# Differentially Private Synthetic Mixed-Type Data Generation For Unsupervised Learning

Uthaipon Tao Tantipongpipat
*Twitter*
uthaipon@gmail.com

Chris Waites
*Stanford University*
waites@stanford.edu

Digvijay Boob
*Southern Methodist University*
dboob@mail.smu.edu

Amaresh Ankit Siva
*Amazon*
ankitsiv@amazon.com

Rachel Cummings
*Columbia University*
rac2239@columbia.edu

***Abstract*—We introduce the DP-auto-GAN framework for synthetic data generation, which combines the low dimensional representation of autoencoders with the flexibility of Generative Adversarial Networks (GANs). This framework can be used to take in raw sensitive data and privately train a model for generating synthetic data that will satisfy similar statistical properties as the original data. This learned model can generate an arbitrary amount of synthetic data, which can then be freely shared due to the post-processing guarantee of differential privacy. Our framework is applicable to unlabeled *mixed-type data*, that may include binary, categorical, and real-valued data. We implement this framework on both binary data (MIMIC-III) and mixed-type data (ADULT), and compare its performance with existing private algorithms on metrics in unsupervised settings. We also introduce a new quantitative metric able to detect diversity, or lack thereof, of synthetic data.**

## I. Introduction

As data storage and analysis are becoming more cost effective, and data become more complex and unstructured, there is a growing need for sharing large datasets for research and learning purposes. This is in stark contrast to the previous statistical model where a data curator would hold datasets and answer specific queries from (potentially external) analysts. Sharing entire datasets allows analysts the freedom to perform their analyses in-house with their own devices and toolkits, without having to pre-specify the analyses they wish to perform. However, datasets are often proprietary or sensitive, and they cannot be shared directly. This motivates the need for *synthetic data generation*, where a new dataset is created that shares the same statistical properties as the original data. These data may not be of a single type: all binary, all categorial, or all real-valued; instead they may be of *mixed-types*, containing data of multiple types in a single dataset. These data may also be unlabeled, requiring techniques for *unsupervised learning*, which is typically a more challenging task than *supervised* learning when data are labeled.

U.T. supported in part by NSF grants AF-1910423 and AF-1717947. C.W. supported in part by a President's Undergraduate Research Award from the Georgia Institute of Technology. D.B. supported in part by NSF grant CCF-1909298. R.C. supported in part by a Mozilla Research Grant, a Google Research Fellowship, a JPMorgan Chase Faculty Award, and NSF grants CNS-1850187 and CNS-194277 (CAREER). Part of this work was completed while R.C. was visiting the Simons Institute for the Theory of Computing. Most of this work was completed while all authors were affiliated with Georgia Institute of Technology.

Privacy challenges naturally arise when sharing highly sensitive datasets about individuals. Ad hoc anonymization techniques have repeatedly led to severe privacy violations when sharing "anonymized" datasets. Notable examples include the Netflix Challenge [1], AOL Search Logs [2], and Massachusetts State Health data [3], where linkage attacks to publicly available auxiliary datasets were used to reidentify individuals in the dataset. Deep learning models have been shown to inadvertently memoize sensitive personal information such as Social Security Numbers during training [4].

Differential privacy (DP) [5] (formally defined in Section II) has become the de facto gold standard of privacy in the computer science literature. Informally, it bounds the extent to which an algorithm depends on a single datapoint in its training set. The guarantee ensures that any differentially privately learned models do not overfit to individuals in the database, and therefore cannot reveal sensitive information about individuals. It is an information theoretic notion that does not rely on any assumptions of an adversary's computational power or auxiliary knowledge. Furthermore, it has been shown empirically that training machine learning models with differential privacy protects against membership inference and model inversion attacks [4], [6]. Differentially private algorithms have been deployed at large scale in practice by organizations such as Apple, Google, Microsoft, Uber, and the U.S. Census Bureau.

Much of the prior work on differentially private synthetic data generation has been either theoretical algorithms for highly structured classes of queries [7], [8] or based on deep generative models such as Generative Adversarial Networks (GANs) or autoencoders. These architectures have been primarily designed for either all-binary or all-real-valued datasets, and have focused on the *supervised* setting.

In this work we introduce the *DP-auto-GAN* framework, which combines the low dimensional representation of autoencoders with the flexibility of GANs. This framework can be used to take in raw sensitive data, and privately train a model for generating synthetic data that satisfies similar statistical properties as the original data. This learned model

can be used to generate arbitrary amounts of publicly available synthetic data, which can then be freely shared due to the post-processing guarantees of differential privacy. We implement this framework on both unlabeled binary data (for comparison with previous work) and unlabeled mixed-type data. We also introduce new metrics for evaluating the quality of synthetic mixed-type data in unsupervised settings, and empirically evaluate the performance of our algorithm according to these metrics on two datasets.

### A. Our Contributions

In this work, we provide two main contributions: a new algorithmic framework for privately generating synthetic data, and empirical evaluations of our algorithmic framework showing improvements over prior work. Along the way, we also develop a novel privacy composition method with tighter guarantees, and we generalize previous metrics for evaluating the quality of synthetic datasets to the unsupervised mixed-type data setting. Both of these contributions may be of independent interest.

**Algorithmic Framework.** We propose a new data generation architecture which combines the versatility of an autoencoder [9] with the recent success of GANs on complex data. Our model extends previous autoencoder-based DP data generation [10], [11] by removing an assumption that the distribution of the latent space follows a Gaussian mixture distribution. Instead, we incorporate GANs into the autoencoder framework so that the generator must learn the true latent distribution against the discriminator. We describe the composition analysis of differential privacy when the training consists of optimizing both autoencoders and GANs (with different noise parameters).

**Empirical Results.** We empirically evaluate the performance of our algorithmic framework on the MIMIC-III medical dataset [12] and UCI ADULT Census dataset [13], and compare against previous approaches in the literature [10], [14]–[16]. Our experiments show that our algorithms perform better and obtain substantially improved $\epsilon$ values of $\epsilon \approx 1$, compared to $\epsilon \approx 200$ in prior work [15]. The performance of our algorithm remains high along a variety of quantitative and qualitative metrics, even for small values of $\epsilon$, corresponding to strong privacy guarantees. Our code is publicly available for future use and research.

### B. Related Work on Differentially Private Data Generation

Early work on differentially private synthetic data generation was focused primarily on theoretical algorithms for solving the *query release problem* of privately and accurately answering a large class of pre-specified queries on a given database. It was discovered that generating synthetic data on which the queries could be evaluated allowed for better privacy composition than simply answering all the queries directly [7], [8], [17], [18]. Bayesian inference has also been used for differentially private data generation [19], [20] by estimating the correlation between features. See [21] for a survey of techniques used in private synthetic data generation.

More recently, [22] introduced a framework for training deep learning models with differential privacy, which involved adding Gaussian noise to a clipped (norm-bounded) gradient in each training step. [22] also introduced the *moment accountant* privacy analysis, which provided a tighter Gaussian-based privacy composition and allowed for significant improvements in accuracy. It was later defined in terms of *Renyi Differential Privacy (RDP)* [23], which is a slight variant of differential privacy designed for easy composition. Much of the work that followed used deep generative models, and can be broadly categorized into two types: autoencoder-based and GAN-based. Our algorithmic framework is the first to combine both DP GANs and autoencoders.

Due to space constraints, we focus here on the three most relevant recent works on privately generating synthetic mixed-type data. [10] considers the problem of generating mixed-type labeled data with $k$ possible labels. Their algorithm, DP-SYN, partitions the dataset into $k$ sets based on the labels and trains a DP autoencoder on each partition. Then the DP expectation maximization (DP-EM) algorithm of [24] is used to learn the distribution in the latent space of encoded data of the given label-class. The main workhorse, DM-EM algorithm, is designed and analyzed for Gaussian mixture models and more general factor analysis models. [11] works in the same setting, but replaces the DP autoencoder and DP-EM with a DP variational autoencoder (DP-VAE). Their algorithm assumes that the mapping from real data to the Gaussian distribution can be efficiently learned by the encoder. Finally, [14] uses a Wasserstein GAN (WGAN) to generate differentially private mixed-type synthetic data, which uses a Wasserstein-distance-based loss function in training. Their algorithmic framework privatizes the WGAN using DP-SGD, similar to previous approaches for image datasets [15], [25]. The methodology of [14] for generating mixed-type synthetic data involves two main ingredients: changing discrete (categorical) data to binary data using one-hot encoding, and adding an output softmax layer to the WGAN generator for every discrete variable.

Our framework is distinct from these three approaches. We use a differentially private autoencoder which, unlike DP-VAE of [11], does not require mapping data to a Gaussian distribution. This allows us to reduce the dimension of the problem handled by the WGAN, hence escaping the issues of high-dimensionality from the one-hot encoding of [14]. We also use DP-GAN, replacing DP-EM in [10], to learn more complex distributions in the latent or encoded space.

## II. Preliminaries on Differential Privacy

In the setting of differential privacy, a dataset $X$ consists of $m$ individuals' sensitive information, and two datasets are neighbors if one can be obtained from the other by the addition or deletion of one datapoint. Differential privacy requires that an algorithm produce similar outputs on neighboring datasets, thus ensuring that the output does not overfit to its input dataset, and that the algorithm learns from the population but not from the individuals.

**Definition 1** (Differential privacy [5]). *For $\epsilon, \delta > 0$, an algorithm $\mathcal{M}$ is $(\epsilon, \delta)$-*differentially private *if for any pair of neighboring databases $X, X'$ and any subset $S$ of possible outputs produced by $\mathcal{M}$,*

$$\Pr[\mathcal{M}(X) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(X') \in S] + \delta.$$

A smaller value of $\epsilon$ implies stronger privacy guarantees (as the constraint above binds more tightly), but usually corresponds with decreased accuracy, relative to non-private algorithms or the same algorithm run with a larger value of $\epsilon$. Differential privacy is typically achieved by adding random noise that scales with the *sensitivity* of the computation being performed, which is the maximum change in the output value that can be caused by changing a single entry. Differential privacy has strong *composition guarantees*, meaning that the privacy parameters degrade gracefully as additional algorithms are run on the same dataset. It also has a *post-processing* guarantee, meaning that any function of a differentially private output will retain the same privacy guarantees.

### A. Differentially Private Stochastic Gradient Descent (DP-SGD)

The DP-SGD framework of [22] can be used to privately train deep learning models, and is used in our framework to train the autoencoder and GAN. The training process involves minimizing an empirical loss function $f(X;\theta) := \frac{1}{m}\sum_{i=1}^{m} f(x_i;\theta)$ on a dataset $X = \{x_i \in \mathbb{R}^n\}_{i=1}^m$. Typically $f$ is nonconvex, and a common method to minimize $f$ is stochastic gradient descent (SGD). To make SGD private, [22] proposed clipping the gradient of each sample to bound the $\ell_2$-norm, and adding multivariate Gaussian noise to the gradient. The clipping reduces the scale of noise that must be added to preserve differential privacy. The noisy-clipped-gradient estimate is then used in the update step instead of the true gradient.

### B. Renyi Differential Privacy Accountant

A variant notion of differential privacy, known as *Renyi Differential Privacy (RDP)* [23], is often used to analyze privacy for DP-SGD. A randomized mechanism $\mathcal{M}$ is $(\alpha, \epsilon)$-RDP if for all neighboring databases $X, X'$ that differ in at most one entry, $RDP(\alpha) := D_\alpha(\mathcal{M}(X)||\mathcal{M}(X')) \leq \epsilon$, where $D_\alpha(P||Q) := \frac{1}{\alpha-1}\log \mathbb{E}_{x\sim X}\left(\frac{P(x)}{Q(x)}\right)^{\alpha}$ is the *Renyi divergence* or *Renyi entropy* of order $\alpha$ between two distributions $P$ and $Q$. Renyi divergence is better tailored to tightly capture the privacy loss from the Gaussian mechanism that is used in DG-SGD, and is a common analysis tool for DP-SGD literature. To compute the final $(\epsilon, \delta)$-differential privacy parameters from iterative runs of DP-SGD, one must first compute the subsampled Renyi Divergence, then compose privacy under RDP, and then convert the RDP guarantee into DP.

## III. Algorithmic Framework

The overview of our algorithmic framework DP-auto-GAN is shown in Figure 1, and the full details are given in Algorithm 1. Details of the subroutines are deferred to the full version. The algorithm takes in $m$ raw data points, and *pre-processes* these points into $m$ vectors $x_1, \ldots, x_m \in \mathbb{R}^n$ to be read by DP-auto-GAN, where usually $n$ is very large. For example, categorical data may be pre-processed using one-hot encoding, or text may be converted into high-dimensional vectors. Similarly, the output of DP-auto-GAN can be *post-processed* from $\mathbb{R}^n$ back to the data's original form. We assume that this pre- and post-processing can done based on public knowledge, such as possible categories for qualitative features and reasonable bounds on quantitative features, and therefore do not incur a privacy cost.

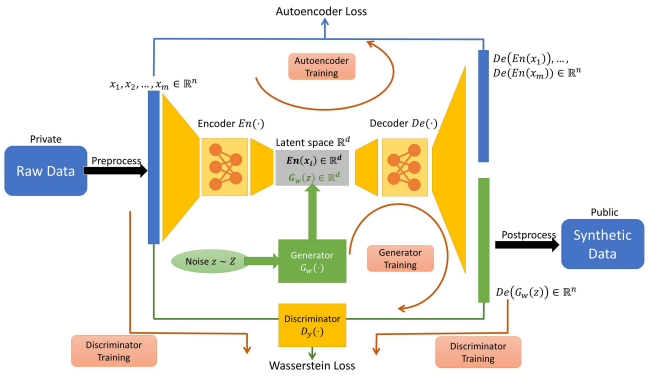



Fig. 1: The summary of our DP-auto-GAN algorithmic framework. Pre- and post-processing (in black) are assumed to be public knowledge. Generator (in green) is trained without noise, whereas encoder, decoder, and discriminator (in yellow) are trained with noise. The four red arrows indicate how data are forwarded for autoencoder, generator, and discriminator training. After training, the generator and decoder are released to the public to generate synthetic data.

Within the DP-auto-GAN, there are two main components: the *autoencoder* and the GAN. The autoencoder serves to reduce the dimension of the data to $d \ll n$ before it is fed into the GAN. The GAN consists of a *generator* that takes in noise $z$ sampled from a distribution $Z$ and produces $G_w(z) \in \mathbb{R}^d$, and a *discriminator* $D_y(\cdot) : \mathbb{R}^n \to \{0, 1\}$. Because of the autoencoder, the generator only needs to synthesize data based on the latent distribution in $\mathbb{R}^d$, which is much easier than synthesizing in $\mathbb{R}^n$. Both components of our architecture, as well as our algorithm's overall privacy guarantee, are described in the remainder of this section.

### A. Autoencoder Framework and Training

An autoencoder consists of an encoder $En_\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^d$ and a decoder $De_\theta(\cdot) : \mathbb{R}^d \to \mathbb{R}^n$ parametrized by weights $\phi, \theta$ respectively. The architecture of the autoencoder assumes that high-dimensional data $x_i \in \mathbb{R}^n$ can be represented compactly in a low-dimensional *latent space* $\mathbb{R}^d$. The encoder $En_\phi$ is trained to find such low-dimensional representations, and the decoder $De_\theta$ maps $En_\phi(x_i)$ in the latent space back to $x_i$. A natural measure of the information preserved in this process is the error between the decoder's image and the original $x_i$. A good autoencoder should minimize the distance $\text{dist}(De_\theta(En_\phi(x_i)), x_i)$ for each point $x_i$ for an appropriate

**Algorithm 1** DPAUTOGAN (full procedure)

**architecture input:** Sensitive dataset $D \in \mathcal{X}^m$ where $\mathcal{X}$ is the (raw) data universe, preprocessed data dimension $n$, latent space dimension $d$, preprocessing function $Pre : \mathcal{X} \to \mathbb{R}^n$, post-processing function $Post : \mathbb{R}^n \to \mathcal{X}$, encoder architecture $En_\phi : \mathbb{R}^n \to \mathbb{R}^d$ parameterized by $\phi$, decoder architecture $De_\theta : \mathbb{R}^d \to \mathbb{R}^n$ parameterized by $\theta$, generator's noise distribution $Z$ on sample space $\Omega(Z)$, generator architecture $G_w : \Omega(Z) \to \mathbb{R}^d$ parameterized by $w$, discriminator architecture $D_y : \mathbb{R}^n \to \{0, 1\}$ parameterized by $y$.
**autoencoder training parameters**: Learning rate $\eta_1$, number of iteration rounds (or optimization steps) $T_1$, loss function $L_{\text{auto}}$, optimization method $\text{OPTIM}_{\text{auto}}$ batch sampling rate $q_1$ (for batch expectation size $b_1 = q_1 m$), clipping norm $C_1$, noise multiplier $\psi_1$, microbatch size $r_1$
**generator training parameters**: Learning rate $\eta_2$, batch size $b_2$, loss function $L_G$, optimization method $\text{OPTIM}_G$, number of generator iteration rounds (or optimization steps) $T_2$
**discriminator training parameters**: Learning rate $\eta_3$, number of discriminator iterations per generator step $t_D$, loss function $L_D$, optimization method $\text{OPTIM}_D$, batch sampling rate $q_3$ (for batch expectation size $b_3 = q_3 m$), clipping norm $C_3$, noise multiplier $\psi_3$, microbatch size $r_3$
**privacy parameter** $\delta > 0$
**procedure** DPautoGAN
  $X \leftarrow Pre(D)$
  Initialize $\phi, \theta, w, y$ for $En_\phi, De_\theta, G_w, D_y$
  ▷ *Phase 1: autoencoder training*
**for** $t = 1 \dots T_1$ **do**
    $\text{DPTRAIN}_{\text{AUTO}}(X,\ En,\ De,$ autoencoder training parameters)
  ▷ *Phase 2: GAN training*
**for** $t = 1 \dots T_2$ **do**
    **for** $j = 1 \dots t_D$ **do**
    ▷ *(privately) train $D_y$ for $t_D$ iterations*
      $\text{DPTRAIN}_{\text{DISCRIMINATOR}}(\ X, Z, G, De, D,$ discriminator training parameters)
    $\text{TRAIN}_{\text{GENERATOR}}(Z, G, De, D,$ generator training parameters)
  ▷ *Privacy accounting*
$\text{RDP}_{\text{auto}}(\cdot) \leftarrow \text{RDP-ACCOUNT}(T_1, q_1, \psi_1, r_1)$
$\text{RDP}_D(\cdot) \leftarrow \text{RDP-ACCOUNT}(T_2 \cdot t_D, q_3, \psi_3, r_3)$
$\epsilon \leftarrow \text{GET-EPS}(\text{RDP}_{\text{auto}}(\cdot) + \text{RDP}_D(\cdot))$
**return** model $(G_w, De_\theta)$, privacy $(\epsilon, \delta)$

distance function dist. Our autoencoder uses binary cross entropy loss: $\text{dist}(x, y) = -\sum_{j=1}^{n} y_{(j)} \log(x_{(j)}) - \sum_{j=1}^{n}(1 - y_{(j)}) \log(1 - x_{(j)})$, where $x_{(j)}$ is the $j$th coordinate of the data $x \in [0, 1]^n$ after our preprocessing.

This motivates a definition of a (true) loss function $\mathbb{E}_{x \sim Z_X}[\text{dist}(De_\theta(En_\phi(x)), x)]$ when data are drawn independently from an underlying distribution $Z_X$. The corresponding empirical loss function when we have an access to sample $\{x_i\}_{i=1}^m$ is

$$L_{\text{auto}}(\phi, \theta) := \sum_{i=1}^{m} \text{dist}(De_\theta(En_\phi(x_i)), x_i). \quad (1)$$

Finding a good autoencoder requires optimizing $\phi$ and $\theta$ to yield small empirical loss in Equation 1.

We minimize Equation 1 privately using DP-SGD (Section II-A). Our approach follows previous work on private training of autoencoders [10], [11], [16] by adding noise to both the encoder and decoder. In our DP-auto-GAN framework, the autoencoder is trained first until completion, and is then fixed while training the GAN.

### B. GAN Framework and Training

A GAN consists of a generator $G_w$ and a discriminator $D_y : \mathbb{R}^n \to \{0, 1\}$, parameterized respectively by weights $w$ and $y$. The aim of the generator $G_w$ is to synthesize (fake) data similar to the real dataset, while the discriminator aims to determine whether an input $x_i$ is from the generator's synthesized data (and assign label $D_y(x_i) = 0$) or is real data (and assign label $D_y(x_i) = 1$). The generator is seeded with a random noise $z \sim Z$ that is independent of the data, such as a multivariate Gaussian vector, and aims to generate a distribution $G_w(z)$ that is hard for $D_y$ to distinguish from the real data. Hence, the generator wants to minimize the probability that $D_y$ makes a correct guess, $\mathbb{E}_{z \sim Z}[1 - D_y(G_w(z))]$. The discriminator wants to maximize its probability of a correct guess, which is $\mathbb{E}_{z \sim Z}[1 - D_y(G_w(z))]$ when the datum is fake and $\mathbb{E}_{x \sim Z_X}[D_y(x)]$ when it is real.

We extend the binary output of $D_y$ to a continuous range $[0, 1]$, with the value indicating the confidence that a sample is real. We use the zero-sum objective for the discriminator and generator [26], which is motivated by the Wasserstein distance of two distributions. Although the proposed Wasserstein objective cannot be computed exactly, it can be approximated by optimizing:

$$\min_y \max_w O(y, w) := \mathbb{E}_{x \sim Z_X}[D_y(x)] - \mathbb{E}_{z \sim Z}[D_y(G_w(z))]. \quad (2)$$

We optimize Equation 2 privately using the DP-SGD framework described in Section II-A. We differ from prior work on DP-GANs in that our generator $G_w(\cdot)$ outputs data $G_w(z)$ in the latent space $\mathbb{R}^d$, which needs to be decoded by the fixed (pre-trained) $De_\theta$ to $De_\theta(G_w(z))$ before being fed into the discriminator $D_y(z)$. The gradient $\nabla_w G_w$ is obtained by backpropagation through this additional component $De_\theta(\cdot)$.

After this two-phase training (of the autoencoder and GAN), the noise distribution $Z$, trained generator $G_w(\cdot)$, and trained decoder $De_\theta(\cdot)$ are released to the public. The public can sample $z \sim Z$ to obtain a synthesized datapoint $De_\theta(G_y(z))$ repeatedly to obtain a synthetic dataset of any desired size.

### C. Privacy Accounting

We use Renyi Differential Privacy (RDP) of [23], to account for privacy in each phase of training as in prior works. Our autoencoder and GAN are trained privately by clipping

gradients and adding noise to the encoder, decoder, and discriminator. Since the generator only accesses data through the discriminator's (privatized) output and $De_\theta$ is first trained privately and then fixed during GAN training, the trained parameters of generator are also private by post-processing guarantees of differential privacy. Privacy accounting is therefore required for only two parts that access real data $X$: training of the autoencoder and of the discriminator. In each training procedure, we apply the RDP accountant described in Section II-B, to analyze privacy of the DP-SGD training.

The RDP accountant is a function $r : [1, \infty) \to \mathbb{R}_+$ and guarantees $(\epsilon, \delta)$-DP for any given $\delta > 0$ with $\epsilon = \min_{\alpha>1} r(\alpha) + \frac{\log 1/\delta}{\alpha-1}$ ( [23]; also used in Tensorflow Privacy [27]). Hence, at the end of two-phase training, we have two RDP accountants $r_1, r_2$. We compose two RDP accountants *before* converting the combined accountant into $(\epsilon, \delta)$-DP. Note that another method used in DP-SYN [10] first converts $r_i$ to $(\epsilon_i, \delta_i)$-DP and then combines them into $(\epsilon_1+\epsilon_2, \delta_1+\delta_2)$-DP by basic composition [5]. For completeness, we show that composing RDP accountants first always results in a better privacy analysis.

**Lemma 2.** *Let $\mathcal{M}_1, \mathcal{M}_2$ be any mechanisms and $r_1, r_2 : [1, \infty) \to \mathbb{R}_+ \cup \{\infty\}$ be functions such that $\mathcal{M}_1, \mathcal{M}_2$ are $(\alpha, r_1(\alpha))$- and $(\alpha, r_2(\alpha))$-RDP, respectively. Let $\delta \in (0, 1]$ and let*

$$\epsilon_1 = \min_{\alpha>1} r_1(\alpha) + \frac{\log(2/\delta)}{\alpha - 1}, \quad \epsilon_2 = \min_{\alpha>1} r_2(\alpha) + \frac{\log(2/\delta)}{\alpha - 1},$$

$$\textit{and} \quad \epsilon = \min_{\alpha>1} r_1(\alpha) + r_2(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}.$$

*Then $\mathcal{M}_1$ is $(\epsilon_1, \delta/2)$-DP, $\mathcal{M}_2$ is $(\epsilon_2, \delta/2)$-DP, and the composition $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ is $(\epsilon, \delta)$-DP. If $\epsilon_1$ and $\epsilon_2$ are finite, then $\epsilon < \epsilon_1 + \epsilon_2$.*

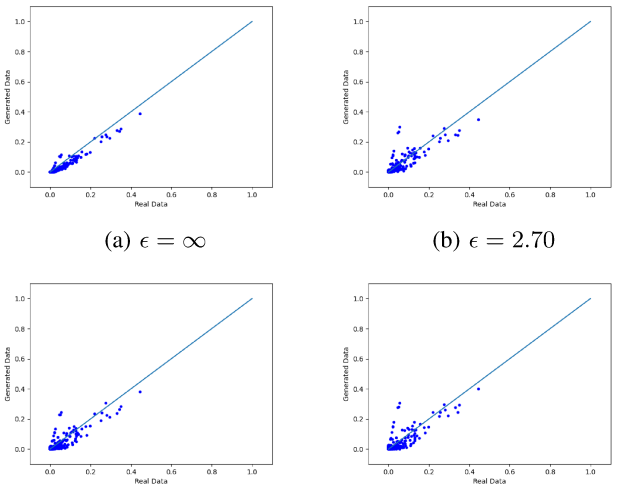
In practice, we observe that composing at the RDP level first in Lemma 2 reduces privacy cost by $\approx 30\%$.

## IV. Experiments

In this section, we empirically evaluate the performance of our DP-auto-GAN framework on the MIMIC-III [12] and ADULT [13] datasets, which have been used in prior works on differentially private synthetic data generation. We compare against these prior approaches using a variety of qualitative and quantitative evaluation metrics, including some from prior work and some novel metrics we introduce. We target $\delta = 10^{-5}$ in all settings. All experimental details and our code is available at https://github.com/DPautoGAN/DPautoGAN.

### A. Binary Data

MIMIC-III [12] is a binary dataset consisting of medical records of 46K intensive care unit (ICU) patients over 11 years old with 1071 features. Even though DP-auto-GAN can handle mixed-type data, we evaluate it first on MIMIC-III since this dataset has been used in similar non-private [28] and private [15] GAN frameworks. We apply the same evaluation metrics used in these papers, namely dimension-wise probability and dimension-wise prediction. Prediction is defined by AUROC score of a logistic regression classifier.



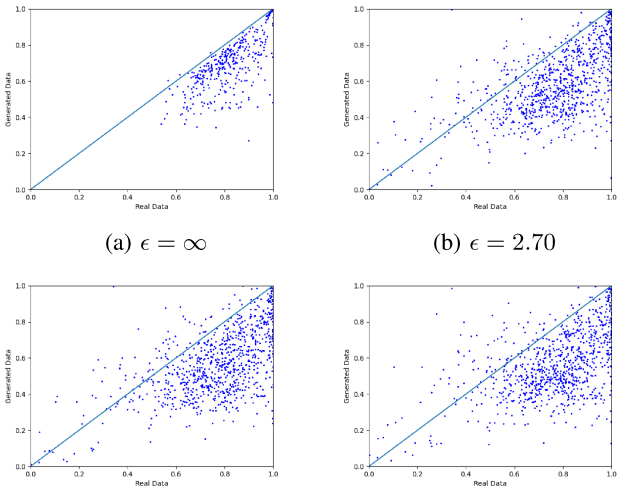

(a) $\epsilon = \infty$ (b) $\epsilon = 2.70$

(c) $\epsilon = 1.33$ (d) $\epsilon = 0.81$

Fig. 2: Dimension-wise probability scatterplots for different values of $\epsilon$. Each point represents one of the 1071 features in the MIMIC-III dataset. The $x$ and $y$ coordinates of each point are the proportion of 1s in real and synthetic datasets of a feature, respectively. The line $y = x$, which represents ideal performance, is shown in each plot. Note that even for small $\epsilon$ values, performance is not degraded much relative to the non-private method. Compare with Figure 4 in [15], which provides worse performance for $\epsilon \in [96, 231]$.

*a) Dimension-Wise Probability:* Figure 2 shows the dimension-wise probability of DP-auto-GAN for different $\epsilon$. Each point in the figure corresponds to a feature in the dataset, and the $x$ and $y$ coordinates respectively show the proportion of 1s in the real and synthetic datasets. Points closer to the $y = x$ line correspond to better performance, because this indicates the distribution is similar in the real and synthetic datasets. As shown in Figure 2, the proportion of 1's in the marginal distribution for is similar on the real and synthetic datasets in the non-private ($\epsilon = \infty$) and private settings. The marginal distributions of the privately generated data from DP-auto-GAN remain a close approximation of the real dataset, even for small values of $\epsilon$, because nearly all points fall close to the line $y = x$. We note that our results are significantly stronger than the ones obtained in [15] with $\epsilon \in [96.5, 231]$ because we obtain dramatically better performance with $\epsilon$ values that are two orders of magnitude smaller. For visual performance comparison, see Figure 4 of [15].

*b) Dimension-Wise Prediction:* Figure 3 shows dimension-wise prediction using DP-auto-GAN for different values of $\epsilon$. Each point in the figure corresponds to a feature in the dataset, and the $x$ and $y$ coordinates respectively show the AUROC score of a logistic regression classifier trained on the real and synthetic datasets, and points closer to the $y = x$ line still correspond to better performance. As shown in the figure, for $\epsilon = \infty$, many points are concentrated along the lower side of line $y = x$, which indicates that the AUROC




(a) $\epsilon = \infty$ (b) $\epsilon = 2.70$

(c) $\epsilon = 1.33$ (d) $\epsilon = 0.81$

Fig. 3: Dimension-wise prediction scatterplots for different values of $\epsilon$. Each point represents one of the 1071 features in the MIMIC-III dataset. The $x$ and $y$ coordinates of each point represent the AUROC score of a logistic regression classifier trained on real and synthetic datasets, respectively. The line $y = x$ corresponds to the ideal performance. Again we note that even for small $\epsilon$ values, performance is not degraded much relative to the non-private method. Compare with Figure 5 in [15], which provides worse performance for $\epsilon \in [96, 231]$.

score of the real dataset is only marginally higher than that of the synthetic dataset. When privacy is added, there is a gradual shift downwards relative to the line $y = x$, with larger variance in the plotted points, indicating that AUROC scores of real and synthetic data show more difference when privacy is introduced. Surprisingly, there is little degradation in performance for smaller $\epsilon$ values, including $\epsilon = 0.81$. For sparse features with few 1's in the data, the generative model will output all 0's for that feature, making AUROC ill-defined. We follow [15] by excluding those features from dimension-wise prediction plots.

Our results for DP-auto-GAN under this metric are also significantly stronger than the ones obtained in [15] with much larger $\epsilon$ values of $\epsilon \in [96.5, 231]$; for visual performance comparison, see Figure 5 of [15]. Our probability and prediction plots of DP-auto-GAN are either comparable to or better than [15], with our prediction plots detecting many more sparse features. The performance of DP-auto-GAN degrades only slightly as $\epsilon$ decreases and is achieved at much smaller $\epsilon$ values, giving a roughly 100x improvement in privacy compared to [15].

### B. Mixed-Type Data

ADULT dataset [13] is an extract of the U.S. Census of 48K working adults, consisting of mixed-type data: nine categorical features (one of which is a binary label) and four continuous. This dataset has been used to evaluate DP-WGAN [14] and DP-SYN [10]. We compare DP-auto-GAN against these methods, as well as DP-VAE [16]. We target $\epsilon = 1.01, 0.51, 0.36$. For DP-SYN, we allow $\epsilon = 1.4, 0.8, 0.5$ because their implementation uses standard privacy composition, which is looser than than RDP composition (Lemma 2). These larger $\epsilon$ values provide comparable privacy guarantees to the smaller $\epsilon$ values achieved by RDP composition, and allow for a fair comparison of architectures without modifying the implementation in [10].

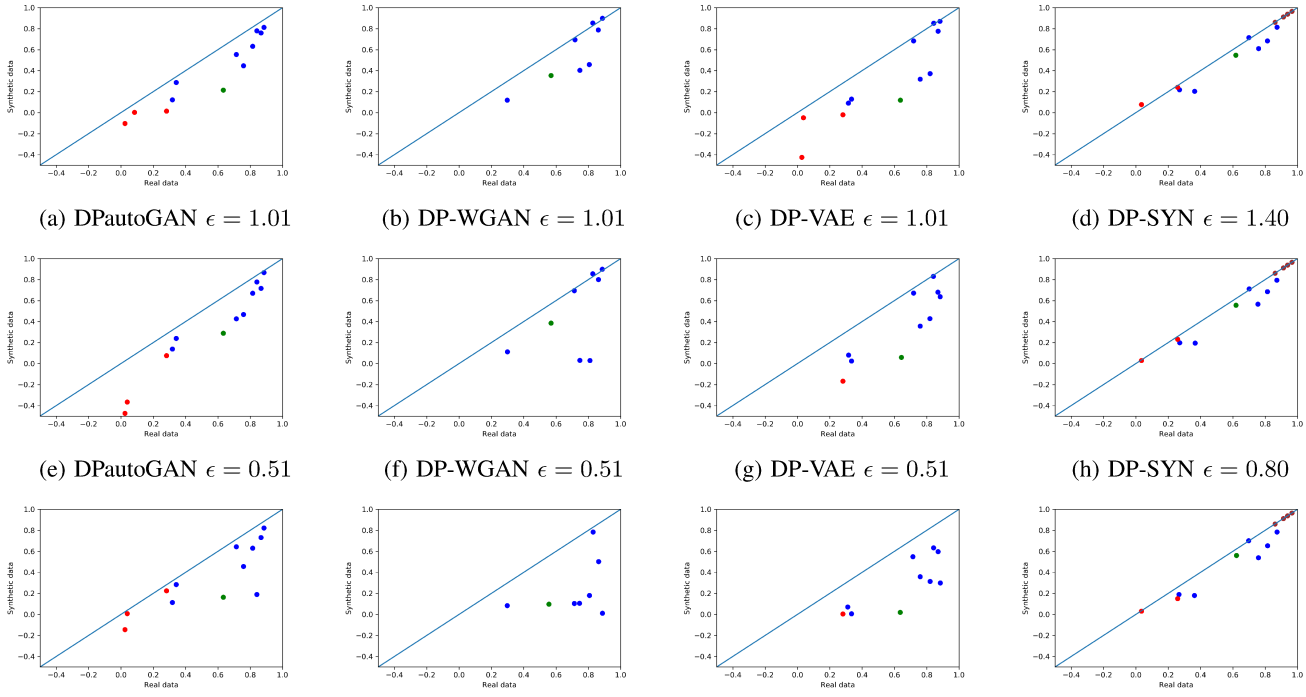
*a) Dimension-Wise Prediction:* Figure 4 compares the performance of DP-auto-GAN with these three prior algorithms for the task of dimension-wise prediction. For categorical features (represented by blue points and a single green point), we use a random forest classifier for prediction as in [14], and we measure performance using $F_1$ score, which is more appropriate than AUROC for multi-class prediction. For continuous features (represented by red points), we used Lasso regression and report $R^2$ scores. The green point corresponds to the *salary feature* of the data, which is real-valued but treated as binary based on the condition $> \$50k$, which was similarly used as a binary label in [14]. We use brown points to indicate the categorical features for which the synthetic data exhibit no diversity, where all synthetic data points have the same category. We explore metrics for measuring diversity later in this section.

Note that in Figure 4, there are not four red points in each plot (corresponding to the four continuous features of the dataset). While AUROC for the binary features is always supported on $[0, 1]$, the $R^2$ score for real-valued features can be negative if the predictive model is poor, and these values for these missing points fell outside the range of Figure 4.

Each point in Figure 4 corresponds to one feature, and the $x$ and $y$ coordinates respectively show the accuracy score on the real data and the synthetic data. Figure 4 shows that DP-auto-GAN achieves considerable performance for all $\epsilon$ values tested. As expected, its performance degrades as $\epsilon$ decreases, but not substantially. DP-WGAN [14] performs well at $\epsilon = 1.01$, but its performance degrades rapidly with smaller $\epsilon$. This is consistent with [14], which uses higher $\epsilon = 3, 7$. DP-auto-GAN outperforms DP-VAE [16] across all $\epsilon$ values. DP-SYN [10] is able to capture relationships between features well even for small $\epsilon$ using this metric.

*b) Random Forest Prediction Scores:* Following [14], we also evaluate the quality of synthetic data by the accuracy of a random forest classifier to predict the label "salary" feature. In particular, we train a random forest classifier on synthetic data and test on the holdout original data, and report the $F_1$ accuracy score. The aim is that a classifier trained on synthetic data should report a similar accuracy score as the one trained on the original data.

In Table I, we report the accuracy of synthetic datasets generated by DP-auto-GAN and DP-WGAN [14]. The results reported in [14] use $\epsilon = 3, 7, \infty$, whereas our algorithms used parameter values $\epsilon = 0.36, 0.51, 1.01, \infty$, a significant improvement in privacy. We see that our accuracy guarantees are higher than those of [14] with smaller $\epsilon$ values, and DP-auto-GAN achieved higher accuracy in the non-private setting. We note that part of the accuracy discrepancy because



(a) DPautoGAN $\epsilon = 1.01$ (b) DP-WGAN $\epsilon = 1.01$ (c) DP-VAE $\epsilon = 1.01$ (d) DP-SYN $\epsilon = 1.40$

(e) DPautoGAN $\epsilon = 0.51$ (f) DP-WGAN $\epsilon = 0.51$ (g) DP-VAE $\epsilon = 0.51$ (h) DP-SYN $\epsilon = 0.80$

(i) DPautoGAN $\epsilon = 0.36$ (j) DP-WGAN $\epsilon = 0.36$ (k) DP-VAE $\epsilon = 0.36$ (l) DP-SYN $\epsilon = 0.50$

Fig. 4: Dimension-wise prediction scatterplot of all (applicable) features of ADULT dataset for different $\epsilon$ values and algorithms. The line $y = x$ represents ideal performance. Blue, green, and red points respectively correspond to unlabeled categorical, labeled binary, and continuous features. Brown points indicate the synthetic data exhibit no diversity (i.e., all data points have the same category). Note that DP-SYN has several features without diversity. Red points with $R^2$ scores close to zero in the original data have unstable (and unmeaningful) synthetic $R^2$ scores due to the sparse nature of those features in the original data, and some of these $R^2$ scores fall outside of the plotted range. The implementation of DP-WGAN in [14] did not allow continuous features, and the implementation of DP-SYN in [10] converted two continuous features to categorical.

TABLE I: Accuracy scores of random forest prediction on salary feature of ADULT dataset by DP-auto-GAN and DP-WGAN in [14] over different privacy parameter $\epsilon$.

| $\epsilon$ value | Real dataset | $\infty$ | 7 | 3 | 1.01 | 0.51 | 0.36 |
|---|---|---|---|---|---|---|---|
| DP-auto-GAN Accuracy (ours) | 84.53% | 79.18% | | | 79.19% | 78.68% | 74.66% |
| DP-WGAN Accuracy | 77.2% | 76.7% | 76.0% | 75.3% | | | |

DP-auto-GAN can handle mixed-typed features, whereas DP-WGAN only handles categorical features.

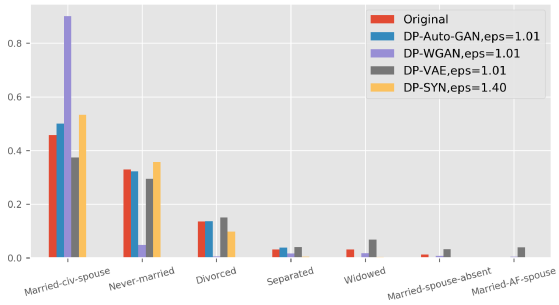
*c) 1-Way Marginal and Diversity Divergence:* While DP-SYN has good dimension-wise prediction, this does not capture *diversity*, a concern of bias known for DP-SGD ( [29]). For features with a large majority class and many minority classes, the classifier often predicts the majority class with probability one. We found that for four features, DP-SYN generates data from only one class, whereas all other algorithms do not behave this way for any feature. Lack of diversity in synthetic data can raise fairness concerns, as societal decisions based on the private synthetic data will inevitably ignore minority groups.

We start by turning to 1-way marginal as a method of evaluation, which is able to detect such issues and give another perspective of synthetic data. Figure 5 shows histograms of synthetic data from the four algorithms on two categorical features: marital-status and race. Marital-status distributes more evenly across categories, and DP-VAE, DP-SYN and DP-auto-GAN are able to learn this distribution well. Race, on the other hand, has an 85.5% majority; DP-SYN only generated data from the majority class, whereas DP-auto-GAN and DP-VAE were able to detect the existence of minority classes. DP-WGAN suffered similar issues on the marital status feature.

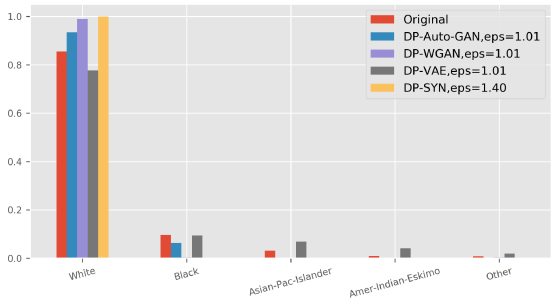
A standard measure for diversity between the original distribution $P$ and synthetic distribution $Q$ includes Kullback-Leibler (KL) divergence $D_{KL}(P||Q)$. Under differential privacy the support of $P$ is a private information, so the private synthetic data inherently cannot ensure its support to align with the original data. This makes $D_{KL}(P||Q)$ and $D_{KL}(Q||P)$ (and related metrics such as Inception score [30]) undefined. One alternative is Jensen–Shannon divergence (JSD) [31], [32]: $\text{JSD}(P||Q) := \frac{1}{2}D_{KL}(P||Q) + \frac{1}{2}D_{KL}(Q||P)$ which is always defined and nonnegative. We use this metric to evaluate the diversity of the synthetic data.

TABLE II: Diversity measures JSD and $D^{\mu}_{KL}$ on different features of ADULT data and the sum of divergences across all eight applicable categorical features (All). Recall that $p_1$ is the maximum probability across all categories of that feature in the original data. Smaller values for the diversity measures imply more diverse synthetic data. For each row (feature), the smallest value for each setting of $\epsilon$ is highlighted in bold.

| | **DP-auto-GAN** | | | **DP-WGAN** | | | **DP-VAE** | | | **DP-SYN** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *$\epsilon$ values* | *0.36* | *0.51* | *1.01* | *0.36* | *0.51* | *1.01* | *0.36* | *0.51* | *1.01* | *0.50* | *0.80* | *1.40* |
| JSD Diversity Measure | | | | | | | | | | | | |
| Marital | .025 | .043 | **.014** | .119 | .624 | .136 | .139 | .043 | .021 | **.017** | **.013** | .017 |
| Race | **.021** | **.014** | .016 | .081 | .053 | .040 | .095 | .031 | **.011** | .053 | .053 | .053 |
| All | 0.33 | **0.23** | **0.19** | 1.29 | 2.41 | 0.73 | 0.80 | 0.44 | 0.23 | **0.25** | 0.27 | 0.28 |
| $D^{\mu}_{KL}$ Diversity Measure, with $\mu = e^{-\frac{1}{1-p_1}}$ | | | | | | | | | | | | |
| Marital | .019 | .053 | **.005** | .165 | 1.16 | .290 | .207 | .044 | .017 | **.017** | **.011** | .012 |
| Race | **.125** | **.064** | .089 | .262 | .465 | .277 | .315 | .102 | **.038** | .465 | .465 | .465 |
| All | **0.81** | **0.48** | **0.53** | 5.26 | 6.39 | 1.53 | 2.52 | 1.17 | 0.58 | 0.99 | 1.00 | 1.02 |




(a) Marital-status




(b) Race

Fig. 5: Histograms of synthetic data generated by different algorithms.

In addition, we propose another diversity measure, *$\mu$-smoothed Kullback-Leibler (KL) divergence* between the original distribution $P$ and synthetic distribution $Q$:

$$D^{\mu}_{KL}(P||Q) := \sum_{x \in supp(P)} (P(x) + \mu) \log(\frac{P(x)+\mu}{Q(x)+\mu}),$$

for small $\mu > 0$. $D^{\mu}_{KL}$ maintains the desirable property that $D^{\mu}_{KL} \geq 0$ and is zero if and only if $P = Q$. Smaller $\mu$ implies stronger penalties for missing minority categories in the synthetic data, and the penalty approaches $\infty$ as $\mu \to 0$. This allows $\mu$ as a knob to adjust the penalty necessary in private setting. In our settings, we are concerned with one category dominating in the original distribution $P$ (e.g., as in Figure 5), say $P = (p_1, \ldots, p_k)$ with high $p_1 = \max_i p_i$, and when the synthetic distribution $Q = (q_1, \ldots, q_k) = (1, 0, \ldots, 0)$ supports only one single category. Then, we have $D^{\mu}_{KL}(P||Q) = \sum_{i=2}^{k}(p_i + \mu)\log(p_i + \mu) - (p_1 + \mu)\log(\frac{p_1+\mu}{1+\mu}) - (\sum_{i=2}^{k}(p_i + \mu))\log\mu$. For small $\mu > 0$, $\log\mu$ dominates $\log(p_i + \mu)$ and $\log(\frac{p_1+\mu}{1+\mu})$, so the dominating term is $\left(\sum_{i=2}^{k}(p_i + \mu)\right)\log\mu \approx (1 - p_1)\log\mu$. Hence, we use $\mu = e^{-\frac{1}{1-p_1}}$ so that this term is a constant, thus normalizing scores across features.

Table II reports the diversity divergences of all four algorithms for marital-status, race, and the sum across eight categorical features. One of the nine categorical features are not used due to a difference in preprocessing of DP-SYN. Both measures are able to detect the lost of diversity in DP-SYN in race, and identify DP-auto-GAN as generating more diverse data than the prior methods for most features and $\epsilon$ values.

We note that predictive scores may also not be appropriate for continuous features when no good classifier exists to predict the feature, even in the original dataset. In our setting, we found three continuous features with $R^2$ scores close to zero even with more complex regression models, and with negative $R^2$ scores on synthetic data, which is not meaningful. For those features, 1-way marginals (histograms, explored next) are preferred to prediction scores.

In general, we suggest that an evaluation of synthetic data should be based on probability measures (distributions of data) and not predictive scores of models. Models may be a source of not only unpredictability and instability, but also of bias and unfairness.

## V. Conclusion

We propose DP-auto-GAN—a combination of DP-autoencoder and DP-GAN—for differentially private data generation of mixed-type data. The inclusion of the autoencoder improves the efficacy of GANs, especially for high-dimensional data. Our method enjoys a 5x privacy improvement compared to [14] on the ADULT dataset in 14 dimensions and greater 100x improvement compared to [15] on a higher 1071-dimensional dataset, and achieves a meaningful privacy $\epsilon < 1$ for practical use. This approach is more complex than assuming a standard Gaussian distribution as in DP-VAE [16], and is better able to learn relationships among features.

## REFERENCES


[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, ser. Oakland S&P '08, 2008, pp. 111–125.

[2] M. Barbaro and T. Zeller, "A face is exposed for AOL searcher no. 4417749," New York Times, August 9 2006, [Online, Retrieved 9/25/2019]. [Online]. Available: https://www.nytimes.com/2006/08/09/technology/09aol.html

[3] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA Law Review*, vol. 57, pp. 1701–1777, 2010.

[4] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *Proceedings of the 28th USENIX Security Symposium*, ser. USENIX Security '19, 2019, pp. 267–284.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography*, ser. TCC '06, 2006, pp. 265–284.

[6] A. Triastcyn and B. Faltings, "Generating artificial data for private deep learning," in *Proceedings of the PAL: Privacy-Enhancing Artificial Intelligence and Language Technologies*, ser. PAL '18, 2018, pp. 33–40.

[7] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, ser. STOC '08, 2008, pp. 609–618.

[8] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '10, 2010, pp. 61–70.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, arXiv preprint 1312.6114.

[10] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Machine Learning and Knowledge Discovery in Databases (ECML PKDD '18)*, ser. Lecture Notes in Computer Science. Springer, 2018, vol. 11051, no. 1, pp. 510–526.

[11] Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, and H. Zhu, "Differentially private data generative models," 2018, arXiv preprint 1812.02274.

[12] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.

[13] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[14] L. Frigerio, A. S. de Oliveira, L. Gomez, and P. Duverger, "Differentially private generative adversarial networks for time series, continuous, and discrete open data," in *International Conference on ICT Systems Security and Privacy Protection*, ser. IFIP SEC '19, 2019, pp. 151–164.

[15] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," 2018, arXiv preprint 1802.06739.

[16] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1109–1121, 2018.

[17] M. Hardt, K. Ligett, and F. McSherry, "A simple and practical algorithm for differentially private data release," in *Advances in Neural Information Processing Systems 25*, ser. NIPS '12, 2012, pp. 2339–2347.

[18] M. Gaboardi, E. J. G. Arias, J. Hsu, A. Roth, and Z. S. Wu, "Dual query: Practical private query release for high dimensional data," in *Proceedings of the 31st International Conference on Machine Learning*, ser. ICML '14, 2014, pp. 1170–1178.

[19] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private data release via Bayesian networks," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 4, p. 25, 2017.

[20] H. Ping, J. Stoyanovich, and B. Howe, "DataSynthesizer: Privacy-preserving synthetic datasets," in *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, ser. SSDBM '17, 2017, pp. 42:1–42:5.

[21] H. Surendra and H. Mohan, "A review of synthetic data generation methods for privacy preserving data publishing," *International Journal of Scientific and Technology*, vol. 6, pp. 95–101, 2017.

[22] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, ser. CCS '16, 2016, pp. 308–318.

[23] I. Mironov, "Rényi differential privacy," in *Proceedings of the 2017 IEEE 30th Computer Security Foundations Symposium*, ser. CSF '17, 2017, pp. 263–275.

[24] M. Park, J. Foulds, K. Choudhary, and M. Welling, "DP-EM: Differentially Private Expectation Maximization," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. AISTATS '17, 2017, pp. 896–904.

[25] X. Zhang, S. Ji, and T. Wang, "Differentially private releasing via deep generative model (technical report)," 2018, arXiv preprint 1801.01594.

[26] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, arXiv preprint 1701.07875.

[27] Google, "Tensorflow privacy," 2018. [Online]. Available: https://github.com/tensorflow/privacy

[28] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Proceedings of Machine Learning for Healthcare*, 2017, pp. 286–305.

[29] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Advances in Neural Information Processing Systems 32*, ser. NeurIPS '19, 2019, pp. 15 479–15 488.

[30] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems 29*, ser. NIPS '16, 2016, pp. 2234–2242.

[31] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, ser. NIPS '14, 2014, pp. 2672–2680.