Private Post-GAN Boosting

Marcel Neunhoeffer
University of Mannheim
Mannheim, Germany
mneunhoe@uni-mannheim.de

Zhiwei Steven Wu
University of Minnesota
Minneapolis, USA

Cynthia Dwork
Harvard University
Cambridge, USA

Abstract

Differentially private GANs have proven to be a promising approach for generating realistic synthetic data without compromising the privacy of individuals. However, due to the privacy-protective noise introduced in the training, the convergence of GANs becomes even more elusive, which often leads to poor utility in the output generator at the end of training. We propose *Private post-GAN boosting (Private PGB)*, a differentially private method that combines samples produced by the sequence of generators obtained during GAN training to create a high-quality synthetic dataset. Our method leverages the Private Multiplicative Weights method (Hardt and Rothblum, 2010) and the discriminator rejection sampling technique (Azadi *et al.*, 2019) for reweighting generated samples, to obtain high quality synthetic data even in cases where GAN training does not converge. We evaluate Private PGB on a Gaussian mixture dataset and two US Census datasets, and demonstrate that Private PGB improves upon the standard private GAN approach across a collection of quality measures. Finally, we provide a non-private variant of PGB that improves the data quality of standard GAN training.

1 Introduction

The vast collection of detailed personal data, including everything from medical history to voting records, to GPS traces, to online behavior, promises to enable researchers from many disciplines to conduct insightful data analyses. However, many of these datasets contain sensitive personal information, and there is a growing tension between data analyses and data privacy. To protect the privacy of individual citizens, many organizations, including Google [15], Microsoft [13], Apple [12], and more recently the 2020 US Census [2], have adopted *differential privacy* [14] as a mathematically rigorous privacy measure. However, working with noisy statistics released under differential privacy requires training.

A natural and promising approach to tackle this challenge is to release *differentially private synthetic data*—a privatized version of the dataset that consists of fake data records and that approximates the real dataset on important statistical properties of interest. Since they already satisfy differential privacy, synthetic data enable researchers to interact with the data freely and to perform the same analyses even without expertise in differential privacy. A recent line of work [8, 36, 37] studies how one can generate synthetic data by incorporating differential privacy into *generative adversarial networks* (GANs) [18]. Although GANs provide a powerful framework for synthetic data, they are also notoriously hard to train and privacy constraint imposes even more difficulty. Due to the added noise in the private gradient updates, it is often difficult to reach convergence with private training.

In this paper, we study how to improve the quality of the synthetic data produced by private GANs. Unlike much of the prior work that focuses on fine-tuning of network architectures and training techniques, we propose *Private post-GAN boosting* (Private PGB)—a differentially private method that boosts the quality of the generated samples after the training of a GAN. Our method can be viewed as a simple and practical amplification scheme that improves the distribution from any existing

Preprint. Under review.

black-box GAN training method – private or not. We take inspiration from an empirical observation in [8] that even though the generator distribution at the end of the private training may be a poor approximation to the data distribution (due to e.g. mode collapse), there may exist a high-quality mixture distribution that is given by several generators over different training epochs. PGB is a principled method for finding such a mixture at a moderate privacy cost and without any modification of the GAN training procedure.

To derive PGB, we first formulate a two-player zero-sum game, called *post-GAN* zero-sum game, between a *synthetic data* player, who chooses a distribution over generated samples over training epochs to emulate the real dataset, and a *distinguisher* player, who tries to distinguish generated samples from real samples with the set of discriminators over training epochs. We show that under a "support coverage" assumption the synthetic data player's mixed strategy (given by a distribution over the generated samples) at an equilibrium can successfully "fool" the distinguisher–that is, no mixture of discriminators can distinguish the real versus fake examples better than random guessing. While the strict assumption does not always hold in practice, we demonstrate empirically that the synthetic data player's equilibrium mixture consistently improves the GAN distribution.

The Private PGB method then privately computes an approximate equilibrium in the game. The algorithm can be viewed as a computationally efficient variant of MWEM (private multiplicative weights with exponential mechanism) [20, 19], which is an inefficient query release algorithm with near-optimal sample complexity. Since MWEM maintains a distribution over exponentially many "experts" (the set of all possible records in the data domain), it runs in time exponential in the dimension of the data. In contrast, we rely on private GAN to reduce the support to only contain the set of privately generated samples, which makes PGB tractable even for high-dimensional data.

We also provide an extension of the PGB method by incorporating the technique of *discriminator rejection sampling* [7, 35]. We leverage the fact that the distinguisher's equilibrium strategy, which is a mixture of discriminators, can often accurately predict which samples are unlikely and thus can be used as a rejection sampler. This allows us to further improve the PGB distribution with rejection sampling without any additional privacy cost since differential privacy is preserved under post-processing. Our Private PGB method also has a natural non-private variant, which we show improves the GAN training without privacy constraints.

We empirically evaluate both the Private and Non-Private PGB methods on three tasks. To visualize the effects of our methods, we first evaluate our methods on a two-dimensional synthetic dataset that consists of samples drawn from a mixture of 25 Gaussian distributions. We define a relevant quality score function and show that the both Private and Non-Private PGB methods improve the score of the samples generated from GAN. We then focus on two US Census datasets and demonstrate that the PGB method can improve the generator distribution on several statistical measures, including 3-way marginal distributions and pMSE. Finally, we evaluate the PGB methods on a machine learning task with a natural classification task. We train predictive models on samples from PGB and sample from GAN (without PGB), and show that the models trained on synthetic data generated with PGB consistently have higher accuracy on real out-of-sample test data.

Related work. Our PGB method can be viewed as a modular boosting method that can improve on a growing line of work on differentially private GANs [8, 36, 17, 34]. To obtain formal privacy guarantees, these algorithms optimize the discriminators in GAN under differential privacy, by using private SGD, RMSprop, or Adam methods, and track the privacy cost using moments accounting [1, 27]. [37] give a private GAN training method by adapting ideas from the PATE framework [28].

Our PGB method is inspired by the Private Multiplicative Weigths method [20] and its more practical variant MWEM [19], which answer a large collection of statistical queries by releasing a synthetic dataset. Our work also draws upon two recent techniques ([35] and [7]) that use the discriminator as a rejection sampler to improve the generator distribution. We apply their technique by using the mixture discriminator computed in PGB as the rejection sampler. There has also been work that applies the idea of boosting to (non-private) GANs. For example, [5] and [21] propose methods that directly train a mixture of generators and discriminators, and [33] proposes AdaGAN that reweighes the real examples during training similarly to what is done in AdaBoost [16]. Both of these methods may be hard to make differentially private: they either require substantially more privacy budget to train a collection of discriminators or increase the weights on a subset of examples, which requires more adding more noise when computing private gradients. In contrast, our PGB method boosts the generated samples *post* training and does not make modifications to the GAN training procedure.

2 Preliminaries

Let $\mathcal{X}$ denote the data domain of all possible observations in a given context. Let p_d be a distribution over $\mathcal{X}$. We say that two datasets $X, X' \in \mathcal{X}^n$ are adjacent, denoted by $X \sim X'$, if they differ by at most one observation. We will write p_X to denote the empirical distribution over X.

Definition 1 (Differential Privacy (DP) [14])**.** *A randomized algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ with output domain $\mathcal{R}$ (e.g. all generative models) is (ε, δ)-differentially private (DP) if for all adjacent datasets $X, X' \in \mathcal{X}^n$ and for all $S \subseteq \mathcal{R}$: $P(\mathcal{A}(X) \in S) \le e^{\varepsilon} P(\mathcal{A}(X') \in S) + \delta$.*

A very nice property of differential privacy is that it is preserved under post-processing.

Lemma 1 (Post-processing)**.** *Let $\mathcal{M}$ be an (ε, δ)-differentially private algorithm with output range R and $f : R \to R'$ be any mapping, the composition $f \circ \mathcal{M}$ is (ε, δ)-differentially private.*

As a result, any subsequent analyses conducted on differentially private synthetic data also satisfy DP.

The *exponential mechanism* [26] is a private mechanism for selecting among the best of a discrete set of alternatives $\mathcal{R}$, where "best" is defined by a quality function $q\colon \mathcal{X}^n \times \mathcal{R} \to \mathcal{R}$ that measures the quality of the result r for the dataset X. The sensitivity of the quality score q is defined as $\Delta(q) = \max_{r \in \mathcal{R}} \max_{X \sim X'} |q(X, r) - q(X', r)|$. Then given a quality score q and privacy parameter ε, the exponential mechanism $\mathcal{M}_E(q, \varepsilon, X)$ simply samples a random alternative from the range $\mathcal{R}$ such that the probability of selecting each r is proportional to $\exp(\varepsilon q(X, r)/(2\Delta(q)))$.

2.1 Differentially Private GAN

The framework of *generative adversarial networks (GANs)* [18] consists of two types of neural networks: *generators* and *discriminators*. A generator G is a function that maps random vectors $z \in Z$ drawn from a prior distribution p_z to a sample $G(z) \in \mathcal{X}$. A discriminator D takes an observation $x \in \mathcal{X}$ as input and computes a probability $D(x)$ that the observation is real. Each observation is either drawn from the underlying distribution p_d or the induced distribution p_g from a generator. The training of GAN involves solving the following joint optimization over the discriminator and generator:

$$\min_G \max_D \; \mathbb{E}_{x \sim p_X}[f(D(x))] + \mathbb{E}_{z \sim p_z}[f(1 - D(G(z)))]$$

where $f\colon [0,1] \to \mathbb{R}$ is a monotone function. For example, in standard GAN, $f(a) = \log a$, and in Wasserstein GAN [3], $f(a) = a$. The standard (non-private) algorithm iterates between optimizing the parameters of the discriminator and the generator based on the loss functions:

$$L_D = -\mathbb{E}_{x \sim p_X}[f(D(x))] - \mathbb{E}_{z \sim p_z}[f(1 - D(G(z)))], \quad L_G = \mathbb{E}_{z \sim p_z}[f(1 - D(G(z)))]$$

The private algorithm for training GAN also performs the same alternating optimization, but it optimizes the discriminator under differential privacy while keeping the generator optimization the same. In general, the training proceeds over epochs $\tau = 1, \ldots, N$, and at the end of each epoch τ the algorithm obtains a discriminator D_τ and a generator G_τ by optimizing the loss functions respectively. In [8, 36], the private optimization on the discriminators is done by running the private SGD method [1] or its variants. [37] performs the private optimization by incorporating the PATE framework [28]. For all of these private GAN methods, the entire sequence of discriminators $\{D_1, \ldots, D_N\}$ satisfies privacy, and thus the sequence of generators $\{G_1, \ldots, G_N\}$ is also private since they can be viewed as post-processing of the discriminators. Our PGB method is agnostic to the exact private GAN training methods.

3 Private Post-GAN Boosting

The noisy gradient updates impede convergence of the differentially private GAN training algorithm, and the generator obtained in the final epoch of the training procedure may not yield a good approximation to the data distribution. Nonetheless, empirical evidence has shown that a mixture over the set of generators can be a realistic distribution [8]. We now provide a principled and practical scheme for computing such a mixture subject to a moderate privacy budget. Recall that during private GAN training method produces a sequence of generators $\mathcal{G} = \{G_1, \ldots, G_N\}$ and discriminators $\mathcal{D} = \{D_1, \ldots, D_N\}$. Our boosting method computes a weighted mixture of the G_j's and a weighted mixture of the D_j's that improve upon any individual generator and discriminator. We do that by computing an equilibrium of the following *post-GAN (training)* zero-sum game.

3.1 Post-GAN Zero-Sum Game.

We will first draw r independent samples from each generator G_j, and let B be the collection of the rN examples drawn from the set of generators. Consider the following *post-GAN* zero-sum game between a *synthetic data player*, who maintains a distribution ϕ over the data in B to imitate the true data distribution p_X, and a *distinguisher player*, who uses a mixture of discriminators to tell the two distributions ϕ and p_X apart. This zero-sum game is aligned with the minimax game in the original GAN formulation, but is much more tractable since each player has a finite set of strategies. To define the payoff in the game, we will adapt from the Wasserstein GAN objective since it is less sensitive than the standard GAN objective to the change of any single observation (changing any single real example changes the payoff by at most $1/n$), rendering it more compatible with privacy tools. Formally, for any $x \in B$ and any discriminator D_j, define the payoff as

$$U(x, D_j) = \mathbb{E}_{x' \sim p_X}\left[D_j(x')\right] + (1 - D_j(x))$$

For any distribution ϕ over B, let $U(\phi, \cdot) = \mathbb{E}_{x\sim\phi}[U(x, \cdot)]$, and similarly for any distribution ψ over $\{D_1, \ldots, D_N\}$, we will write $U(\cdot, \psi) = \mathbb{E}_{D\sim\psi}[U(\cdot, D)]$. Intuitively, the payoff function U measures the predictive accuracy of the distinguisher in classifying whether the examples are drawn from the synthetic data player's distribution ϕ or the private dataset X. Thus, the synthetic data player aims to minimize U while the distinguisher player aims to maximize U.

Definition 2. *The pair $(\overline{D}, \overline{\phi})$ is an α-approximate equilibrium of the post-GAN game if*

$$\max_{D_j \in \mathcal{D}} U(\overline{\phi}, D_j) \leq U(\overline{\phi}, \overline{D}) + \alpha, \quad \textit{and} \quad \min_{\phi \in \Delta(B)} U(\phi, \overline{D}) \geq U(\overline{\phi}, \overline{D}) - \alpha \tag{1}$$

By von Neumann's minimax theorem, there exists a value V – called the *game value* – such that

$$V = \min_{\phi\in\Delta(B)} \max_{j\in[N]} U(\phi, D_j) = \max_{\psi\in\Delta(\mathcal{D})} \min_{x\in B} U(x, \psi)$$

The game value corresponds to the payoff value at an exact equilibrium of the game (that is $\alpha = 0$). When the set of discriminators cannot predict the real versus fake examples better than random guessing, the game value $V = 1$. We now show that under the assumption that the generated samples in B approximately cover the support of the dataset X, the distinguisher player cannot distinguish the real and fake distributions much better than by random guessing.

Theorem 1. *Fix a private dataset $X \in (\mathbb{R}^d)^n$. Suppose that for every $x \in X$, there exists $x_b \in B$ such that $\|x - x_b\|_2 \leq \gamma$. Suppose $\mathcal{D}$ includes a discriminator network $D^{1/2}$ that outputs $1/2$ for all inputs, and assume that all networks in $\mathcal{D}$ are L-Lipschitz. Then there exists a distribution $\phi \in \Delta(B)$ such that $(\phi, D^{1/2})$ is a $L\gamma$-approximate equilibrium, and so $1 \leq V \leq 1 + L\,\gamma$.*

We defer the proof to the appendix. While the support coverage assumption is strong, we show empirically the synthetic data player's mixture distribution in an approximate equilibrium improves on the distribution given by the last generator G_N even when the assumption does not hold. We now provide a method for computing an approximate equilibrium of the game.

3.2 Boosting via Equilibrium Computation.

Our post-GAN boosting (PGB) method computes an approximate equilibrium of the post-GAN zero-sum game by simulating the so-called *no-regret dynamics*. Over T rounds the synthetic data player maintains a sequence of distributions $\phi^1, \ldots, \phi^T$ over the set B, and the distinguisher plays a sequence of discriminators $D^1, \ldots, D^T$. At each round t, the distinguisher first selects a discriminator D using the exponential mechanism $\mathcal{M}_E$ with the payoff $U(\phi^t, \cdot)$ as the score function. This will find an accurate discriminator D^t against the current synthetic distribution ϕ^t, so that the synthetic data player can improve the distribution. Then the synthetic data player updates its distribution to ϕ^t based on an online no-regret learning algorithm–the multiplicative weights (MW) method [23]. We can view the set of generated examples in B as a set of "experts", and the algorithm maintains a distribution over these experts and, over time, places more weight on the examples that can better "fool" the distinguisher player. To do so, MW updates the weight for each $x \in B$ with

$$\phi^{t+1}(x) \propto \phi^t \exp\left(-\eta U(x, D^t)\right) \propto \exp\left(\eta D^t(x)\right) \tag{2}$$

where η is the learning rate. At the end, the algorithm outputs the average plays $(\overline{D}, \overline{\phi})$ for both players. We will show these form an approximate equilibrium of the post-GAN zero-sum game [16].

Algorithm 1 Differentially Private Post-GAN Boosting

Require: a private dataset $X \in \mathcal{X}^n$, a synthetic dataset B generated by the set of generators $\mathcal{G}$, a collection of discriminators $\{D_1, \dots, D_N\}$, number of iterations T, per-round privacy budget ϵ_0, learning rate parameter η.

Initialize ϕ^1 to be the uniform distribution over B

for $t = 1, \dots, T$ **do**

Distinguisher player: Run exponential mechanism $\mathcal{M}_E$ to select a discriminator D^t using quality score $q(X, D_j) = U(\phi^t, D_j)$ and privacy parameter ϵ_0.

Synthetic data player: Multiplicative weights update on the distribution over B: for each example $b \in B$:

$$\phi^{t+1}(b) \propto \phi^t(b)\, \exp(\eta D^t(b))$$

Let $\overline{D}$ be the discriminator defined by the uniform average over the set $\{D^1, \dots, D^T\}$, and $\overline{\phi}$ be the distribution defined by the average over the set $\{\phi^1, \dots, \phi^T\}$

Note that the synthetic data player's MW update rule does not involve the private dataset, and hence is just a post-processing step of the selected discriminator D^t. Thus, the privacy guarantee follows from the composition of T runs of the exponential mechanism (proof in the appendix).

Theorem 2 (Privacy Guarantee)**.** *For any $\delta \in (0, 1)$, the private MW post-amplification algorithm satisfies (ϵ, δ)-DP with $\epsilon = \sqrt{2\log(1/\delta)T}\epsilon_0 + T\epsilon_0(\exp(\epsilon_0) - 1)$.*

Note that if the private GAN training algorithm satisfies $(\varepsilon_1, \delta_1)$-DP and the Private PGB method satisfies $(\varepsilon_2, \delta_2)$-DP, then the entire procedure is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-DP.

We now show that the pair of average plays form an approximate equilibrium of the game.

Theorem 3 (Approximate Equilibrium)**.** *With probability $1 - \beta$, the pair $(\overline{D}, \overline{\phi})$ is an α-approximate equilibrium of the post-GAN zero-sum game with $\alpha = 4\eta + \frac{\log|B|}{\eta T} + \frac{2\log(NT/\beta)}{n\varepsilon_0}$. If $T \geq n^2$ and $\eta = \frac{1}{2}\sqrt{\log(|B|)/T}$, then*

$$\alpha = O\left(\frac{\log(nN|B|/\beta)}{n\varepsilon_0}\right)$$

We provide a proof sketch here and defer the full proof to the appendix. By the result of [16], if the two players have low regret in the dynamics, then their average plays form an approximate equilibrium, where the regret of the two players is defined as $R_{\text{syn}} = \sum_{t=1}^{T} U(\phi^t, D^t) - \min_{b\in B} \sum_{t=1}^{T} U(b, D^t)$ and $R_{\text{dis}} = \max_{D_j} \sum_{t=1}^{T} U(\phi^t, D_j) - \sum_{t=1}^{T} U(\phi^t, D^t)$. Then approximate equilibrium guarantee directly follows from bounding R_{syn} with the regret bound of MW and R_{dis} with the approximate optimality of the exponential mechanism.

Non-Private PGB. The Private PGB method has a natural non-private variant: in each round, instead of drawing from the exponential mechanism, the distinguisher player will simply compute the exact best response: $D^t = \arg\max_{D_j} U(\phi^t, D_j)$. Then if we set learning rate $\eta = \frac{1}{2}\sqrt{\log(|B|)/T}$ and run for $T = \log(|B|)/\alpha^2$ rounds, the pair $(\overline{D}, \overline{\phi})$ returned is an α-approximate equilibrium.

Extension with Discriminator Rejection Sampling. The mixture discriminator $\overline{D}$ at the equilibrium provides an accurate predictor on which samples are unlikely. As a result, we can use $\overline{D}$ to further improve the data distribution $\overline{\phi}$ by the *discriminator rejection sampling* (DRS) technique of [7]. The DRS scheme in our setting generates a single example as follows: first draw an example x from $\overline{\phi}$ (the proposal distribution), and then accept x with probability proportional to $\overline{D}(x)/(1 - \overline{D}(x))$. Note that the optimal discriminator D^* that distinguishes the distribution $\overline{\phi}$ from true data distribution p_d will accept x with probability proportional to $p_d(x)/p_{\overline{\phi}}(x) = D^*(x)/(1 - D^*(x))$. Our scheme aims to approximate this ideal rejection sampling by approxinating D^* with the equilibrium strategy $\overline{D}$, whereas prior work uses the last discriminator D_N as an approximation.

4 Empirical Evaluation

We empirically evaluate how both the Private and Non-Private PGB methods affect the utility of the generated synthetic data from GANs. We show two appealing advantages of our approach: 1) non-private PGB outperforms other GAN post-processing methods, and 2) our approach can significantly improve the synthetic examples generated by a GAN under differential privacy.

Datasets. We assess our method with a toy dataset drawn from a mixture of 25 Gaussians, which is commonly used to evaluate the quality of GAN [31, 7, 35]. We then synthesize real datasets from the American Census, and a stanard machine learning dataset (Titanic).

Privacy budget. We set the privacy budget to be the same across all algorithms. Since Private PGB requires additional privacy budget this means that the DP GAN training has to be stopped earlier as compared to running only DP GAN to achieve the same privacy guarantee. Our principle is to allocate the majority of the privacy budget to the DP GAN training, and a much smaller budget for our Private PGB method. Throughout we used 80% of the final privacy budget on DP GAN training.[1]

Utility measures. Utility of synthetic data can be assessed along two dimensions; general utility and specific utility [30, 4]. General utility describes the overall distributional similarity between the real data and synthetic datasets, but does not capture specific use cases of synthetic data. To assess general utility, we use the propensity score mean squared error (pMSE) measure [30].[2] Specific utility of a synthetic dataset depends on the specific use an analyst has in mind. In general, specific utility can be defined as the similarity of results for analyses using synthetic data instead of real data. For each of the experiments we define specific utility measures that are sensible for the respective example. For the toy dataset of 25 gaussians we look at the number of high quality samples. For the American Census data we compare marginal distributions of the synthetic data to marginal distributions of the true data and look at the similarity of regression results.

4.1 Mixture of 25 Gaussians

We first examine the performance of our approach on a two dimensional dataset with a mixture of 25 multivariate Gaussian distributions, each with a covariance matrix of $0.0025I$. The left column in Figure 1 displays the training data. Each of the 25 clusters consists of $1,000$ observations. The architecture of the GAN is the same across all results. [3] To compare the utility of the synthetic datasets with the real data, we inspect the visual quality of the results, calculate the pMSE ratio score,[4] and calculate the proportion of high quality synthetic examples similar to [7, 35, 31][5]. In the real data, given the data generating process outlined above, at each mode 90% of the observations lie within a circle with radius $r = \sqrt{0.0025 \cdot 4.60517}$ around the mode centroids, with 4.60517, the critical value at $p = 0.9$ of a χ^2 distribution with 2 degrees of freedom and 0.0025 the variance of the spherical gaussian. To calculate the quality score we count the number of observations within each of these 25 circles. If one of the modes contains more points than we would expect given the true distribution the count is capped accordingly. Our quality score for the toy dataset of 25 gaussians can be expressed as $Q = \sum_i^{25}(\min(p^i_{real} \cdot N_{syn}, N^i_{syn})/N_{syn})$, where i indexes the clusters, p_{real} is the true distribution of points per cluster, N^i_{syn} the number of observations at a cluster within radius r, and N_{syn} the total number of synthetic examples.

[1] Our observation is that the DP GAN training is doing the "heavy lifting". Providing a good "basis" for PGB requires a substantial privacy expenditure in training DP GAN. The privacy budget allocation is a hyperparameter for PGB that could be tuned. In general, the problem of differentially private hyperparameter selection is extremely important and the literature is thin [24, 10].

[2] To calculate the pMSE one trains a discriminator to distinguish between real and synthetic examples. The predicted probability of being classified as real or synthetic is the propensity score. Taking all propensity scores into account the mean squared error between the propensity scores and the proportion of real data examples is calculated. A synthetic dataset has high general utility, if the model can at best predict probabilities of 0.5 for both real and synthetic examples, then the pMSE would be 0.

[3] A description of the architecture is in the appendix. The code for the GANs and the PGB can be found on `https://github.com/mneunhoe/post-gan-boosting`.

[4] The pMSE ratio score is the ratio of the pMSE score to its null expectation [30] For perfect synthesis we would expect a pMSE ratio score of 1. Higher values indicate lower general utility.

[5] Note that the scores in [7] and [35] do not account for the synthetic data distribution across the 25 modes.

Results without differential privacy. Visual inspection of the results without differential privacy (the top row in Figure 1) shows that our proposed method outperforms the synthetic examples generated by the last Generator of the GAN as well as the last Generator enhanced by DRS. PGB over the last 200 stored Generators and Discriminators trained for $T = 400$ update steps, visibly improves the results. Taking the distribution of synthetic examples after PGB and using it as the proposal distribution for DRS seems to improve the results further. The visual impression is confirmed both by the pMSE values as well as the proportion of high quality samples in Table 1. The combination of PGB and DRS achieves the best pMSE as well as the highest proportion of high quality samples.

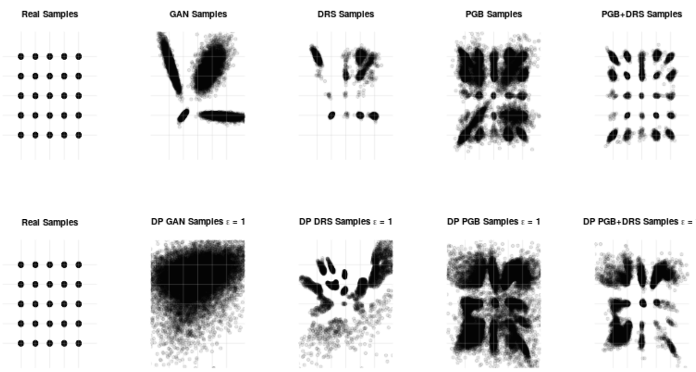

Figure 1: Real samples from 25 multivariate normal distributions (*left column*), synthetic examples without privacy from a GAN and three post-processing methods (*top row*) and synthetic examples from a GAN with differential privacy and the respective differentially private post processing methods (*bottom row*).

Results with differential privacy. We then run the experiment with DP.[6] Our final value of ϵ is 1 and δ is $\frac{1}{2N} = 0.00002$. For the results with PGB, DP GAN training contributes 0.8 to the overall epsilon and the Private PGB algorithm 0.2. Again a first visual inspection of the results in the bottom row of Figure 1 shows that post-processing the results of the last GAN Generator is worthwhile. PGB over the last 200 stored Generators and Discriminators trained for $T = 400$ update steps, again, visibly improves the results. Combining PGB with DRS further improves the quality of the samples. Our visual impression is confirmed by the general utility pMSE Ratio scores, and the proportion of high quality samples displayed in Table 1.

Table 1: Quality of Synthetic Data for the toy dataset of 25 Gaussians. Without differential privacy and with differential privacy ($\epsilon = 1$, $\delta = 0.00002$).

	GAN	DRS	PGB	PGB +DRS
pMSE Ratio	11.221	10.856	9.780	**6.862**
pMSE Ratio DP	12.488	13.026	**10.745**	10.952
Quality	0.055	0.147	0.164	**0.252**
Quality DP	0.008	0.019	0.032	**0.035**

4.2 American Census Data

The results on the toy dataset of 25 gaussians are encouraging. However, the ultimate goal of private synthetic data is to protect the privacy of actual persons in data collections, and to provide useful data to interested analysts. In this section we report the results of synthesizing data from the 1940 American Census. We rely on the public use micro data samples (PUMS) as provided in [29].[7]

[6] To achieve DP, we trained the Discriminator with the DP Adam optimizer as implemented in `tensorflow_privacy`. We keep track of the values of ϵ and δ by using the moments accountant [1, 27].

[7] More experiments using data from the 2010 American Census can be found in the appendix.

Private Synthetic 1940 American Census Samples. For 1940 we synthesize an excerpt of the 1% sample of all Californians that were at least 18 years old.[8] Our training sample consists of 39,660 observations and 8 attributes (sex, age, educational attainment, income, race, Hispanic origin, marital status and county). The test set contains another 9,915 observations. The GAN networks consist of two fully connected hidden layers (256, 128) with Leaky ReLu activation functions. To sample from categorical attributes we apply the Gumbel-Softmax trick [25, 22] to the output layer of the Generator. We run our PGB algorithm over the last 150 stored Generators and Discriminators and train it for $T = 400$ update steps. Our final value of ϵ is 1 and δ is $\frac{1}{2N} \approx 1.26 \times 10^{-5}$ (after DP GAN training with $\epsilon = 0.8$ and PGB $\epsilon = 0.2$). The general utility scores as measured by the pMSE ratio score are 2.357 (DP GAN), 2.313 (DP DRS), 2.253 (DP PGB), and 2.445 (DP PGB+DRS). This indicates that PGB achieves the best general utility. To assess the specific utility of our synthetic census samples we compare one-way marginal distributions to the same marginal distributions in the original data. In panel (A) of Figure 2 we show the distribution of race membership. Comparing the synthetic data distributions to the true distribution, we can conclude that PGB, as well as PGB+DRS, improves upon the last Generator as well as DRS. To underpin the visual impression we calculate the total variation distance between each of the synthetic distributions and the real distribution, the data from DP GAN has a total variation distance of 0.58, DP DRS of 0.44, DP PGB of 0.22 and DP PGB+DRS of 0.13. Furthermore, we evaluate whether more complex analysis models, such as regression models, trained on synthetic samples could be used to make sensible out-of-sample predictions. Panel (B) of Figure 2 shows that the out-of-sample root mean squared error of predicted income is lower for all linear regression models trained with three independent variables from the set of on the synthetic data generated with Private PGB as compared to DP GAN and DP DRS.

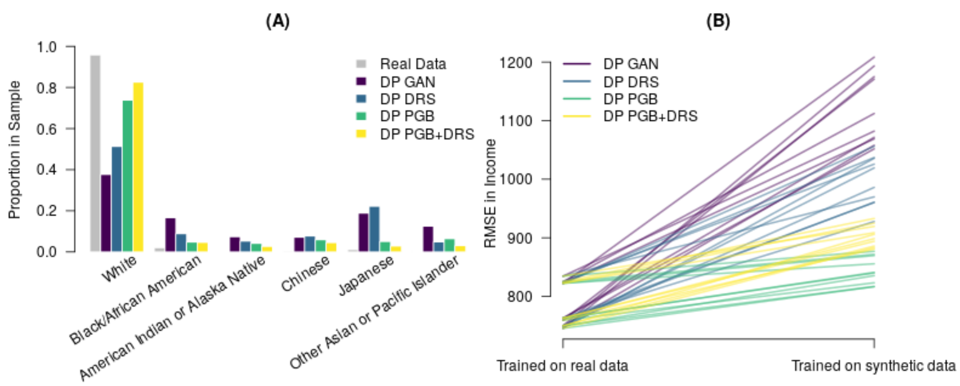

Figure 2: Specific Utility of Synthetic 1940 American Census Data. Panel (A): Distribution of Race Membership in Synthetic Samples. Panel (B): Regression RMSE with Synthetic Samples.

4.3 Machine Learning Prediction with Synthetic Data

In a final set of experiments we evaluate the performance of machine learning models trained on synthetic data (with and without privacy) and tested on real out-of-sample data. We synthesize the Kaggle Titanic[9] training set (891 observations of Titanic passengers on 8 attributes) and train three machine learning models (Logistic Regression, Random Forests (RF) [9] and XGBoost [11]) on the synthetic datasets to predict whether someone survived the Titanic catastrophe. We then evaluate the performance on the test set with 418 observations. To address missing values in both the training set and the test set we independently impute values using the MissForest [32] algorithm. For the private synthetic data our final value of ϵ is 2 and δ is $\frac{1}{2N} \approx 5.6 \times 10^{-4}$ (for PGB this implies DP GAN training with $\epsilon = 1.6$ and PGB $\epsilon = 0.4$). The models trained on synthetic data generated with our approaches (PGB and PGB+DRS) consistently perform better than models trained on synthetic data from the last generator or DRS – with or without privacy.[10]

[8] A 1% sample means that the micro data contains 1% of the total American (here Californian) population.

[9] `https://www.kaggle.com/c/titanic/data`

[10] Table 2 in the appendix summarizes the results in more detail. We present the accuracy, ROC AUC and PR AUC to evaluate the performance.

5 Broader Impact

Over past few years, there has been a wave of practical deployments of differential privacy across many organizations, including Google, Microsoft, Apple, LinkedIn, and, most notably, the US Census. However, almost all of the information released under differential privacy takes the form of noisy statistics, which can be hard for non-experts to interpret and impose limitations on downstream data analyses. One promising approach to enable non-experts to work with differential privacy is the release of synthetic data. Since the synthetic dataset already satisfies differential privacy guarantees, it places essentially no limitations on data analysts. Our work makes a solid contribution in this direction by providing a method for *general-purpose* synthetic data that are not optimized for a specific data analysis task. In the long run, we hope that our techniques will enable a wide range of privacy-preserving data releases that can provide useful scientific insights without compromising the privacy of individuals.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 308–318, New York, NY, USA, 2016. ACM.

[2] John M. Abowd. The U.S. census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, page 2867, 2018.

[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[4] Christian Arnold and Marcel Neunhoeffer. Really useful synthetic data–a framework to evaluate the quality of differentially private synthetic data. *arXiv preprint arXiv:2004.07740*, 2020.

[5] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 224–232, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[6] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(6):121–164, 2012.

[7] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian J. Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

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

[9] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[10] Kamalika Chaudhuri and Staal A Vinterbo. A stability-based validation procedure for differentially private machine learning. In *Advances in Neural Information Processing Systems*, pages 2652–2660, 2013.

[11] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[12] Differential Privacy Team, Apple. Learning with privacy at scale. `https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf`, December 2017.

[13] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 3571–3580. Curran Associates, Inc., 2017.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, volume 3876, pages 265–284, 2006.

[15] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.

[16] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[17] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*, pages 151–164, 2019.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[19] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 2348–2356, 2012.

[20] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 61–70, 2010.

[21] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[23] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

[24] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 298–309, 2019.

[25] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[26] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[27] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 263–275, 2017.

[28] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[29] Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas, and Matthew Sobek. Ipums usa: Version 9.0 [dataset]. *Minneapolis, MN: IPUMS*, 10:D010, 2019.

[30] Joshua Snoke, Gillian M. Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):663–688, 2018.

[31] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[32] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[33] Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5430–5439, USA, 2017. Curran Associates Inc.

[34] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. DP-CGAN: differentially private synthetic data and label generation. *CoRR*, abs/2001.09700, 2020.

[35] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-Hastings generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6345–6353, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[36] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018.

[37] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

A Appendix

A.1 Proof of Theorem 1

Proof of Theorem 1. Note that if the synthetic data player plays the distribution over X, then $U(p_X, D) = \mathbb{E}_{x\sim p_X}[D(x')] + \mathbb{E}_{x\sim\phi}[1 - D(x)] = 1$ for any discriminator $D \in \mathcal{D}$. Now let us replace each element in X with its γ-approximation in B and obtain a new dataset X_B, and let p_{X_B} denote the empirical distribution over X_B. By the Lipschitz conditions, we then have $|U(p_X, D) - U(p_{X_B}, D)| \leq L\gamma$. This means $U(p_{X_B}, D) \in [1 - L\gamma, 1 + L\gamma]$ for all D. Also, for all $\phi \in \Delta(B)$, we have $U(\phi, D^{1/2}) = 1$. Thus, $(p_{X_b}, D^{1/2})$ satisfies (1) with $\alpha = L\gamma$. □

A.2 Proof of the Approximate Equilibrium

Proof. We will use the seminal result of [16], which shows that if the two players have low regret in the dynamics, then their average plays form an approximate equilibrium. First, we will bound the regret from the data player. The regret guarantee of the multiplicative weights algorithm (see e.g. Theorem 2.3 of [6]) gives

$$\sum_{t=1}^{T} U(\phi^t, D^t) - \min_{b\in B} \sum_{t=1}^{T} U(b, D^t) \leq 4\eta T + \frac{\log|B|}{\eta} \tag{3}$$

Next, we bound the regret of the distinguisher using the accuracy guarantee of the exponential mechanism [26]. For each t, we know with probability $(1 - \beta/T)$,

$$\max_{D_j} U(\phi^t, D_j) - U(\phi^t, D^t) \leq \frac{2\log(NT/\beta)}{n\varepsilon_0}$$

Taking a union bound, we have this accuracy guarantee holds for all t, and so

$$\max_{D_j} \sum_{t=1}^{T} U(\phi^t, D_j) - \sum_{t=1}^{T} U(\phi^t, D^t) \leq \frac{2T\log(NT/\beta)}{n\varepsilon_0} \tag{4}$$

Then following the result of [16], their average plays $(\overline{D}, \overline{\phi})$ is an α-approximate equilibrium with

$$\alpha = 4\eta + \frac{\log|B|}{\eta T} + \frac{2\log(NT/\beta)}{n\varepsilon_0}$$

Plugging in the choices of T and η gives the stated bound. □

A.3 GAN Architecture for the Mixture of 25 Gaussians experiment

The generator and discriminator are neural nets with three fully connected hidden layers with Leaky ReLu activations. We add dropout layers to the generator net with a dropout rate of 50%. The latent noise vector Z is of dimension 32 and independently sampled from a gaussian distribution with mean 0 and standard deviation of 1.

A.4 Private Synthetic 2010 American Decennial Census Samples.

The 2010 data is similar to the data that the American Census will collect for the 2020 decennial Census. For this experiment, we synthesize a 10% sample for California with 3,723,669 observations of 5 attributes (gender, age, Hispanic origin, race and puma district membership). Our final value of ϵ is 0.795 and δ is $\frac{1}{2N} \approx 1.34 \times 10^{-7}$ (for PGB the GAN training contributes $\epsilon = 0.786$ and PGB $\epsilon = 0.09$). The pMSE ratio scores are 1.934 (DP GAN), 1.889 (DP DRS), 1.609 (DP PGB) and 1.485 (DP PGB+DRS), here PGB achieves the best general utility. For specific utility, we compare the accuracy of three-way marginals on the synthetic data to the proportions in the true data.[11] We tabulate race (11 answer categories in the 2010 Census) by Hispanic origin (25 answer categories in the 2010 Census) by gender (2 answer categories in the 2010 Census) giving us a total of 550 cells.

[11] A task that is similar to the tables released by the Census.

To assess the specific utility for these three-way marginals we calculate the average accuracy across all 550 cells. Compared to the true data DP GAN achieves 99.82%, DP DRS 99.89%, DP PGB 99.89% and the combination of DP PGB and DRS 99.93%. Besides the average accuracy across all 550 cells another interesting metric of specific utility is the number of cells in which each synthesizer achieves the highest accuracy compared to the other methods, this is the case 43 times for DP GAN, 30 times for DP DRS, 90 times for DP PGB and 387 times for DP PGB+DRS. Again, this shows that our proposed approach can improve the utility of private synthetic data.

A.5 Detailed Results of Machine Learning Prediction with Synthetic Data

Table 2 summarizes the results for the machine learning prediction experiment with the Titanic data. We present the accuracy, ROC AUC and PR AUC to evaluate the performance. It can be seen that the models trained on synthetic data generated with our approach consistently perform better than models trained on synthetic data from the last generator or DRS – with or without privacy. To put these values into perspective, the models trained on the real training data and tested on the same out-of-sample data achieve the scores in table 3.

Table 2: Predicting Titanic Survivors with Machine Learning Models trained on synthetic data and tested on real out-of-sample data. Median scores of 20 repetitions with independently generated synthetic data. With differential privacy ϵ is 2 and δ is $\frac{1}{2N} \approx 5.6 \times 10^{-4}$.

	GAN	DRS	PGB	PGB + DRS
Logit Accuracy	0.626	0.746	0.701	**0.765**
Logit ROC AUC	0.591	0.760	0.726	**0.792**
Logit PR AUC	0.483	0.686	0.655	**0.748**
RF Accuracy	0.594	0.724	0.719	**0.742**
RF ROC AUC	0.531	0.744	0.741	**0.771**
RF PR AUC	0.425	0.701	0.706	**0.743**
XGBoost Accuracy	0.547	0.724	0.683	**0.740**
XGBoost ROC AUC	0.503	0.732	0.681	**0.772**
XGBoost PR AUC	0.400	0.689	0.611	**0.732**
	DP GAN	DP DRS	DP PGB	DP PGB +DRS
Logit Accuracy	0.566	0.577	0.640	**0.649**
Logit ROC AUC	0.477	0.568	0.621	**0.624**
Logit PR AUC	0.407	0.482	0.532	**0.547**
RF Accuracy	0.487	0.459	0.481	**0.628**
RF ROC AUC ROC AUC	0.512	0.553	0.558	**0.652**
RF PR AUC PR AUC	0.407	0.442	0.425	**0.535**
XGBoost Accuracy	0.577	0.589	0.609	**0.641**
XGBoost ROC AUC	0.530	0.586	**0.619**	0.596
XGBoost PR AUC	0.398	0.479	0.488	**0.526**

A.6 Results with the same GANs across all methods

To compare our method against other we produced synthetic data that has the same privacy guarantees across all methods. To achieve this we trained the DP GAN for additional update steps. This means, however, that it is difficult to see how much Post-GAN Boosting improves upon the last Generator that was used in PGB. Therefore, a second possibility to assess the contribution of our proposed Post-GAN Boosting approach is to take the same sequence of generators and discriminators across all the results. For these results DP PGB uses an additional privacy budget compared to DP GAN or DP DRS. The following results together with the results in the main part of the paper show that spending more privacy budget on the GAN training does not improve the sample quality by as much as if the same privacy budget was spent on our PGB methods.

Table 3: Predicting Titanic Survivors with Machine Learning Models trained on real data and tested on real out-of-sample data.

Model	Score
Logit Accuracy	0.764
Logit ROC AUC	0.813
Logit PR AUC	0.785
RF Accuracy	0.768
RF ROC AUC	0.809
RF PR AUC	0.767
XGBoost Accuracy	0.768
XGBoost ROC AUC	0.773
XGBoost PR AUC	0.718

Table 4: Quality of Synthetic Data for the toy dataset of 25 Gaussians. Without differential privacy and with differential privacy where DP GAN and DP DRS satisfy ($\epsilon = 0.635$, $\delta = 0.00002$) differential privacy and DP PGB and DP PGB+DRS ($\epsilon = 1$, $\delta = 0.00002$) differential privacy.

	GAN	DRS	PGB	PGB +DRS
pMSE Ratio DP	3.019	3.124	2.694	**2.671**
Quality DP	0.009	0.010	0.058	**0.072**

Mixture of 25 Gaussians. In figure 3 we display the synthetic data that used the same sequence of generators and discriminators across all results. The corresponding quality scores can be found in table 4.

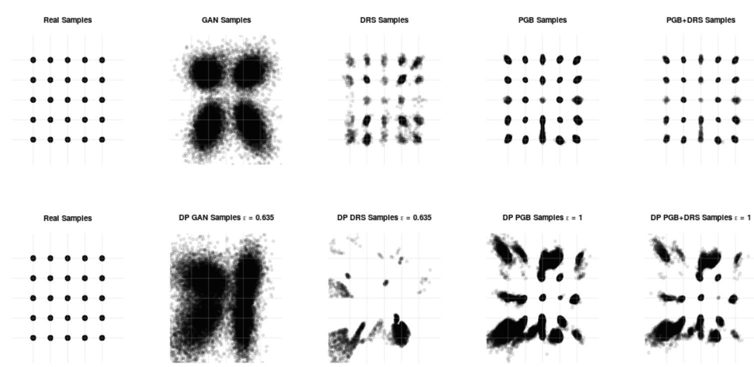

Figure 3: Real samples from 25 multivariate normal distributions (*left column*), synthetic examples without privacy from a GAN and three post-processing methods (*top row*) and synthetic examples from a GAN with differential privacy and the respective differentially private post processing methods (*bottom row*).

Private Synthetic 1940 American Census Samples. Our final value of ϵ is 1 and δ is $\frac{1}{2N} \approx 1.26 \times 10^{-5}$ for DP PGB and DP PGB+DRS (after GAN training with $\epsilon = 0.799$ and PGB $\epsilon = 0.201$). The results for DP GAN and DP DRS satisfy $\epsilon = 0.799$ and $\delta = \frac{1}{2N} \approx 1.26 \times 10^{-5}$.

For this setup, the general utility scores as measured by the pMSE ratio score are 7.109 (DP GAN), 7.898 (DP DRS), 6.353 (DP PGB), and 5.214 (DP PGB+DRS). Figure 4 displays the distribution of race membership and figure 5 shows that the out-of-sample root mean squared error of predicted income is lower for all linear regression models trained with three independent variables from the set of on the synthetic data generated with Private PGB as compared to DP GAN and DP DRS.

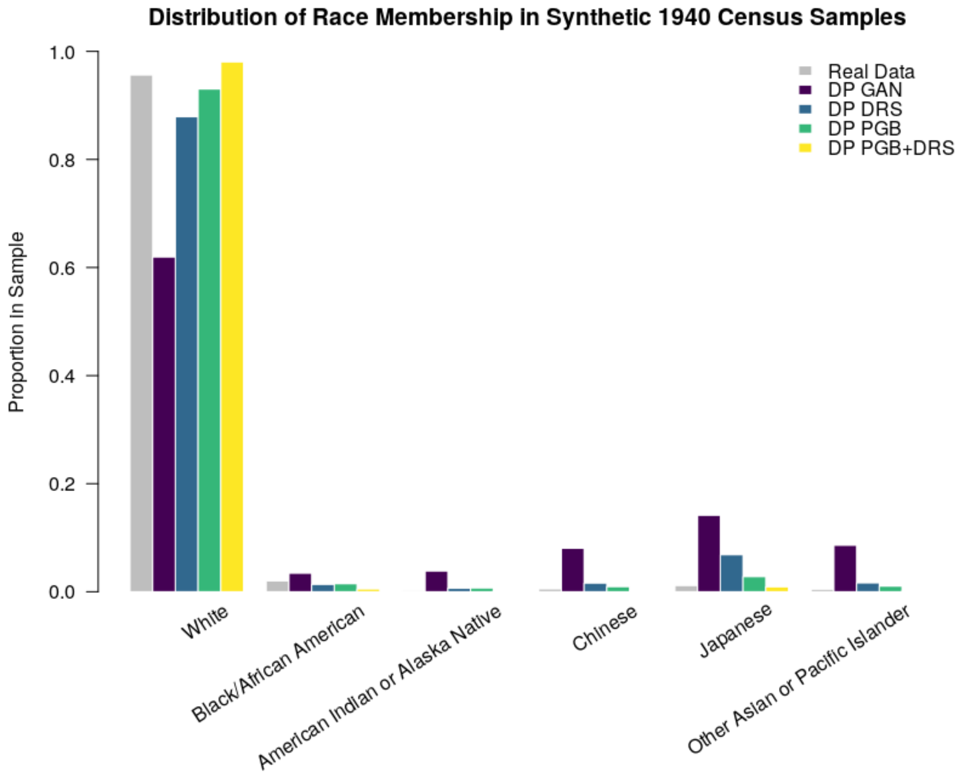

Figure 4: The distribution of races in private synthetic 1940 census data.

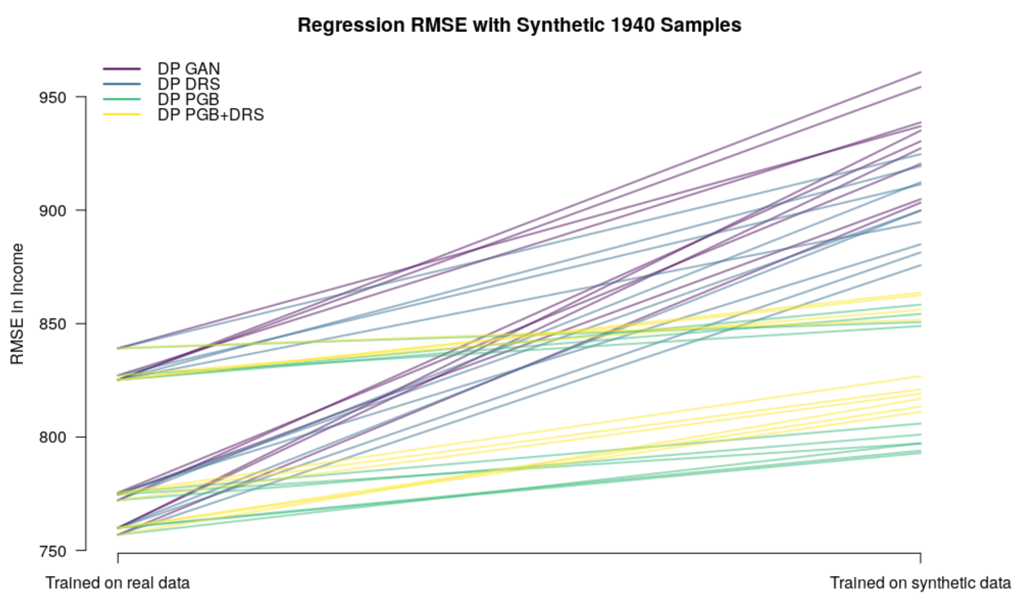

Figure 5: Private synthetic 1940 census data. Comparison of regression out-of-sample prediction root mean squared errors of income for all combinations of linear regression with three independent variables from the attributes sex, age, educational attainment, race and marital status.

Machine Learning Prediction with Synthetic Data. For the private synthetic data our final value of ϵ is 2 and δ is $\frac{1}{2N} \approx 5.6 \times 10^{-4}$ (after GAN training with $\epsilon = 1.547$ and PGB $\epsilon = 0.453$). Thus, DP GAN and DP DRS are $\epsilon = 1.547, \delta = 5.6 \times 10^{-4}$) differentially private. Table 5 summarizes the results. As before, we present the accuracy, ROC AUC and PR AUC to evaluate the performance. It can be seen that the models trained on synthetic data generated with our approaches (PGB and PGB+DRS) consistently perform better than models trained on synthetic data from the last generator or DRS – with or without privacy.

Table 5: Predicting Titanic Survivors with Machine Learning Models trained on differentially private synthetic data and tested on real out-of-sample data. The ϵ for DP GAN and DP DRS is 1.547, and 2 for DP PGB and DP PGB+DRS. Median scores of 25 repetitions of independently generated synthetic data.

	DP GAN	DP DRS	DP PGB	DP PGB +DRS
Logit Accuracy	0.537	0.606	0.583	**0.615**
Logit ROC AUC	0.487	0.604	**0.631**	0.570
Logit PR AUC	0.411	0.483	**0.538**	0.467
RF Accuracy	0.495	0.621	0.591	**0.628**
RF ROC AUC	0.467	0.644	0.628	**0.682**
RF PR AUC	0.389	0.509	0.517	**0.549**
XGBoost Accuracy	0.520	0.591	0.596	**0.639**
XGBoost ROC AUC	0.529	**0.643**	0.620	0.626
XGBoost PR AUC	0.410	0.508	0.509	**0.568**