more noise to maintain the same privacy level, compromising the original DP guarantees unless modifying DP techniques to add more noise. However, this modification is also not straightforward as assessing the increased sensitivity by fairness techniques can be challenging (Tran et al., 2021b).

Adding Privacy Can Worsen the Fairness-Utility Tradeoff Another direction is to ensure *privacy* in fair generative models, but *configuring an appropriate privacy bound can be challenging*, leading to unexpected fairness-utility tradeoffs.

We can extend reweighting for fairness to also satisfy DP using DP-SGD, but finding the clipping threshold C that balances fairness and utility can be challenging. Note that training reweighting-based fair generative models with DP-SGD provides valid DP guarantees. However, the clipping now undoes the fairness adjustments where reweighted gradients $\boldsymbol{g}(\mathbf{x})\cdot h(\mathbf{x})$ are clipped to $\boldsymbol{g}(\mathbf{x})\cdot h(\mathbf{x})/\max(1, ||\boldsymbol{g}(\mathbf{x})\cdot h(\mathbf{x})||_2/C)$, and Eq. 1 does not hold if $C \leq \boldsymbol{g}(\mathbf{x})$. Here one solution is to use a larger C such that $C \geq \boldsymbol{g}(\mathbf{x})$. However, increasing C also increases the noise required for DP, which reduces utility (Fig. 3). As a result, selecting a C that balances fairness and utility may necessitate extensive hyperparameter tuning (Bu et al., 2024), complicating the systematic integration of fairness into DP generative models.

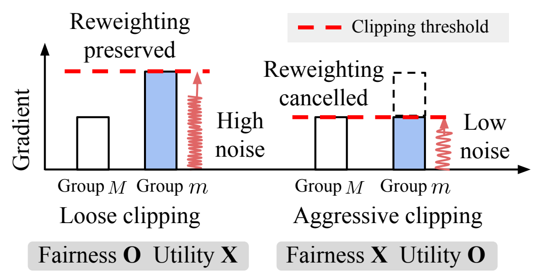

Figure 3: Fairness-utility tradeoff caused by DP-SGD when used on top of reweighting. Depending on the choice of C, DP-SGD may compromise utility (left) or fairness (right).

Overall, we show that a naïve combination of existing fairness-only and privacy-only techniques can be *insufficient* to achieve both objectives. While we have not exhaustively covered all possible combinations, one can see how privacy breaches and unexpected fairness-utility tradeoffs can easily occur without a careful design. To avoid these downsides, we emphasize the need for a *unified design* that integrates both privacy and fairness in generative models.

Remark 1. *We emphasize the need for a framework tailored to generative settings. There are notable fairness-privacy techniques for classification, but directly extending them to data generation can be challenging due to the fundamentally different goals of the two settings – see more details in Sec. A.*

4 Framework

We now propose PFGuard, the *first* generative framework that simultaneously achieves statistical fairness and DP on high-dimensional data, such as images. As shown in Fig. 2, PFGuard balances privacy-fairness conflicts between fair and private training stages using an ensemble of teacher models as a key component. In Sec. 4.1, we first explain the *fair training stage*, which trains a fair teacher ensemble. In Sec. 4.2, we then explain the *private training stage*, which transfers the knowledge of this teacher ensemble to the generator with DP guarantees – ultimately training a generator that is both fair and private. In Sec. 4.3, we lastly discuss how PFGuard's integrated design offers advantages in terms of fairness, utility, and privacy compared to the naïve approaches discussed in Sec. 3.

4.1 Fair Training with Balanced Minibatch Sampling

Intuition We ensure fairness in the teachers by balancing the *minibatches* used for training. Here we assume a general training setup of stochastic gradient descent, where we iteratively pick a subset $\mathcal{B}$ of the training data (i.e., minibatches) to update model parameters more efficiently. Since generative models then only learn the underlying data distribution through $\mathcal{B}$, using *balanced minibatches* $\mathcal{B} \sim p_{bal}$ *will result in modeling* p_{bal} even if we have biased training data $\mathcal{D} \sim p_{\text{bias}}$. While another approach is to debias the training data itself by acquiring more minority data, this approach is costly and often infeasible for private domains with limited publicly-available data (Jordon et al., 2018).

Theoretical Foundation To develop a fair minibatch sampling technique with a convergence guarantee, we leverage *Sampling-Importance Resampling* (SIR) (Rubin, 1988; Smith & Gelfand, 1992) as the theoretical foundation. SIR is a statistical method for drawing random samples from a target distribution $\pi(x)$ by using a proposal distribution $\psi(x)$. SIR proceeds in two steps: 1) we draw a set of n independent random samples $\mathcal{R}_1 = \{x_i\}_{i=1}^n$ from $\psi(x)$ and 2) we *resample* a smaller set of m independent random samples $\mathcal{R}_2 = \{x_i\}_{i=1}^m$ from $\mathcal{R}_1$. Here, the resampling probability $w(x_i)$ is set proportional to $h(x_i)$, where $h(x_i) = \pi(x_i)/\psi(x_i)$. Then, the resulting samples in $\mathcal{R}_2$ are approximately distributed according to $\pi(x)$ as follows:

$$\Pr(x \leq t) = \sum\nolimits_{i:x_i \leq t} w(x_i) = \sum\nolimits_{i:x_i \leq t} \frac{h(x_i)}{\sum_i h(x_i)} = \frac{\sum_i \mathbb{1}\{x_i \leq t\}\pi(x_i)/\psi(x_i)}{\sum_i \pi(x_i)/\psi(x_i)} \quad (2)$$

$$\underset{n\to\infty}{\to} \frac{\int \mathbb{1}\{x \leq t\}\{\pi(x)/\psi(x)\}\psi(x)dx}{\int \{\pi(x)/\psi(x)\}\psi(x)dx} = \int \mathbb{1}\{x \leq t\}\pi(x)dx \quad (3)$$

where $\mathbb{1}(\cdot)$ is the indicator function. The distribution becomes exact when $n \to \infty$.

Methodology We now present our sampling technique, which guarantees $\mathcal{B} \sim p_{\text{bal}}$ based on SIR. We first make the following reasonable assumptions: 1) each data sample has a uniquely defined sensitive attribute $\mathbf{s} \in \mathcal{S}$ (e.g., race); 2) p_{bal} is uniformly distributed over $\mathbf{s}$; 3) following Choi et al. (2020), the same relevant input features are shared for each group $\mathbf{s}$ between the balanced and biased datasets (e.g., $p_{\text{bal}}(\mathbf{x}|\mathbf{s}{=}s) = p_{\text{bias}}(\mathbf{x}|\mathbf{s}{=}s)$), and similarly between the training dataset $\mathcal{D}$ and any subset $\mathcal{D}_i$ (e.g., $p_{\mathcal{D}}(\mathbf{x}|\mathbf{s}{=}s) = p_{\mathcal{D}_i}(\mathbf{x}|\mathbf{s}{=}s)$). We now outline the technique step-by-step below.

(1) We set the target distribution to $p_{\text{bal}}(\mathbf{x})$ and the proposal distribution to $p_{\text{bias}}(\mathbf{x})$, as our goal is to sample a balanced minibatch $\mathcal{B} \sim p_{\text{bal}}$ from the biased training dataset $\mathcal{D} \sim p_{\text{bias}}$.

(2) We divide $\mathcal{D}$ into n_T disjoint subsets $\{\mathcal{D}_i\}_{i=1}^{n_T}$ to train each teacher model T_i, such that each $\mathcal{D}_i$ retains the same value distribution of $\mathbf{s}$ as $\mathcal{D}$ using the $\mathbf{s}$ labels (i.e., $p_{\mathcal{D}_i}(\mathbf{s} = s) = p_{\mathcal{D}}(\mathbf{s} = s)$). Then, we can derive $\mathcal{D}_i \sim p_{\text{bias}}$ using assumption 3) above as follows:

$$p_{\mathcal{D}_i}(\mathbf{x}) = \Sigma_s p_{\mathcal{D}_i}(\mathbf{x}|\mathbf{s}{=}s) p_{\mathcal{D}_i}(\mathbf{s}{=}s) = \Sigma_s p_{\mathcal{D}}(\mathbf{x}|\mathbf{s}{=}s) p_{\mathcal{D}}(\mathbf{s}{=}s) = p_{\text{bias}}(\mathbf{x}) \quad (4)$$

(3) We sample $\mathcal{B}$ from $\mathcal{D}_i$ with a resampling probability $w(\mathbf{x})$ that is proportional to $h(\mathbf{x}) = p_{\text{bal}}(\mathbf{x})/p_{\text{bias}}(\mathbf{x})$, which is computed as follows:

$$h(\mathbf{x}) = \frac{p_{\text{bal}}(\mathbf{x})}{p_{\text{bias}}(\mathbf{x})} = \frac{p_{\text{bal}}(\mathbf{x}|\mathbf{s}{=}s) p_{\text{bal}}(\mathbf{s}{=}s)}{p_{\text{bias}}(\mathbf{x}|\mathbf{s}{=}s) p_{\text{bias}}(\mathbf{s}{=}s)} = \frac{p_{\text{bal}}(\mathbf{s}{=}s)}{p_{\text{bias}}(\mathbf{s}{=}s)} \simeq \frac{1/|\mathcal{S}|}{|\{\mathbf{x}{\in}\mathcal{D}|\mathbf{s}{=}s\}|/|\mathcal{D}|} \quad (5)$$

where the second and third equality follows from assumption 1) and 3) above, respectively, and the last approximation follows from assumption 2) above and $\mathcal{D} \sim p_{\text{bias}}$.

A sample $\mathcal{B}$ from the above procedure is approximately distributed according to p_{bal} based on SIR; we sample $\mathcal{D}_i$ from p_{bias} and resample $\mathcal{B}$ from D_i, where the resampling probability is proportional to $h(\mathbf{x}) = p_{\text{bal}}(\mathbf{x})/p_{\text{bias}}(\mathbf{x})$. Since a large number of minibatch samplings is needed to train generative models, the $\mathcal{B}$ distribution eventually converges to p_{bal}, leading to a fair generative modeling of p_{bal}.

Extensions Our fair sampling technique is also extensible to private settings where the label of sensitive attribute $\mathbf{s}$ is *unavailable*, for example due to privacy regulations (Jagielski et al., 2019; Mozannar et al., 2020; Tran et al., 2022). In such settings, we can employ a binary classification approach to estimate $h(\mathbf{x})$ like Choi et al. (2020). While their work focuses on *non-private settings* and assumes an unbiased public reference data on the order of 10%–100% of $|\mathcal{D}|$ for the estimation, this assumption can be unrealistic in private domains due to the lack of public data. Our empirical study in Sec. 5.3 shows that we can achieve fairness with only 1–10% of the data, leveraging the ensemble learning of *multiple-teacher structure*, which can further reduce the estimation error. Note that in this extension, the convergence guarantee may not hold, as $\mathcal{D}_i \sim p_{\text{bias}}$ in step (2) might not be true in practice if the dataset is randomly partitioned without considering sensitive attribute labels.

4.2 Private Training with Private Teacher Ensemble Learning

Intuition We ensure DP by privatizing *knowledge transfer* from a teacher ensemble to the generator. Although the sampling technique in Sec. 4.1 transfers more balanced knowledge, *privacy risks* can be also transferred due to privacy-fairness conflicts. For example, if certain data samples are resampled repeatedly during teacher training, privacy risks like memorization can occur in the teacher models and be transferred to the generator. We thus privatize the knowledge transfer with DP techniques to provide strict DP guarantees in the generator. Note that only the generator needs to have privacy as it is the one that is released publicly to produce synthetic data.

Privacy Guarantee We utilize Private Teacher Ensemble Learning (PTEL) (Papernot et al., 2016; 2018) to ensure DP in the knowledge transfer. Compared to non-private ensemble learning, PTEL 1) assumes each teacher model is trained on a disjoint data subset and 2) adds noise proportional to the sensitivity of the knowledge aggregation. Here, sensitivity is derived from *data disjointness*, where a single data point affects at most one teacher. For example, GNMax aggregator (Papernot et al., 2018) aggregates prediction of teacher models $\{T_i\}_{i=1}^{n_T}$ on a query input $\bar{\mathbf{x}}$ for its class label as follows:

$$\text{GNMax}(\bar{\mathbf{x}}) = \arg\max_j \{n_j(\bar{\mathbf{x}}) + \mathcal{N}(0, \sigma^2)\} \quad \text{for } j = 1, ..., c \tag{6}$$

where $n_j(\mathbf{x})$ denotes the vote count for the j-th class (i.e., $n_j(\bar{\mathbf{x}}) = |\{i : T_i(\bar{\mathbf{x}}){=}j\}|$), and $\mathcal{N}(0, \sigma^2)$ denotes random Gaussian noise. Here, the l_2-sensitivity (Def. 2.2) is $\sqrt{2}$, as a single data point affects at most one teacher, increasing the vote counts by 1 for one class and decreasing the count by 1 for another class (see a more detailed analysis in Sec. B.2). Consequently, the GNMax aggregator satisfies (ε, δ)-DP for $\sigma \geq \sqrt{8 \ln(1.25/\delta)}/\varepsilon$ based on the Gaussian mechanism (Thm. 2.1).

Methodology PFGuard can be easily integrated with existing PTEL-based generative models by simply modifying the minibatch sampling process as described in Sec. 4.1. PTEL has been widely adopted to train generators with privatized teacher output to ensure DP (Jordon et al., 2018; Chen et al., 2020; Long et al., 2021; Wang et al., 2021a). Although the exact sensitivity values of these PTEL-based generative models vary depending on what teacher knowledge is aggregated (e.g., votes on class labels (Jordon et al., 2018) or gradient directions (Wang et al., 2021a)), PFGuard preserves any sensitivity as long as the PTEL enforce data disjointness; even with fair sampling, a single data point still affects only one teacher. PFGuard is thus compatible with various PTEL-based generative models, enhancing fairness while preserving DP guarantees – see Sec. C for a more detailed analysis.

Number of Teachers We provide guidelines on how to set the number of teachers n_T for PFGuard, which affects the privacy-fairness-utility tradeoff. While n_T is typically tuned via experiments (Long et al., 2021; Wang et al., 2021a), we suggest a fairness constraint when setting n_T. Since a large n_T would result in a diverse ensemble that can generalize better, but also lead to a teacher receiving a data subset that is too small for training, we suggest n_T to be at most $\lfloor |\mathcal{D}| \min_{s \in \mathcal{S}} p_{\text{bias}}(s) \rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor function. Since this equation captures the size of the smallest minority data group, this mathematical upper bound guarantees that each teacher *probabilistically* gets at least one sample of the smallest minority data group. In Sec. 5.3, we show how this bound helps avoid compromising fairness. We also discuss how to set n_T when sensitive attribute labels are unavailable in Sec. C.3.

4.3 ADVANTAGES OF INTEGRATED DESIGN

We discuss how PFGuard overcomes the challenges of naïve approaches discussed in Sec. 3.

Balances Privacy-Fairness Conflict PFGuard can sidestep *privacy breaches* and *fairness cancellation* arising from privacy-fairness conflicts. Applying fairness-only techniques to existing DP generators can compromise DP guarantees and require complex sensitivity assessments; in contrast, PFGuard automatically preserves DP guarantees of any PTEL-based DP generators through data disjointness, eliminating the need of such assessments. Privacy-only techniques often use sample-level bounding, which *directly* limits an individual sample's influence (e.g., gradient clipping discussed in Sec. 3) and can lead to fairness cancellation by suppressing the use of minority data. In contrast, PFGuard uses *indirect* privacy bounding, in the sense that we limit the knowledge transfer of teacher models in order to limit individual sample's influence. Since there are no DP constraints during the teacher learning, the teacher models can effectively learn balanced knowledge across data groups.

Achieves Better Fairness-Utility Tradeoff The fair training of PFGuard adds minimal training complexity, preserving the utility for the subsequent private training stage. The proposed sampling technique requires a simple modification in the minibatch sampling process for fairness, avoiding the need for additional fairness loss terms (Sattigeri et al., 2019; Yu et al., 2020) or auxiliary classifiers (Tan et al., 2020; Um & Suh, 2023) typically employed in fairness-only techniques. In Sec. 5, we also show that PFGuard incurs negligible overhead in computation time when integrated with existing PTEL-based generative models.

5 EXPERIMENTS

We perform experiments to evaluate PFGuard's effectiveness in terms of fairness, privacy, and utility.

Datasets We evaluate PFGuard on three image datasets: 1) *MNIST* (LeCun et al., 1998) and *FashionMNIST* (Xiao et al., 2017) for various analyses and baseline comparisons, and 2) *CelebA* (Liu et al., 2015) to observe performance in real-world scenarios more closely related to privacy and fairness concerns. Here, MNIST contains handwritten digit images, FashionMNIST contains clothing item images, and CelebA contains facial images. While MNIST and FashionMNIST are simplistic and less reflective of real-world biases, they enable reliable fairness analyses on top of high-performing DP generative models on these datasets, making them widely adopted in recent studies addressing the privacy-fairness intersections (Bagdasaryan et al., 2019; Farrand et al., 2020; Ganev et al., 2022). For CelebA, we resize the images to $32 \times 32 \times 3$ (i.e., *CelebA(S)*) and to $64 \times 64 \times 3$ (i.e., *CelebA(L)*) following the conventions in the DP generative model literature (Long et al., 2021; Wang et al., 2021a; Cao et al., 2021). More dataset details are in Sec. D.1.

Bias Settings We create various bias settings across classes and subgroups, focusing on four scenarios: 1) *binary class bias*, which is a basic scenario often addressed in DP generative models, and 2) *multi-class bias*, *subgroup bias*, and *unknown subgroup bias*, which are more challenging scenarios typically addressed in fairness techniques, but not in DP generative models. We observe that DP generative models mostly perform poorly in these challenging scenarios, especially with complex datasets like CelebA, so we use MNIST for more reliable analyses. While recent privacy-fairness studies on MNIST mostly focus on class bias (Bagdasaryan et al., 2019; Farrand et al., 2020), we additionally analyze subgroup bias using image rotation for more fine-grained fairness analyses and to support prominent fairness metrics like equalized odds (Hardt et al., 2016). In all experiments, we denote $Y = 0$ as the minority class and $S = 0$ as the minority group. More details on bias levels (e.g., size ratios between majority and minority data) and bias creation are in Sec. D.1.

Metrics We evaluate utility, privacy, and fairness in both synthetic data and downstream tasks.

- *Utility.* We measure the overall and groupwise *Frechet Inception Distance* (FID) (Heusel et al., 2017) to evaluate the sample quality of synthetic data. We evaluate *model accuracy* in downstream tasks by training Multi-layer Perceptrons (MLP) and Convolutional Neural Networks (CNN) on synthetic data and testing on real datasets (Chen et al., 2020).

- *Fairness.* We measure the group size disparity in synthetic data with the *KL divergence* to uniform distribution $U(S)$ (i.e., $D_{KL}(p_G(S)||U(S))$) (Yu et al., 2020) and the *distribution disparity* (i.e., $|p_G(S) - U(S)|$) (Choi et al., 2020), where $p_G(S)$ denotes generated distribution w.r.t. S. We measure the fairness disparities in downstream tasks as follows: *equalized odds disparity* (i.e., $\max_{y,s_1,s_2} |\Pr(\hat{Y}{=}y|Y{=}y, S{=}s_1) - \Pr(\hat{Y}{=}y|Y{=}y, S{=}s_2)|,\ \forall y \in \mathcal{Y},\ s_1, s_2 {\in} \mathcal{S}$), *demographic disparity* (i.e., $\max_{s_1,s_2} |\Pr(\hat{Y}{=}1|S{=}s_1) - \Pr(\hat{Y}{=}1|S{=}s_2)|,\ \forall s_1, s_2 \in \mathcal{S}$), and *accuracy disparity* (i.e., $\max_{y_1,y_2} |\Pr(\hat{Y}{=}y_1|Y{=}y_1) - \Pr(\hat{Y}{=}y_2|Y{=}y_2)|,\ \forall y_1, y_2 \in \mathcal{Y}$).
- *Privacy.* We use privacy budget ε for DP (Def. 2.1), which is preserved in both synthetic data and downstream tasks due to the post-processing property of DP (see more details in Sec. B.1).

Baselines We compare PFGuard with three types of baselines: 1) privacy-only and fairness-only approaches for data generation, 2) simple combinations of these methods, and 3) recent privacy-fairness classification methods applicable to data generation. For 1) and 2), we use three state-of-the-art DP generative models – GS-WGAN (Chen et al., 2020), G-PATE (Long et al., 2021) and DataLens (Wang et al., 2021a) – and a widely-adopted fair reweighting method (Choi et al., 2020). For 3), we extend DP-SGD-F (Xu et al., 2020) and DPSGD-Global-Adapt (Esipova et al., 2022), which are fair variants of DP-SGD (Abadi et al., 2016). Specifically, we replace the DP-SGD used in GS-WGAN with these fairness-enhanced variants. We faithfully implement all baseline methods with their official codes and reported hyperparameters. More details on baseline methods are in Sec. D.2.

5.1 Improving Existing Privacy-Only Generative Models

We evaluate how PFGuard enhances the performance of existing DP generative models. As PFGuard guarantees the same level of DP, we focus on the fairness and utility performances while fixing the privacy budget to $\varepsilon{=}10$, which is one of the most conventional values (Ghalebikesabi et al., 2023).

Analysis on Synthetic Data Table 1 shows the fairness and utility performances on synthetic data. Private generative models generally produce synthetic data with better overall image quality, but exhibit high group size disparity. In contrast, PFGuard significantly improves fairness by balancing group size and groupwise image quality, with a slight decrease in overall image quality.

Table 1: Fairness and utility performances of private generative models with and without PFGuard on synthetic data, evaluated on MNIST with subgroup bias under $\varepsilon{=}10$. Blue and red arrows indicate positive and negative changes, respectively. Lower values are better across all metrics.

	Fairness		Utility				
Method	KL (↓)	Dist. Disp. (↓)	FID (↓)	Y=1, S=1	Y=1, S=0	Y=0, S=1	Y=0, S=0
GS-WGAN	0.177±0.103	0.383±0.097	**77.97±2.25**	**95.58±3.35**	155.20±16.25	89.66±0.79	101.39±7.09
G-PATE	0.305±0.011	0.522±0.008	176.03±3.03	182.50±1.27	183.31±2.99	178.89±4.13	187.37±3.51
DataLens	0.220±0.030	0.450±0.028	192.29±3.67	197.13±6.18	197.99±6.01	202.86±4.12	207.12±12.75
GS-WGAN + PFGuard	**0.067±0.036** (↓)	**0.242±0.080** (↓)	83.67 ±6.98 (↑)	114.54±27.74	**149.47±17.31**	**79.94±7.08**	**72.44±7.96**
G-PATE + PFGuard	0.206 ±0.062 (↓)	0.431 ±0.066 (↓)	166.89 ±21.61 (↓)	173.48±19.93	173.79±19.43	174.98±24.06	185.92±19.89
DataLens + PFGuard	0.161 ±0.019 (↓)	0.389 ±0.022 (↓)	200.23 ±3.11 (↑)	209.74±1.70	208.80±0.39	207.03±4.67	207.05±3.17

Analysis on Downstream Tasks Table 2 shows the fairness and utility performances on downstream tasks. Compared to the synthetic data analysis, PFGuard enhances not only fairness, but also overall utility, especially for CNN models. We suspect that the increased overall utility results from the improved fairness in the input synthetic data, promoting more balanced learning among groups.

Table 2: Fairness and utility performances of private generative models with and without PFGuard on downstream tasks, evaluated on MNIST with subgroup bias under $\varepsilon{=}10$. Blue and red arrows indicate positive and negative changes, respectively.

	MLP			CNN		
	Fairness		Utility	Fairness		Utility
Method	EO Disp. (↓)	Dem. Disp. (↓)	Acc (↑)	EO Disp. (↓)	Dem. Disp. (↓)	Acc (↑)
GS-WGAN	0.153±0.030	0.061±0.012	**0.910±0.007**	0.172±0.045	0.069±0.014	**0.927±0.008**
G-PATE	0.166±0.082	0.063±0.053	0.896±0.005	0.256±0.046	0.111±0.001	0.888±0.015
DataLens	0.226±0.062	0.112±0.035	0.867±0.028	0.238±0.044	0.110±0.023	0.893±0.022
GS-WGAN + PFGuard	**0.067±0.029** (↓)	**0.044±0.012** (↓)	0.900 ±0.003 (↓)	**0.063±0.059** (↓)	**0.035±0.037** (↓)	**0.927±0.009** (–)
G-PATE + PFGuard	0.085 ±0.052 (↓)	0.044±0.033 (↓)	0.906±0.008 (↑)	0.084 ±0.036 (↓)	0.044 ±0.011 (↓)	0.898±0.023 (↑)
DataLens + PFGuard	0.169±0.081 (↓)	0.106±0.043 (↓)	0.859±0.056 (↓)	0.141±0.050 (↓)	0.078±0.051 (↓)	0.898±0.020 (↑)

Table 3: Comparison of privacy-fairness-utility performance on MNIST under ε=10, using GS-WGAN as the base DP generator (see Sec. D.2 for more details). The first three rows represent upper bound performances for vanilla, DP-only, and fair-only models. Evaluations cover both subgroup bias and unknown subgroup bias, where "no S" indicates whether the method is applicable without group labels. "perc" denotes the proportion of public data used compared to the training data size. "-" indicates no samples are generated. Lower values are better across all metrics, and we boldface the best results in each subgroup bias and unknown subgroup bias settings.

	Privacy	Fairness			Utility				
Method	ε (↓)	KL (↓)	Dist. Disp. (↓)	no S	FID (↓)	Y=1, S=1	Y=1, S=0	Y=0, S=1	Y=0, S=0
Vanila	✗	0.229	0.459	✗	31.95	28.01	55.53	44.04	63.50
DP-only	10	0.177	0.383	✗	77.97	95.58	155.20	89.66	101.39
Fair-only	✗	0.021	0.117	✓	38.62	50.78	52.69	75.46	53.86
Reweighting	13	**0.009**	**0.044**	✗	106.94	139.28	178.18	128.08	110.54
DP-SGD → DP-SGD-F	11	0.659	0.494	✗	90.20	121.78	-	**73.07**	159.83
DP-SGD → DPSGD-GA	**10**	0.693	0.707	✓	127.02	167.65	-	126.15	-
PFGuard	**10**	0.067	0.242	✗	**83.67**	**114.54**	**149.47**	79.94	**72.24**
Reweighting (perc=1.0)	13	0.025	0.148	✓	98.57	144.55	182.29	96.05	99.59
Reweighting (perc=0.1)	13	0.013	0.113	✓	106.94	139.28	178.18	128.08	110.54
PFGuard (perc=0.1)	**10**	**0.004**	**0.041**	✓	**89.43**	**130.36**	**157.80**	**78.75**	**89.76**

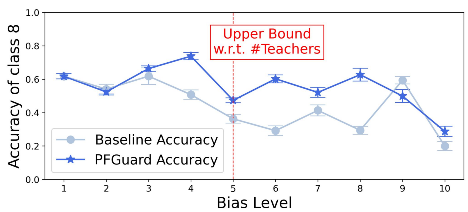

Figure 4: Fairness performances when varying bias levels (γ) given a fixed number of teachers, evaluated on MNIST with multi-class bias. We downsize the class '8' to γ times smaller than the other classes to make it the minority class and use GS-WGAN as the baseline model.

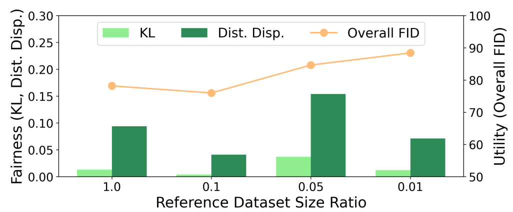

Figure 5: Fairness and utility performances for varying reference dataset size ratio compared to the training dataset size, evaluated on MNIST with unknown subgroup bias under ε=10. Lower values are better across all metrics used to evaluate fairness and utility.

5.2 Privacy-Fairness-Utility Tradeoff

We compare our privacy-fairness-utility performance with naïve combinations of prior approaches. We evaluate performance under two bias settings: 1) subgroup bias and 2) unknown subgroup bias. Table 3 shows the results, which aligns with our privacy-fairness counteraction analysis in Sec. 3. On the one hand, fairness-only reweighting approaches compromise privacy due to the increased iterations from modifying the loss function for fair training (i.e., the more a model uses the data, the weaker privacy it provides). On the other hand, privacy-fairness classification techniques can maintain the original privacy guarantees, but significantly degrade utility and fairness, resulting in lower image quality and size disparities across groups (see more discussions in Sec. A) In contrast, PFGuard is the only method that successfully achieves both privacy and fairness and preserves the closest utility to the original models – more results including Pareto Frontier results are in Sec. E.1.

5.3 Ablation Study

Fairness Upper Bound on Number of Teachers We validate the proposed theoretical upper bound on the number of teachers for fairness, which depends on the bias level of the training data. To effectively simulate scenarios where a teacher receives only a small subset of minority data, we evaluate PFGuard in a multi-class bias setting, downsizing the minority class (i.e., class 8 for MNIST) by a factor of γ. Given that MNIST has fewer than 6,000 samples for class 8, our proposed upper bound is $\gamma \leq 5$ if we fix the number of teachers to 1,000. Fig. 4 shows that exceeding γ=5 leads to a noticeable decline in accuracy for the minority class, which is consistent with our theoretical results. It is noteworthy that even with the decline, PFGuard shows higher accuracy than the privacy-only baseline, which shows a consistent decrease in accuracy for the minority as γ increases.

Impact of Reference Dataset Size We explore the influence of the reference dataset size when PFGuard is extended to unknown subgroup bias setting. Fig. 5 shows PFGuard achieves comparable fairness even with a small reference dataset size, while showing a slight decline in the overall utility.

More Analyses We provide more experiments in Sec. E, including a comparison of computation time (Sec. E.3), results on different datasets such as FashionMNIST (Sec. E.4), and employing an additional normalization technique to further enhance the overall image quality (Sec. E.5).

5.4 Analysis with Stronger Privacy, High-Dimensional Images

We provide preliminary results with CelebA dataset, which mirrors real-world scenarios with high-dimensional facial images. As our study is the first to address both privacy and fairness in image data, this exploration is crucial for understanding the challenges in real-world settings. To reflect the need of stronger privacy protection in practical applications, we limit the privacy budget to ε=1.

Table 4 shows the fairness and utility performances under these challenging conditions. We observe that DP generative models often exhibit extreme accuracy disparities even with a simplistic class bias setting, achieving over 90% accuracy for the majority class while achieving accuracy below 25% for the minority class. PFGuard consistently enhances the minority class performance and reduces accuracy disparity, while there is still room for improvements. Our results underscore the importance of tackling both privacy and fairness in future studies, encouraging more research in this critical area.

Table 4: Fairness and utility performances of private generative models with and without PFGuard on downstream tasks, evaluated on CelebA with binary class bias under ε=1. GS-WGAN is excluded due to lower image quality in this setting. Blue and red arrows indicate positive and negative changes, respectively. We provide the full table with standard deviations in Sec. E.6.

	CelebA(S)				CelebA(L)			
	Fairness	Utility			Fairness	Utility		
Method	Acc. Disp. (↓)	Acc (↑)	Y=0	Y=1	Acc. Disp. (↓)	Acc (↑)	Y=0	Y=1
G-PATE	0.978	0.666	0.014	**0.992**	0.968	0.668	0.023	**0.991**
DataLens	0.793	0.643	0.114	0.907	0.678	0.686	0.234	0.912
G-PATE + PFGuard	0.736 (↓)	0.678 (↑)	0.187 (↑)	0.923 (↓)	**0.277** (↓)	0.563 (↓)	**0.378** (↑)	0.655 (↓)
DataLens + PFGuard	**0.725** (↓)	**0.689** (↑)	**0.205** (↑)	0.931 (↑)	0.641 (↓)	**0.704** (↑)	0.276 (↑)	0.917 (↑)

6 Related Work

We cover the private and fair data generation literature here and cover the 1) private-only data generation, 2) fair-only data generation, 3) privacy-fairness intersection literature in Sec. F. Compared to these lines of works, only a few works focus on private and fair data generation (Xu et al., 2021; Pujol et al., 2022). First, (Xu et al., 2021) proposes a two-step approach that removes bias from the training data via a fair pre-processing technique (Celis et al., 2020) and learns a DP generative model (Chanyaswad et al., 2019) from the debiased data. However, this framework is limited to low-dimensional structural data due to data binarization step in pre-precessing stage, which can incur significant information loss in high-dimensional image data. PFGuard, on the other hand, can generate high-dimensional image data with high quality. Second, (Pujol et al., 2022) proposes private data generation techniques satisfying causality-based fairness (Salimi et al., 2019), which consider the causal relationship between attributes. In comparison, PFGuard focuses on statistical fairness to achieve similar model performances for sensitive groups (Barocas et al., 2018). While causality-based approaches can better reveal the causes of discrimination than statistical approaches, modeling an underlying causal mechanism for real-world scenarios is also known to be challenging.

7 Conclusion

We proposed PFGuard, a fair and private generative model training framework. We first identified the counteractive nature between privacy preservation and fair training, demonstrating potential adverse effects – such as privacy breaches or fairness cancellation – when two objectives are addressed independently. We then designed PFGuard, which prevents the counteractions by using multiple teachers to harmonize fair sampling and private teacher ensemble learning. We showed how this integrated design of PFGuard offers multiple advantages, including a better fairness-privacy-utility tradeoff compared to other baselines, ease of deployment, and support for high-dimensional data.

Ethics Statement & Limitation We believe our research addresses the critical issue of Trustworthy AI. Our focus on privacy and fairness underscores the need to design AI models that simultaneously safeguard individual privacy and mitigate biases without perpetuating them. In addition, our research and experiments are conducted with a strong commitment to ethical standards. All datasets used in this study, including publicly available human images, are widely used within the research community and do not contain sensitive or harmful content. There are also limitations, and we note that choosing the right privacy and fairness measures for an application can be challenging and also depends on the social context. We also note that the use of multiple teacher does increase the cost of training, but provides more benefits particularly in balancing privacy and fairness.

Reproducibility Statement All datasets, methodologies, and experimental setups utilized in our study are described in detail in the supplementary material. More specifically, we provide a description of the proposed algorithm in Sec. C.2, details of datasets and preprocessing in Sec. D.1, and implementation details including hyperparameters in Sec. D.2 to ensure reproducibility.

REFERENCES

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*, pp. 308–318, 2016.

Sushant Agarwal. Trade-offs between fairness and privacy in machine learning. In *IJCAI 2021 Workshop on AI for Social Good*, 2021.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness in machine learning. *Nips tutorial*, 1: 2, 2017.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. Fairness and machine learning. fairmlbook. org, 2019, 2018.

Alex Bie, Gautam Kamath, and Guojun Zhang. Private gans, revisited. *arXiv preprint arXiv:2302.02936*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. *Advances in Neural Information Processing Systems*, 36, 2024.

Blake Bullwinkel, Kristen Grabarz, Lily Ke, Scarlett Gong, Chris Tanner, and Joshua Allen. Evaluating the fairness impact of differentially private synthetic data. *arXiv preprint arXiv:2205.04321*, 2022.

Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34:12480–12492, 2021.

L Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach. In *International Conference on Machine Learning*, pp. 1349–1359. PMLR, 2020.

Thee Chanyaswad, Changchang Liu, and Prateek Mittal. Ron-gauss: Enhancing utility in non-interactive private data release. *Proceedings on Privacy Enhancing Technologies*, 2019(1):26–46, 2019.

Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.

Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. Dpgen: Differentially private generative energy-guided network for natural image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8387–8396, 2022.

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*, pp. 149–160, 2021.

Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, pp. 1887–1898. PMLR, 2020.

Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pp. 309–315, 2019.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pp. 265–284. Springer, 2006b.

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.

Maria S Esipova, Atiyeh Ashari Ghomi, Yaqiao Luo, and Jesse C Cresswell. Disparate impact in differential privacy from gradient misalignment. *arXiv preprint arXiv:2206.07737*, 2022.

Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 workshop on privacy-preserving machine learning in practice*, pp. 15–19, 2020.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.

Germain Forestier and Cédric Wemmert. Semi-supervised learning using multiple clusterings with limited labeled data. *Information Sciences*, 361:48–65, 2016.

Georgi Ganev, Bristena Oprisanu, and Emiliano De Cristofaro. Robin hood and matthew effects: Differential privacy has disparate impact on synthetic data. In *International Conference on Machine Learning*, pp. 6944–6959. PMLR, 2022.

Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceceding of the Advances in Neural Information Processing Systems*, 2014.

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*, pp. 1819–1827. PMLR, 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol.*, 2019(4): 232–249, 2019.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Magnet: Uniform sampling from deep generative network manifolds without retraining. In *International Conference on Learning Representations*, 2021.

Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. In *International Conference on Machine Learning*, pp. 3000–3008. PMLR, 2019.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Yeongmin Kim, Byeonghu Na, Minsang Park, JoonHo Jang, Dongjun Kim, Wanmo Kang, and Il-Chul Moon. Training unbiased diffusion models from biased dataset. *arXiv preprint arXiv:2403.01189*, 2024.

Bogdan Kulynych, Yao-Yuan Yang, Yaodong Yu, Jarosław Błasiok, and Preetum Nakkiran. What you see is what you get: Principled deep learning via distributional generalization. *Advances in Neural Information Processing Systems*, 35:2168–2183, 2022.

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.

Seng Pei Liew, Tsubasa Takahashi, and Michihiko Ueno. Pearl: Data synthesis via private embeddings and adversarial reconstruction learning. *arXiv preprint arXiv:2106.04590*, 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

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. *Advances in Neural Information Processing Systems*, 34:2965–2977, 2021.

Andrew Lowy, Devansh Gupta, and Meisam Razaviyayn. Stochastic differentially private and fair learning. In *Workshop on Algorithmic Fairness through the Lens of Causality and Privacy*, pp. 86–119. PMLR, 2023.

J Macqueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.

Ilya Mironov. Rényi differential privacy. In *CSF*, pp. 263–275. IEEE, 2017.

Hussein Mozannar, Mesrob Ohannessian, and Nathan Srebro. Fair learning with private demographic data. In *International Conference on Machine Learning*, pp. 7066–7075. PMLR, 2020.

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.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

David Pujol, Ryan McKenna, Satya Kuppam, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair decision making using privacy-protected data. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 189–199, 2020.

David Pujol, Amir Gilad, and Ashwin Machanavajjhala. Prefair: Privately generating justifiably fair synthetic data. *arXiv preprint arXiv:2212.10310*, 2022.

Yuji Roh, Weili Nie, De-An Huang, Steven Euijong Whang, Arash Vahdat, and Anima Anandkumar. Dr-fairness: Dynamic data ratio adjustment for fair training on real and generated data. *Transactions on Machine Learning Research*, 2023.

Lucas Rosenblatt, Julia Stoyanovich, and Christopher Musco. A simple and practical method for reducing the disparate impact of differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21554–21562, 2024.

Donald B Rubin. Using the sir algorithm to simulate posterior distribution. *Bayesian statistics*, 3: 395–402, 1988.

Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, pp. 793–810, 2019.

Amartya Sanyal, Yaxi Hu, and Fanny Yang. How unfair is private learning? In *Uncertainty in Artificial Intelligence*, pp. 1738–1748. PMLR, 2022.

Prasanna Sattigeri, Samuel C Hoffman, Vijil Chenthamarakshan, and Kush R Varshney. Fairness gan: Generating datasets with fairness properties using a generative adversarial network. *IBM Journal of Research and Development*, 63(4/5):3–1, 2019.

Øivind Skare, Erik Bølviken, and Lars Holden. Improved sampling-importance resampling and reduced bias importance sampling. *Scandinavian Journal of Statistics*, 30(4):719–737, 2003.

Adrian FM Smith and Alan E Gelfand. Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

Hui Sun, Tianqing Zhu, Zhiqiu Zhang, Dawei Jin, Ping Xiong, and Wanlei Zhou. Adversarial attacks against deep generative models on data: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3367–3388, 2021.

Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 169–180. IEEE, 2021.

Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020.

Christopher TH Teo, Milad Abdollahzadeh, and Ngai-Man Cheung. Fair generative models via transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2429–2437, 2023.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Cuong Tran, My Dinh, and Ferdinando Fioretto. Differentially private empirical risk minimization under the fairness lens. *Advances in Neural Information Processing Systems*, 34:27555–27565, 2021a.

Cuong Tran, Ferdinando Fioretto, and Pascal Van Hentenryck. Differentially private and fair deep learning: A lagrangian dual approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9932–9939, 2021b.

Cuong Tran, Keyu Zhu, Ferdinando Fioretto, and Pascal Van Hentenryck. Sf-pate: scalable, fair, and private aggregation of teacher ensembles. *arXiv preprint arXiv:2204.05157*, 2022.

Soobin Um and Changho Suh. A fair generative model using lecam divergence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10034–10042, 2023.

Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. Hermite polynomial features for private data generation. In *International Conference on Machine Learning*, pp. 22300–22324. PMLR, 2022.

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*, pp. 2146–2168, 2021a.

Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021b.

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

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

Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pp. 570–575. IEEE, 2018.

Depeng Xu, Shuhan Yuan, and Xintao Wu. Achieving differential privacy and fairness in logistic regression. In *Companion proceedings of The 2019 world wide web conference*, pp. 594–599, 2019.

Depeng Xu, Wei Du, and Xintao Wu. Removing disparate impact of differentially private stochastic gradient descent on model accuracy. *arXiv preprint arXiv:2003.03699*, 2020.

Weijie Xu, Jinjin Zhao, Francis Iannacci, and Bo Wang. Ffpdg: Fast, fair and private data generation. In *ICLR 2021 Workshop on Synthetic Data Generation*, 2021.

Mohammad Yaghini, Patty Liu, Franziska Boenisch, and Nicolas Papernot. Learning with impartiality to walk on the pareto frontier of fairness, privacy, and utility. *arXiv preprint arXiv:2302.09183*, 2023.

Yilin Yang, Kamil Adamczewski, Danica J Sutherland, Xiaoxiao Li, and Mijung Park. Differentially private neural tangent kernels for privacy-preserving data generation. *arXiv preprint arXiv:2303.01687*, 2023.

Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision*, pp. 377–393. Springer, 2020.

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

Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.

D. Zietlow, M. Lohaus, G. Balakrishnan, M. Kleindessner, F. Locatello, B. Scholkopf, and C. Russell. Leveling down in computer vision: Pareto inefficiencies in fair deep classifiers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10400–10411, 2022.

A Challenges of Extending Classification Techniques

Continuing from Sec. 3, we provide more details of potential challenges when one tries to extend fairness-privacy classification techniques (Jagielski et al., 2019; Mozannar et al., 2020; Tran et al., 2021b; 2022) to generative settings due to the fundamentally different goals.

Different DP Notions and Assumptions Classification and generation settings often concentrate on *different DP notions* or rely on *different assumptions*, which hinders simple extensions of techniques between them. In the classification setting, Differential Privacy *w.r.t. sensitive attribute* (Jagielski et al., 2019) is commonly addressed (Jagielski et al., 2019; Mozannar et al., 2020; Tran et al., 2021b; 2022), which considers the demographic group attribute as the only private information. This DP notion requires less DP noise compared to a more general notion of DP (Def. 2.1), which protects *all* input features, and enables a better privacy-utility tradeoff for DP classifiers. However, in the generative setting, a general notion of DP is mostly addressed, as presumed non-private features may in fact encode private information (e.g., pixel values in a facial image). Therefore, simply extending classification techniques to generative settings can be challenging, as it necessitates rigorous mathematical proofs for corresponding DP notions and may add a large DP noise when adapting to a general DP notion. Moreover, classification techniques can rely on convex objective functions (Tran et al., 2021a), but the assumption of convexity does not usually hold in generative models (Goodfellow et al., 2014).

Challenges of Adjusting Privacy Bound While recent studies have proposed fair variants of gradient sanitization (Xu et al., 2020; Esipova et al., 2022), directly adopting them in existing private generative models can undermine the original utility and privacy guarantee. To prevent aggressive gradient clipping in minority data groups, approaches to tune clipping threshold C during training have been proposed, such as dynamically adjusting C during training (Esipova et al., 2022) or utilizing different C values w.r.t. groups (Xu et al., 2020). However, these adjustments of C not only consume additional privacy budget, but also can significantly affect model utility, as private generative models often demonstrates high sensitivity in model convergence to these clipping values (Chen et al., 2020; Wang et al., 2021a; Dockhorn et al., 2022). Hence, given the limited privacy budget and the necessity to carefully set the value of C, these approaches of tuning C may drastically change the original privacy-utility tradeoff of existing models to achieve fairness.

Here, we do not claim that extending classification techniques to generative settings is always impossible, but introduce the challenges that can complicate such extensions. To exemplify some possible cases, we extend methods from Esipova et al. (2022) and Xu et al. (2020), using them as baselines in our experiments – see Sec. 5 for results.

B Differential Privacy

Continuing from Sec. 2 and Sec. 4.2, we provide more details on differential privcy (DP).

B.1 Post-processing Property of Differential Privacy

Continuing from Sec. 2, we detail the post-processing property of DP. Let G be an (ε, δ)-DP generator, which produces the synthetic data from input random noise $\mathbf{z} \in \mathcal{Z}$. Then, the synthetic dataset $\tilde{D} = G(\mathbf{z})$ is also satisfies (ε, δ)-DP due to the post-processing theorem, as the random noise $\mathbf{z}$ is independent of the private dataset D, which is used to train the DP generator.

Theorem B.1. (Post-processing (Dwork et al., 2014)) Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}_1$ be a randomized mechanism that is (ε, δ)-DP. Let $f : \mathcal{R}_1 \to \mathcal{R}_2$ be an arbitrary function. Then $f \circ \mathcal{M} : \mathcal{D} \to \mathcal{R}_2$ is (ε, δ)-DP.

B.2 Sensitivity Analysis of GNMax Aggregator

Continuing from Sec. 4.2, we echo the sensitivity analysis of GNMax aggregator provided by Papernot et al. (2018) for readers' convenience.

Given $\{T_i\}_{i=1}^{n_T}$ teachers, c possible label classes, and a query input $\bar{\mathbf{x}}$, the teachers' vote count for the j-th class to a query input $\bar{\mathbf{x}}$ is denoted as:

$$n_j(\bar{\mathbf{x}}) = |\{i : T_i(\bar{\mathbf{x}}) = j\}| \quad \text{for } j = 1, ..., c \tag{7}$$

where T_i denotes the i-th teacher model. The vote count for each class is aggregated as follows:

$$\mathbf{n}(\bar{\mathbf{x}}) = (n_1, \ldots, n_c) \in \mathbb{N}^c \tag{8}$$

Since a single training data point only affects at most one teacher due to data disjointness, changing one data sample will at most change the votes by 1 for two classes, where we denote here as classes i and j. Given the two adjacent datasets $\mathcal{D}, \mathcal{D}'$ which differ by a single data point, let the aggregated vote counts are $\mathbf{n} = (n_1, \ldots, n_c)$ and $\mathbf{n}' = (n'_1, \ldots, n'_c)$, respectively. The l_2-sensitivity (Def. 2.2) can be derived as follows:

$$\Delta^2 = \max_{\mathcal{D},\mathcal{D}'} \|(n_1, \ldots, n_c) - (n'_1, \ldots, n'_c)\|_2 \tag{9}$$

$$= \max_{n_i, n'_i, n_j, n'_j} \|(0, \ldots, 0, n_i - n'_i, 0, \ldots, 0, n_j - n'_j, 0, \ldots, 0)\|_2 \tag{10}$$

$$= \max_{n_i, n'_i, n_j, n'_j} \sqrt{(n_i - n'_i)^2 + (n_j - n'_j)^2} \leq \sqrt{2} \tag{11}$$

C PFGuard Framework

Continuing from Sec. 4.2, we provide more details on the PFGuard framework.

C.1 Privacy anlaysis of PFGuard

Continuing from Sec. 4.2, we provide a theoretical proof of how PFGuard preserve a sensitivity of an arbitrary PTEL mechanism, and thus a sensitivity of arbitrary PTEL-based generative model.

PTEL-based generative models We first characterize the common training scheme of PTEL-based generative models (Jordon et al., 2018; Chen et al., 2020; Long et al., 2021; Wang et al., 2021a). To train a DP generator G parametrized with θ_G, the goal is to make its training algorithm $\mathcal{A} : \mathcal{D} \to G$ satisfy DP (Long et al., 2021). Given the typical training scheme of stochastic gradient descent that updates θ_G with gradient information $\boldsymbol{g}_G^{\text{up}}$, the training algorithm satisfies DP if $\boldsymbol{g}_G^{\text{up}}$ is processed with DP mechanism due to post-processing property of DP (Thm. B.1). To produce privatized gradient $\tilde{\boldsymbol{g}}_G^{\text{up}}$, PTEL-based generative models typically use gradient sanitization approach (Chen et al., 2020), which proceeds in two steps as follows:

(1) *Training teacher ensemble from disjoint data subsets.* The private dataset $\mathcal{D}$ is first divided into n_T disjoint subsets $\{\mathcal{D}_i\}_{i=1}^{n_T}$, where each subset $\mathcal{D}_i$ is uniquely used to train a teacher model T_i with parameter θ_{T_i}. Thus, total n_T teachers are trained and form a teacher ensemble $T = \{T_i\}_{i=1}^{n_T}$.

(2) *Training generator by querying teacher ensemble.* The target generator G with the parameter θ_G is trained by interacting exclusively with the teacher ensemble T, without accessing the original dataset $\mathcal{D}$. Given a random input noise $\mathbf{z}$, the generator G generates an output $G(\mathbf{z})$ and queries the teacher ensemble T with $G(\mathbf{z})$. The teachers in T vote on the gradient $\boldsymbol{g}_G^{\text{up}}$ for the query input $G(\mathbf{z})$, resulting in a vote count $\mathbf{n}(G(\mathbf{z}))$. This vote count is processed by the PTEL mechanism $\mathcal{M}$ (e.g., GNMax aggregator in Sec. 4.2) to finally produce a DP-sanitized gradient $\tilde{\boldsymbol{g}}_G^{\text{up}}$. The generator G updates its parameter θ_G using $\tilde{\boldsymbol{g}}_G^{\text{up}}$.

Thus, given a total N training iterations and a PTEL mechanism $\mathcal{M}$ with (ε, δ)-DP guarantee, total DP guarantee of the training algorithm $\mathcal{A}$ can be computed via composition theorem of DP (Dwork et al., 2006b).

Sensitivity preservation We now provide theoretical proof of how training with PFGuard preserves the original privacy analysis of an arbitrary PTEL-generative model G. As explained above, the privacy analysis of PTEL-generative models depends on a total number of training iterations N and (ε, δ)-DP guarantee of the given PTEL mechanism $\mathcal{M}$. Here, we focus more on theoretical DP guarantee than in practice, where we assume N is preserved with fair sampling algorithm $Sample(\cdot)$ used in PFGuard. We thus show how $\mathcal{M}$ and $\mathcal{M} \circ Sample$ result in the same (ε, δ)-DP guarantee, which leads to the same total DP guarantee of the algorithm. In particular, we show how $\mathcal{M}$ and

$\mathcal{M} \circ Sample$ result in the same sensitivity (Def. 2.2), which ensures the same (ε, δ) values given the same amount of DP noise.

Since PTEL mechanism $\mathcal{M}$ operates on a vote count $\mathbf{n}(\bar{\mathbf{x}})$ from the teacher ensemble T given a generated sample as a query input (i.e., $\bar{\mathbf{x}} = G(\mathbf{z})$), we can denote $\mathbf{n}(\bar{\mathbf{x}}) = \mathbf{n}(\bar{\mathbf{x}}; T)$. Let $\mathcal{A}_T$ be the training algorithm of T, where $\mathcal{A}_T(\mathcal{D}_i) = T_i$ and $\mathcal{A}_T(\mathcal{D}) = T = \{T_i\}_{i=1}^{n_T}$ due to data disjointness used in training of n_T teachers. Since $Sample(\cdot)$ samples a subset of dataset $\mathcal{B}_i$ from $\mathcal{D}_i$, data disjointness is preserved in $\mathcal{B}_i$, and thus we derive $\mathcal{A}_T(Sample(\mathcal{D}_i)) = \mathcal{A}_T(\mathcal{B}_i) = T_{s,i}$ and $\mathcal{A}_T(Sample(\mathcal{D})) = T_s = \{T_{s,i}\}_{i=1}^{n_T}$.

Let the original sensitivity value over $\mathbf{n}(\bar{\mathbf{x}}; T_s) = \mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(\mathcal{D}))$ is k. Then, the following holds:

$$\Delta^2 = \max_{\mathcal{D},\mathcal{D}'} \|\mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(\mathcal{D})) - \mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(\mathcal{D}'))\|_2 \tag{12}$$

$$= \max_{\mathcal{D},\mathcal{D}'} \|\mathbf{n}(\bar{\mathbf{x}}; T_1, ..., T_{n_T-1}, T_{n_T}) - \mathbf{n}(\bar{\mathbf{x}}; T_1, ..., T_{n_T-1}, T'_{n_T}))\|_2 \tag{13}$$

$$= \max_{\mathcal{D},\mathcal{D}'} \|(n_1, \ldots, n_c) - (n'_1, \ldots, n'_c)\|_2 \tag{14}$$

$$= \max_{n_i, n'_i, n_j, n'_j} \|(0, \ldots, 0, n_i - n'_i, 0, \ldots, 0, n_j - n'_j, 0, \ldots, 0)\|_2 \tag{15}$$

$$= \max_{n_i, n'_i, n_j, n'_j} \sqrt{(n_i - n'_i)^2 + (n_j - n'_j)^2} \leq k \tag{16}$$

where the second equality is due to data disjointness (i.e., one data can affect a particular teacher that receives the data partition including the data sample), and T_{n_T} denotes the affected teacher without loss of generality. The last equality denotes how much one teacher can maximally contribute to the voting scheme of PTEL, captured by the sensitivity value.

We now compute the sensitivity value over $\mathbf{n}(\bar{\mathbf{x}}; T) = \mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(Sample(\mathcal{D})))$ as follows:

$$\Delta^2 = \max_{\mathcal{D},\mathcal{D}'} \|\mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(Sample(\mathcal{D}))) - \mathbf{n}(\bar{\mathbf{x}}; \mathcal{A}_T(Sample(\mathcal{D}')))\|_2 \tag{17}$$

$$= \max_{\mathcal{D},\mathcal{D}'} \|\mathbf{n}(\bar{\mathbf{x}}; T_{s,1}, ..., T_{s,n_T-1}, T_{s,n_T}) - \mathbf{n}(\bar{\mathbf{x}}; T_{s,1}, ..., T_{s,n_T-1}, T'_{s,n_T}))\|_2 \tag{18}$$

$$= \max_{\mathcal{D},\mathcal{D}'} \|(v_1, \ldots, v_c) - (v'_1, \ldots, v'_c)\|_2 \tag{19}$$

$$= \max_{v_i, v'_i, v_j, v'_j} \|(0, \ldots, 0, v_i - v'_i, 0, \ldots, 0, v_j - v'_j, 0, \ldots, 0)\|_2 \tag{20}$$

$$= \max_{v_i, v'_i, v_j, v'_j} \sqrt{(v_i - v'_i)^2 + (v_j - v'_j)^2} \leq k \tag{21}$$

Here, note that 1) only one particular teacher T_{s,n_T} is affected without loss of generality, and 2) last equality again captures how much one teacher can maximally contribute, which is a property of PTEL's voting scheme and is independent of the sampling algorithm. Thus, using the sampling technique in PFGuard results in a different teacher ensemble compared to the original PTEL-based generative model, but preserves sensitivity value by maintaining data disjointness. We note that this sensitivity remains valid when $\mathcal{B}$ includes duplicate samples due to potential oversampling.

C.2 Training Algorithm

Continuing from Sec. 4.2, we provide the pseudocode to describe the full training algorithm when PFGuard is integrated on top of a PTEL-based generative model.

Notably, PFGuard requires to only modify the minibatch sampling process for training the teacher models (e.g., line 5 in the above pseudocode) to enable fair training within the training algorithm of PTEL-based generative models. PFGuard also preserves their privacy analyses (e.g., privacy cost ε of each training epoch), as long as these models rely on data disjointness to derive sensitivity.

C.3 Extension

Continuing from Sec. 4.2, we provide more details on the extensibility of PFGuard, supporting 1) scenarios with unknown sensitive attribute labels, and 2) integration with existing fairness techniques.

Algorithm 1 Integrating PFGuard with PTEL-based generative models

Input Training dataset $\mathcal{D}$, ensemble of teacher model $T = \{T_i\}_{i=1}^{n_T}$ with each parameters θ_{T_i}, batch size B, PTEL mechanism $\mathcal{M}(\mathbf{x}; T)$, teacher loss function $\mathcal{L}_T$, generator loss function $\mathcal{L}_G$
Output Differentially private generator G with parameters θ_G, total privacy cost ε

Divide the dataset $\mathcal{D}$ into subsets $\{\mathcal{D}_i\}_{i=1}^{n_T}$
for each training epoch **do**
    ///Phase 1: Fair Training
    for each teacher model T_i **do**
        Draw a minibatch $\{\mathbf{x}_i\}_{i=1}^{B} \subseteq \mathcal{D}_i$ with sampling ratio $w(\mathbf{x}) \propto h(\mathbf{x})$ using Eq. 5
        Draw a set of random noise $\{\mathbf{z}_i\}_{i=1}^{B}$ from input random noise distribution p_z of G
        Update teacher model T_i with $\mathcal{L}_{\mathcal{T}}(\theta_{T_i}; \mathbf{x}, G(\mathbf{z}; \theta_G))$
    end for
    ///Phase 2: Private Training
    Draw a set of random noise $\{\mathbf{z}_i\}_{i=1}^{B}$ from input random noise distribution p_z of G
    Generate synthetic data samples $G(\mathbf{z}; \theta_G)$
    Aggregate teacher output with PTEL mechanism $\tilde{o} \leftarrow \mathcal{M}(G(\mathbf{z}; \theta_G); T)$
    where the voting is on gradient of $\mathcal{L}_G(\theta_G)$
    Update generator model G with $\tilde{o}$
    Accumulate privacy cost ε
end for
return Generator G, privacy cost ε

Setting Number of Teachers without Sensitive Attribute Labels We discuss how to extend the proposed upper bound on the number of teachers (i.e., $\lfloor |\mathcal{D}| \min_{s\in\mathcal{S}} p_{\text{bias}}(s) \rfloor$) in settings where the label of sensitive attribute $\mathbf{s}$ is unavailable. The proposed upper bound does not rely on full knowledge of p_{bias}, but the distribution w.r.t. sensitive attributes. Thus, given the training data, we can estimate the subgroup distributions using traditional techniques like K-means clustering (Macqueen, 1967) or random subset labeling (Forestier & Wemmert, 2016). We note that these estimations can be effective, but may introduce some errors or additional overhead, such as increased computational time.

Integration with other fairness techniques PFGuard supports integrating existing methods for other fairness metrics, such as Rawlsian Max-Min fairness while preserving privacy guarantee if they meet two conditions: 1) applied to teacher models to avoid direct impact on the target generator, and 2) maintain data disjointness where one sample affects only one teacher, which is a foundation of privacy guarantee of PTEL (detailed in Sec. 4.2).

D Experimental Settings

Continuing from Sec. 5, we provide more details on experiment settings. In all experiments, we use PyTorch and perform experiments using NVIDIA Quadro RTX 8000 GPUs. Also, we repeat all experiments 10 times and report the mean and standard deviation of the top 3 results. The reason we report the top-3 results is to favor the simple privacy-fairness baselines (e.g., "Reweighting" in Table 3), which tend to fail frequently. We compare their best performances with PFGuard.

D.1 Datasets and Bias Settings

Continuing from Sec. 5, we provide more details on datasets. We use three datasets: MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), and CelebA (Liu et al., 2015). *MNIST and FashionMNIST* contain grayscale images with 28 x 28 pixels and 10 classes. Both datasets have 60,000 training examples and 10,000 testing examples. *CelebA* contains 202,599 celebrity face images. We use the official preprocessed version with face alignment and follow the official training and testing partition (Liu et al., 2015). Note that we are using image datasets instead of the traditional smaller tabular benchmarks for fairness because our goal is to make PFGuard work on higher dimensional data such as images.

MNIST & FashionMNIST We create four bias scenarios across classes and subgroups as follows.

- *Binary Class Bias*. For MNIST, we set digit "3" as the majority class $Y = 1$ and "1" as the minority class $Y = 0$; for FasionMNIST, we set "Sneakers" as $Y = 1$ and "Trousers" as $Y = 0$. For each class pair, we select two classes that share the fewest false negatives and thus can be considered independent, following the convention of prior approaches (Bagdasaryan et al., 2019; Farrand et al., 2020; Ganev et al., 2022). We set *bias level* as 2, meaning the minority class $Y = 0$ is 2 times smaller than the majority class $Y = 1$. After creating bias, we apply random affine transformations to augment the datasets, ensuring they match the original dataset size.
- *Multi-class Bias.* We set "8" as the minority class $Y = 0$, reducing its size while maintaining the size of other 9 classes, following the above prior approaches. We vary the bias level from 1 to 10.
- *Subgroup Bias.* For both MNIST and FashionMNIST datasets, we use image rotation to define subgroups. We set non-rotated images as the majority group $S = 1$ and rotated image as the minority group $S = 0$. We also considered other options including adding lines and changing colors, but we observed that the other options often show the adverse affect of making the images noisier and thus reducing the model accuracy unnecessarily. The rotation also allows for simple and effective verification of subgroup labels in generated synthetic data by comparing the mean values of synthetic image vectors to the centroids of real image vectors. To validate this heuristic, we compared the results with 400 manually labeled images from each baseline model and observed high accuracy (e.g., 96.5% for MNIST).
- *Unknown Subgroup Bias.* In the previous subgroup bias setting, S labels are not used during model training; they are only used for evaluation purposes after training.

CelebA We create binary class bias using gender attributes, where we set female and male images as $Y = 1$ and $Y = 0$, respectively. As discussed in the main text, DP generative models often show low performance on CelebA in challenging bias scenarios like multi-class bias, which can hinder the reliability of fairness analyses (e.g., a random generator achieves perfect fairness by outputting random images regardless of data groups). Notably, we show that DP generative models can produce highly biased synthetic data even in this simple binary class bias setting (Table 4).

D.2 BASELINES

Continuing from Sec. 5, we provide more details on baseline approaches used in our experiments.

DP Generative Models We use three state-of-the-art PTEL-based generative models: GS-WGAN (Chen et al., 2020), G-PATE (Long et al., 2021), and DataLens (Wang et al., 2021a). For all models, we use their official Github codes to implement their models and to use their best-performing hyperparameters for MNIST, FashionMNIST and CelebA.

- *GS-WGAN*. GS-WGAN is extensively used in our experiments, as it leverages both PTEL and DP-SGD (Abadi et al., 2016) to ensure DP and thus allows various integration with other techniques. Their approach first trains a multiple teacher ensemble and considers one teacher as the representative of the other teachers. The output of representative teacher (i.e., gradients) is then sanitized with a DP-SGD based mechanism to train a DP generator. Compared to DP-SGD which operates on the whole minibatch, the DP mechanism in GS-WGAN operates on each data sample and thus can be considered as a composition of B Gaussian mechanism where B is the minibatch size. Our fair sampling preserves their sensitivity analyses despite the potential oversampling as it does not change the sensitivity of each Gaussian mechanism on one input data (i.e, $2C$ due to triangle inequality).
- *G-PATE and DataLens.* G-PATE and DataLens leverage teachers' votes on intermediate gradients to update the generator. To generate histograms of teachers' votes and sanitize with DP mechanisms, G-PATE uses random projection and gradient discretization while DataLens uses a top-k stochastic sign quantization of the gradients. Our fair sampling preserves their sensitivity analyses despite the potential oversampling as each teacher still throws only one vote.

Privacy-Fairness Approaches We use a prominent fair training approach based on reweighting (Choi et al., 2020), and use two recent classification techniques which address both privacy and fairness: DP-SGD-F (Xu et al., 2020) and DPSGD-Global-Adapt (Esipova et al., 2022).

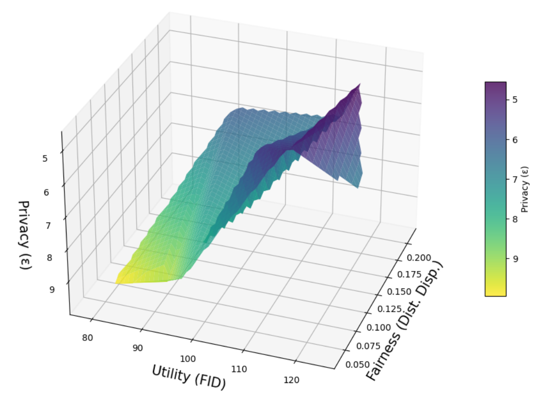

Figure 6: Visualization of Pareto frontier results of PF-Guard. A darker color indicates a more stronger privacy constraint. Both fairness metric (Distributon disparity) and utility metric (Overall FID) are lower the better.

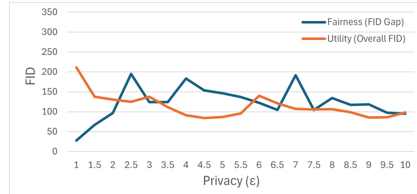

Figure 7: Fairness-utility tradeoff of GS-WGAN (i.e., private generative model).

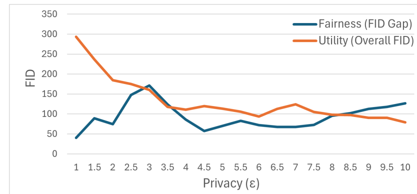

Figure 8: Fairness-utility tradeoff of GS-WGAN when trained with PFGuard.

- *Reweighting*. As outlined in Sec. 3, the reweighting approach modifies the loss term of a generative model to achieve fairness. We use likelihood ratio computed in each bias setting as the reweighting factor and only modifies the loss term of a discriminator, following their approach. When computing likelihood ratio, we directly compute the value as in Eq. 5 using sensitive group labels for subgroup bias setting; we estimate the value using binary classification approach, implemented in their official Github code for unknown subgroup setting. We note that a public reference dataset is required in the estimation process to effectively train a binary classifier.
- *DP-SGD-F and DPSGD-Adapt-Global* DP-SGD-F and DPSGD-Adapt-Global are both fair variants of DP-SGD, where clipping bounds are dynamically adjusted to control the fairness-utility tradeoff. To prevent excessive gradient clipping for minority data group samples, DP-SGD-F employs a groupwise clipping approach where each data group has its own clipping bound, while DPSGD-Adapt employs a scaling approach where all per-sample gradients are scaled down depending on a dynamically adjusted scaling factor. As DP-SGD-F and DPSGD-Adapt-Global do not provide the official codes to our knowledge, we faithfully implemented each algorithm based on their papers. We note that DP-SGD-F is not applicable in the unknown subgroup setting as they require number of group samples present in the batch to compute clipping bounds for each group; DPSGD-Global-Adapt is applicable as the scaling factor does not require knowledge on group labels.

E Additional Experiments

Continuing from Sec. 5, Sec. 5.1, Sec. 5.3, and Sec. 5.4, we provide more experimental results.

E.1 More experiments on Privacy-Fairness-Utility Tradeoff

Continuing from Sec. 5.2, we provide more results on privacy-fairness-utility tradeoff.

Fairness-Utility Tradeoff with Varying Privacy Level We analyze the fairness-utility tradeoff under varying privacy levels, following (Tran et al., 2021b). Specifically, we examine how training with PFGuard impacts the original fairness-utility tradeoff of private generative models, using the MNIST dataset under the subgroup bias setting using. The results are shown in Fig. 7 and Fig. 8. In the weaker privacy regimes (i.e., higher values of ε), both the privacy-only generative model and PFGuard converge to similar performance levels. However, PFGuard demonstrates notable fairness improvements in the stronger privacy regime, albeit showing utility degradation due to slower convergence in the early training stages. We suspect this slower convergence is a natural consequence of learning a more balanced distribution since the generative model should capture more diverse input features at the early training stage.

Table 5: Comparison of privacy-fairness-utility performance on Adult under ε=1, using PATE-GAN as the base DP generator (see Sec. D.2 for more details). The first five rows represent upper bound performances for vanilla, DP-only, and fair-only models. Evaluations cover subgroup bias. Lower values are better across all metrics except AUROC. We boldface the best results and underline the second best results.

	Privacy	Fairness		Utility
Method	ε	EO Disp. (↓)	Dem. Disp. (↓)	AUROC (↑)
Vanila	✗	0.56	0.58	**0.80**
Fair-only	✗	**0.07**	**0.07**	0.75
DP-only (DP-WGAN)	✓	0.31	0.30	0.69
DP-only (PATE-GAN)	✓	0.19	0.22	0.74
DP-only (RON-Gauss)	✓	0.18	0.14	0.70
FFPDG	✓	0.12	0.20	0.75
PFGuard	✓	<u>0.08</u>	<u>0.12</u>	<u>0.76</u>

Pareto Frontier Results Fig. 6 visualizes the Pareto frontier results of PFGuard, showing the privacy-fairness-utility tradeoff. The non-linear surface highlights the intricate relationship between these objectives. Here, lower values for privacy, utility, and fairness metrics indicate better performances. As privacy constraints weaken (i.e., the light-colored region), both utility and fairness improve, converging toward a more favorable region. However, as privacy constraints become stronger, the utility narrows down to a specific range, while fairness greatly varies, which is particularly evident in the purple-colored region. This suggests that stronger DP noise consistently degrades image quality but results in highly variable fairness outcomes, which implies there can be sweet-spot regions that can achieve both fairness and utility.

E.2 Experiment results on Tabluar Data

Continuing from Sec. 5.3, we provide additional results on the tabular dataset. While one of the key contributions of PFGuard is scalability to high-dimensional data such as images, we show how PFGuard can support tabular data as well as images. Table 5 shows that PFGuard achieves fairness comparable to fair-only generative models only showing a slight decline in overall utility, demonstrating effectiveness in fair and private tabular data generation.

E.3 Comparison of Computational Time

Continuing from Sec. 5.3, we compare the computational time when integrating PFGuard with existing PTEL-based generative models. Table shows that PFGuard incurs minimal overhead in computational time ($< 4\%$), due to the simple modification in minibatch sampling for fairness.

Table 6: Comparison of computational time of private generative models with and without PFGuard.

	MNIST			FashionMNIST		
Method	w/o PFGuard	w/ PFGuard	Overhead (%)	w/o PFGuard	w/ PFGuard	Overhead (%)
GS-WGAN	7378.30	7467.48	1.21	8114.96	8392.58	3.42
G-PATE	30810.25	31852.11	3.38	25238.84	26317.07	3.56
DataLens	41590.34	41638.19	0.12	547740.47	55714.41	1.78

E.4 Experimental Results on FashionMNIST

Continuing from Sec. 5.3, we show the results of the analysis in synthetic data (Table 7) and downstream tasks (Table 8) evaluated on FashionMNIST. Compared to the results evaluated on MNIST, private generative models often generate more imbalanced synthetic data w.r.t. sensitive groups and exhibit lower overall image quality. In comparison, PFGuard consistently improves both fairness and overall utility in most cases, similar to the results observed in the MNIST evaluation.

Table 7: Fairness and utility performances of private generative models with and without PFGuard on synthetic data, evaluated on FashionMNIST with subgroup bias under $\varepsilon = 10$. Blue and red arrows indicate positive and negative changes, respectively. Lower values are better across all metrics.

	Fairness		Utility				
Method	KL (↓)	Dist. Disp. (↓)	FID (↓)	Y=1, S=1	Y=1, S=0	Y=0, S=1	Y=0, S=0
GS-WGAN	0.558$\pm$0.147	0.651$\pm$0.007	124.85$\pm$0.00	**130.95$\pm$0.00**	278.06$\pm$0.00	155.36$\pm$0.00	217.00$\pm$0.00
G-PATE	0.270$\pm$0.026	0.494$\pm$0.021	245.13$\pm$24.85	271.28$\pm$15.32	249.66$\pm$10.95	275.61$\pm$38.26	282.58$\pm$30.95
DataLens	0.160$\pm$0.022	0.388$\pm$0.026	165.90$\pm$6.50	197.61$\pm$8.90	191.72$\pm$8.46	173.60$\pm$9.00	225.93$\pm$6.53
GS-WGAN + PFGuard	**0.009$\pm$0.065** (↓)	**0.065$\pm$0.049** (↓)	**113.13$\pm$7.24** (↓)	149.54$\pm$3.96	**166.69$\pm$10.00**	**114.87$\pm$12.26**	**146.67$\pm$22.52**
G-PATE + PFGuard	0.190$\pm$0.050 (↓)	0.418$\pm$0.049 (↓)	242.20$\pm$42.70 (↓)	267.14$\pm$33.95	248.92$\pm$51.93	266.91$\pm$51.47	295.32$\pm$31.69
DataLens + PFGuard	0.127 $\pm$0.037 (↓)	0.345$\pm$0.050 (↓)	209.48 $\pm$12.01 (↑)	248.43$\pm$13.12	222.16$\pm$16.69	222.46$\pm$15.17	262.62$\pm$11.37

Table 8: Fairness and utility performances of private generative models with and without PFGuard on downstream tasks, evaluated on FashionMNIST with subgroup bias under $\varepsilon = 10$. Blue and red arrows indicate positive and negative changes, respectively.

	MLP			CNN		
	Fairness		Utility	Fairness		Utility
Method	EO Disp. (↓)	Dem. Disp. (↓)	Acc (↑)	EO Disp. (↓)	Dem. Disp. (↓)	Acc (↑)
GS-WGAN	0.773$\pm$0.019	**0.021$\pm$0.019**	0.812$\pm$0.009	0.795$\pm$0.008	**0.007$\pm$0.007**	0.804$\pm$0.003
G-PATE	0.636$\pm$0.065	0.162$\pm$0.059	0.875$\pm$0.004	0.525$\pm$0.056	0.095$\pm$0.064	0.884$\pm$0.010
DataLens	0.484$\pm$0.168	0.203$\pm$0.092	**0.901$\pm$0.030**	**0.328$\pm$0.039**	0.072$\pm$0.045	**0.925$\pm$0.009**
GS-WGAN + PFGuard	**0.296$\pm$0.099** (↓)	0.152 $\pm$0.033 (↑)	0.884 $\pm$0.015 (↑)	0.449 $\pm$0.082 (↓)	0.203 $\pm$0.037 (↑)	0.910 $\pm$0.011 (↑)
G-PATE + PFGuard	0.556 $\pm$0.152 (↓)	0.154 $\pm$0.087 (↓)	0.885 $\pm$0.013 (↑)	0.476 $\pm$0.051 (↓)	0.124 $\pm$0.041 (↑)	0.899 $\pm$0.017 (↑)
DataLens + PFGuard	0.387 $\pm$0.154 (↓)	0.153 $\pm$0.103 (↓)	0.858 $\pm$0.025 (↓)	0.394 $\pm$0.109 (↑)	0.093 $\pm$0.074 (↑)	0.877 $\pm$0.025 (↓)

E.5 Additional Normalization Technique for Faster Convergence

Continuing from Sec. 5.3, we investigate the impact of the normalization factor on the overall image quality of PFGuard. While we use a traditional normalization factor $N_1 = \sum_i h(x_i)$ for $w(x_i) \propto h(x_i)$ in our SIR-based sampling algorithm, we can employ additional normalization techniques to boost the performance. For example, we can use $N_2 = \sum_i h(x_i)/N_{-i}$ for $w(x_i) \propto h(x_i)/N_{-i}$, where $N_{-i} = \sum_i h(x_i) - h(x_i)$, which is known to help faster convergence of SIR algorithms to the target distribution p_{bal} (Skare et al., 2003).

We thus compare the overall image quality resulting from two different normalization options, N_1 and N_2, varying batch sizes to analyze their effects on model convergence. We note that using a larger batch size can change the DP analysis (i.e., the more a model uses the data, the weaker privacy it provides). To effectively compare the difference in convergence speed, we create both binary class bias and subgroup bias on the MNIST dataset, where class $Y = 0$ is 3 times smaller than class $Y = 1$, and group $S = 0$ is 3 times smaller than group $S = 1$.

Table 9 shows the comparison of image quality using the MNIST dataset, measuring image quality with FID, where a lower value is better. While both N_1 and N_2 demonstrate comparable performance when using a large batch size, the performance gap becomes more evident as the batch size decreases. This empirical evidence shows that the performance of PFGuard can be further improved by additionally employing various normalization techniques.

Table 9: Influence of normalization factor on MNIST with corresponding DP guarantees (ε) to different batch sizes. GS-WGAN is used as the base DP generator.

	Normalization factor	
Batch size	N_1	N_2
128 ($\varepsilon = 29.91$)	75.05$\pm$2.26	74.58$\pm$2.96
64 ($\varepsilon = 19.58$)	75.35$\pm$6.67	72.20$\pm$4.58
32 ($\varepsilon = 9.99$)	82.68$\pm$7.13	78.18$\pm$1.85

E.6 Full Results with Standard Deviation

Continuing from Sec. 5.2 and Sec. 5.4, we show full results with standard deviation. Table 10 and Table 11 shows the full results of Table 4.

Table 10: Full results of fairness and utility performances of private generative models with and without PFGuard on downstream tasks, evaluated on CelebA(S) with binary class bias under $\varepsilon = 1$. GS-WGAN is excluded due to lower image quality in this setting. Blue and red arrows indicate positive and negative changes, respectively.

	Fairness	Utility		
Method	Acc. Disp. (↓)	Acc (↑)	Acc (Y=1)	Acc (Y=0)
G-PATE	0.978±0.024	0.666±0.003	0.014±0.014	**0.992±0.010**
DataLens	0.793±0.173	0.643±0.031	0.114±0.087	0.907±0.087
G-PATE + PFGuard	0.736 ±0.126 (↓)	0.678±0.003 (↑)	0.187 ±0.085 (↑)	0.923 ±0.041 (↓)
DataLens + PFGuard	**0.725±0.055** (↓)	**0.689±0.004** (↑)	**0.205±0.040** (↑)	0.931 ±0.015 (↑)

Table 11: Full results of fairness and utility performances of private generative models with and without PFGuard on downstream tasks, evaluated on CelebA(L) with binary class bias under $\varepsilon = 1$. GS-WGAN is excluded due to lower image quality in this setting. Blue and red arrows indicate positive and negative changes, respectively.

	Fairness	Utility		
Method	Acc. Disp. (↓)	Acc (↑)	Acc (Y=1)	Acc (Y=0)
G-PATE	0.968±0.025	0.668±0.001	0.023±0.018	**0.991±0.007**
DataLens	0.678±0.027	0.686±0.011	0.234±0.028	0.912±0.005
G-PATE + PFGuard	**0.277±0.314** (↓)	0.563±0.028 (↓)	**0.378±0.181** (↑)	0.655±0.133 (↓)
DataLens + PFGuard	0.641±0.038 (↓)	**0.704±0.007** (↑)	0.276 ±0.031 (↑)	0.917 ±0.008 (↑)

F Related Work

Continuing from Sec. 6, we present more related work.

Private-only Data Generation Most privacy-preserving data generation techniques focus on satisfying differential privacy (DP) (Dwork et al., 2014). The majority of these techniques use Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) with DP training techniques, although privatizing other generative models have recently been proposed as well (Takagi et al., 2021; Cao et al., 2021; Harder et al., 2021; Liew et al., 2021; Chen et al., 2022; Vinaroz et al., 2022; Yang et al., 2023; Ghalebikesabi et al., 2023). One approach is based on DP-SGD (Abadi et al., 2016), which is a DP-enabled standard stochastic gradient descent algorithm to train ML models (Xie et al., 2018; Zhang et al., 2018; Torkzadehmahani et al., 2019; Bie et al., 2023). Another approach is based on the Private Aggregation of Teacher Ensembles (PATE) framework (Papernot et al., 2016; 2018), which trains multiple models on private data, and updates the generator with differentially private aggregation of multiple model outcomes (Jordon et al., 2018; Long et al., 2021; Wang et al., 2021a). GS-WGAN (Chen et al., 2020) is a state-of-the-art GAN-based technique that combines DP-SGD and PATE where multiple models are trained as in PATE, but their outcomes are processed with the Gaussian mechanism to update the generator as in DP-SGD. In comparison, PFGuard complements private GANs by also improving the fairness of data generation.

Fair-only Data Generation The goal of model fairness is to avoid discriminating against certain demographics (Barocas et al., 2017; Feldman et al., 2015; Hardt et al., 2016), and fair data generation solves this problem by generating synthetic data to remove data bias. The main approaches of fair data generation are as follows: 1) modifying training objectives to balance model training (Xu et al., 2018; Sattigeri et al., 2019; Yu et al., 2020; Choi et al., 2020; Teo et al., 2023) and 2) modifying latent distributions of the input noise to obtain fairer outputs (Tan et al., 2020; Humayun et al., 2021).

In comparison, PFGuard modifies sampling procedures to balance model training while preserving original training objectives and makes the key contribution of satisfying both privacy and fairness. There is another recent line of work using generated data together with original training data for model fairness (Roh et al., 2023; Zietlow et al., 2022), but they focus on classification tasks and assume to use given generative models.

Privacy-Fairness Intersection Recent studies have shown that achieving DP can hurt model fairness in classification tasks (Bagdasaryan et al., 2019; Farrand et al., 2020; Xu et al., 2020; Esipova et al., 2022), decision-making processes (Pujol et al., 2020), and even in generation tasks (Cheng et al., 2021; Ganev et al., 2022; Bullwinkel et al., 2022; Rosenblatt et al., 2024). In addition, there is another notable line of work to investigate privacy-fairness-utility tradeoff, showing that achieving both privacy and fairness will necessarily sacrifice utility (Cummings et al., 2019; Agarwal, 2021; Sanyal et al., 2022). In comparison, our study uncovers the counteractive nature of privacy and fairness, which can result in compromised privacy and fairness, and ultimately compromised utility. To effectively achieve both privacy and fairness in model training, various techniques have been developed for classification tasks (Jagielski et al., 2019; Xu et al., 2019; 2020; Tran et al., 2022; Esipova et al., 2022; Kulynych et al., 2022; Yaghini et al., 2023; Lowy et al., 2023). In comparison, PFGuard focuses on data generation tasks, specifically tailoring its fair training phase to generative modeling objectives, which is learning the underlying training data distributions to generate synthetic data. Among the classification techniques, we make comparisons with studies that also leverage 1) importance sampling or 2) PTEL, which are the two key components of PFGuard. (Kulynych et al., 2022) also extends importance sampling to private settings, but focuses on fairness in classification accuracy. In contrast, PFGuard focuses exclusively on private settings, covering fairness in both data generation and classification. (Yaghini et al., 2023) and (Tran et al., 2022) use PTEL, but rely on public datasets to train student classifiers. In contrast, PFGuard eliminates the need for public datasets by making PTEL queries using generated samples from the student generator. We finally note that (Lowy et al., 2023) introduces the first DP fair learning method with convergence guarantees for empirical risk minimization. In contrast, PFGuard provides convergence guarantees for fair generative modeling.