Differentially Private Small Dataset Release Using Random Projections

Lovedeep Gondara **Ke Wang**
Department of Computing Science
Simon Fraser University

Abstract

Small datasets form a significant portion of releasable data in high sensitivity domains such as healthcare. But, providing differential privacy for small dataset release is a hard task, where current state-of-the-art methods suffer from severe utility loss. As a solution, we propose DPRP (Differentially Private Data Release via Random Projections), a reconstruction based approach for releasing differentially private small datasets. DPRP has several key advantages over the state-of-the-art. Using seven diverse real-life datasets, we show that DPRP outperforms the current state-of-the-art on a variety of tasks, under varying conditions, and for all privacy budgets.

1 INTRODUCTION

1.1 MOTIVATION AND PROBLEM STATEMENT

Publicly available data aids reproducibility and promotes new discoveries. However, sharing data "as-is" can lead to privacy breaches with severe personal and legal consequences, especially in high sensitivity domains such as healthcare [1]. The dilemma of data sharing while protecting an individual's privacy often leads to ad hoc measures of data sanitization, such as removing primary identifiers (name, date of birth, social insurance number, etc.), and/or arbitrary binning or rounding of variables. Such data sanitization practices are ineffective and combining multiple such releases, an adversary can accumulate information about an individual, leading to uncontrolled privacy leakage or worse, a complete disclosure [2].

1.2 CURRENT APPROACH

Differential privacy [3] offers a solution. Formalizing the notion of privacy as a mathematical definition, differential privacy promises any released data will not unduly disclose any information about an individual. However, direct application of differential privacy to "raw data" release is non-trivial. Recently, Generative Adversarial Networks (GANs) [4] have been studied as a potential solution. Trained on real data, GANs learn the input data distribution and can be used for sampling from the learned distribution. This guarantees the sampled data is "synthetic", but still follows the distribution similar to real data. But as GANs are trained unconstrained on real data, they can implicitly or explicitly disclose sensitive information contained in the training set [5] (Example: A GAN trained on medical records can leak sensitive information about patients in the training set). Acknowledging this issue, recent approaches have proposed combining differential privacy with GANs [6, 7], where the generated data follows the real data distribution and is differentially private. Differentially private GANs, however, fall severely short on the utility front, especially for small-size datasets, which form a significant portion of releasable data in high sensitivity domains.

1.3 CHALLENGES WITH CURRENT STATE-OF-THE-ART

The typical size of structured tabular datasets in many high sensitivity domains such as healthcare is only a few hundred. In special settings such as in the clinical trials, the dataset size can be even smaller as collecting patient data is costly and to avoid unnecessary patient harm, most studies are designed with *minimum* sample size requirements.

But GANs, based on two contesting neural networks, require large training datasets for effectively capturing the data generating distribution. Combined with the noise addition required for preserving differential pri-

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

vacy, where the noise scale is inversely proportional to the dataset size (smaller datasets require larger noise for a constant privacy budget, based on the trade-off between sampling ratio, minibatch size, and the number of training iterations, see Theorem 1 of Abadi et al. [8], the current state-of-the-art training paradigm for differentially private neural networks), it leads to severe utility loss for small datasets (our extensive empirical evaluation in Section 4 supports this claim). The recent approach of using the PATE framework for differentially private GANs [7] fares even worse as it requires splitting the dataset into *many* disjoint partitions before training, leaving little to learn from each partition in case of small datasets[1]. Moreover, PATE-GAN only works with binary outcomes, limiting the use in case of multi-class datasets. There are other "non-deep learning" methods that strive to achieve similar release goals. But, either they can only release projections [9], aggregated statistics/histograms [10], or are computationally infeasible, fail to capture higher-order interactions, and only work with binary outcomes [11]. This leaves the GANs trained via noisy stochastic gradient descent the current state-of-the-art generic models for differentially private data release, leaving a void for the availability of a suitable generic method that can guarantee the differentially private release of small datasets while preserving the dataset's inferential utility.

1.4 PROPOSED SOLUTION

In light of the issues discussed above with current state-of-the-art differentially private models, we now propose our solution. Our method extends non-private image compression and reconstruction techniques exploiting low-rank approximation [12, 13] to differentially private release of small, tabular datasets. Our method is a *model-free* approach, offering *one-shot* reconstruction, we call it DPRP (Differentially Private Data Release via Random Projections). DPRP's formulation, the ability for one-shot reconstruction, and a minimal amount of hyperparameter tuning contribute to its superior utility. Specifically, DPRP takes the original dataset (X) as input, computes the Singular Value Decomposition (SVD) of the covariance matrix X_C of X, and then uses the right singular vector ($\hat{V}^T$) in conjunction with a random projection P across the columns to reconstruct $X' \approx X$. For preserving differential privacy, we ensure that $\hat{V}^T$ and P are differentially private (the only instances of real data needed for reconstruction). Complete details of DPRP are presented in Section 3, and the overall schema of DPRP is shown in Figure 1. To summarize, our main contributions are as follows.

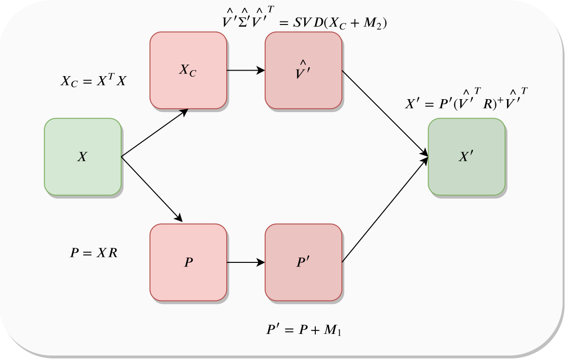
$\hat{V}'\hat{\Sigma}'\hat{V}'^T = SVD(X_C + M_2)$
$X_C = X^T X$
X_C
$\hat{V}'$
$X' = P'(\hat{V}'^T R)^+ \hat{V}'^T$
X
X'
$P = XR$
P
P'
$P' = P + M_1$

Figure 1: DPRP schema: Using X, we create a random projection P across the columns, and the covariance matrix X_C. We decompose the covariance matrix using SVD. Using noisy right singular vector ($\hat{V}'$) from the decomposition along with noisy P', we reconstruct $X' \approx X$. Details on noise addition (M_1, M_2) for differential privacy and the reconstruction are explained in detail in Section 3, Algorithm 1.

1. We propose DPRP, a model-free, reconstruction based approach for releasing differentially private small datasets, a utility bottleneck for state-of-the-art generative models.

2. Being a model-free approach, DPRP is easy to implement, computationally cheap, and offers a *one-shot* reconstruction. DPRP also avoids extensive hyperparameter optimization, often required in deep generative models.

3. Finally, our extensive empirical evaluation on seven diverse real-life datasets shows that DPRP outperforms state-of-the-art differentially private generative models by a significant margin.

2 PRELIMINARIES

Here we introduce preliminary concepts central to the rest of our work.

2.1 DIFFERENTIAL PRIVACY

Differential privacy [3] provides us with formal and provable privacy guarantees with the intuition that a randomized algorithm behaves similarly on similar input datasets, where we define the dataset similarity using

[1] PATE-GAN's utility is directly proportional to the number of disjoint data partitions as every partition is used to train a separate discriminator. A dataset with 100 observations split into 10 partitions will result in each discriminator training on 10 observations. This issue is compounded when we have slight class imbalance as some partitions will not have members of some classes, making the associated discriminator incapable of learning that class distribution.

the notion of *neighbouring* datasets. That is the datasets which differ on any one row, formally

Definition 1. (Differential privacy [3]) *A randomized mechanism* $\mathcal{M} : D^n \rightarrow \mathbb{R}^d$ *preserves* (ϵ, δ)*-differentially privacy if for any pair of neighbouring databases* $(x, y \in D^n)$ *such that* $d(x, y) = 1$*, and for all sets* $\mathcal{S}$ *of possible outputs:*

$$Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

Intuitively, Definition 1 states that for any pair of two neighboring datasets, x, y, a randomized mechanism $\mathcal{M}$'s outcome does not change by more than a multiplicative factor of e^{ϵ} and the guarantee fails with probability no larger than δ. If $\delta = 0$, we have pure-ϵ differential privacy.

Differential privacy has many intriguing properties such as the *post-processing*, which tells us if an algorithm protects an individual's privacy via differential privacy, no external adversaries acting only on the algorithm's output can increase the privacy loss. Formally,

Theorem 1. (Post processing [14]) *Let* $\mathcal{M} : D^n \rightarrow \mathbb{R}^d$ *be a randomized mechanism that is* (ϵ, δ)*-differentially private. Let* $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ *be a deterministic function. Then* $f \circ \mathcal{M} : D^n \rightarrow \mathbb{R}^{d'}$ *is* (ϵ, δ)*-differentially private.*

2.2 RANDOM PROJECTIONS

Random projection is a method to project original d-dimensional data to a k-dimensional subspace ($k \neq d$ usually) through the origin, using a random $k \times d$ matrix. The idea for random projections originates from the much-celebrated Johnson-Lindenstrauss Lemma [15].

Lemma 1. (Johnson-Lindenstrauss Lemma [15]) *Let* $\nu \in (0, 1/2)$*. Let* $Q \subset \mathbb{R}^d$ *be a set of* n *points and* $k = \frac{20 \log n}{\nu^2}$*. There exist a Lipschitz mapping* $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$*, such that for all* $u, v \in Q$*, we have*

$$(1 - \nu)||u - v||_2^2 \leq ||f(u) - f(v)||_2^2 \leq (1 + \nu)||u - v||_2^2$$

In essence, the lemma states if the points in a vector space are randomly projected to a suitable space of high enough dimensions, the approximate distance between them is preserved. To project $X^{(n\times d)}$ to a k-dimensional subspace, we create a random matrix $R^{(d\times k)}$ and take the product, that is XR. There are several methods to create R, but here we focus on a simple Gaussian R, where the entries are drawn from $\mathcal{N}(0, 1/\sqrt{k})$ [16].

3 DPRP

We first introduce DPRP, our proposed method for releasing differentially private small datasets. Then we proceed to state and prove DPRP's formal privacy guarantees.

3.1 DPRP OVERVIEW

DPRP takes inspiration from non-private image compression and reconstruction techniques [12, 13] based on the low-rank approximation, and further extends the idea for the differentially private reconstruction of small tabular datasets. DPRP constitutes a *model-free* approach, whereby no parameter estimation of any sort is required, leading to minimal hyperparameter tuning and no iterative learning process. Due to its reconstruction based nature, DPRP works extremely well on small datasets (a performance bottleneck for current state-of-the-art). We present DPRP succinctly as Algorithm 1 and provide a line by line walkthrough for the readers.

```
Algorithm 1: DPRP: Differentially Private Reconstruction of Input Data
  Input: Dataset:$X$; Privacy parameters: $\epsilon, \delta$;
         Privacy budget allocation: $b_1$% for
         random projection $P$, $1 - b_1$% for
         SVD($X_C$); Number of dimensions for
         random projection $P$: $k_1$; Number of
         values from right singular vector to keep
         from SVD($X_C$): $k_2$
  Output: Differentially private dataset: $X'$
1 $R \sim \mathcal{N}(0, 1/\sqrt{k_1})^{d\times k_1}$
2 $P = XR$
3 $P' = P + M_1; M_1 \sim \mathcal{N}(0, \sigma_1^2)^{n\times k_1}$ // With
    budget $b_1$%
4 $X_C = X^T X$
5 $\hat{V}'\hat{\Sigma}'\hat{V}'^T$ = SVD($X_C$+$M_2$); $M_2 \sim \mathcal{N}(0, \sigma_2^2)^{d\times d}$
    // With budget $1 - b_1$%
6 $V'_{k_2} = \hat{V}'[1, \cdots, k_2]$ // First $k_2$ columns
7 $X' = P'(\hat{V'_{k_2}}^T R)^+ \hat{V'_{k_2}}^T$
```

To start, the user provides the dataset $X^{n\times d}$ as an input to DPRP, along with the overall privacy budget, ϵ, δ; the allocation of the privacy budget, that is the share of the privacy budget for making the random projection P differentially private (b_1%) and the share of the privacy budget for the differentially private SVD ($1 - b_1$%); dimensionality of random projection, P, k_1; and the number of values from the right singular vector from $\hat{V}$ to use for the reconstruction, k_2.

Line 1 - 2 (Creating random projection): We start with creating the random projection, where we create a random matrix $R^{d\times k_1}$ with entries drawn from $\mathcal{N}(0, 1/\sqrt{k_1})$ and create the projection $P^{n\times k_1} = XR$. We have to remember that up to this point, we have not made any differential privacy claims, so P still contains sensitive information from X.

Line 3 (Differential privacy of P): To ensure differential privacy of P, we add a noise matrix M_1 ($P' = P + M_1$). Specifically, $M_1 \sim \mathcal{N}(0, \sigma_1^2)$ for some σ_1. Where σ_1 is chosen using Theorem 2.

Line 4-6 (Differential privacy of SVD(X_C)): For the reconstruction of X, we only need the right singular vector of decomposed X. But as X contains sensitive information, so will the right singular vector from decomposed X. Making the right singular vector differentially private is non-trivial. We do not directly add noise to the right singular vector as it can lead to an overly noisy result and the right singular vector does not directly relate to the "per-user" principle of differential privacy. We follow a different approach [17], where we first calculate the covariance matrix ($X_C = X^T X$), and then add noise to ensure the differential privacy of the covariance matrix ($X'_C = X_C + M_2$), where $M_2 \sim \mathcal{N}(0, \sigma_2^2)$ and σ_2 is chosen according to Theorem 2. Then we perform the singular value decomposition on X'_C and choose the first k_2 values from the right singular vector (V'_{k_2}).

Line 7 (Noisy reconstruction): Now, for the *main* part, we perform our *noisy* reconstruction of X. "$+$" refers to the Moore-Penrose pseudoinverse. It is noteworthy that only noisy P' and $\hat{V'_{k_2}}$ are required for the reconstruction, which we have earlier made differentially private, in addition to a *random* matrix R, which does not have any real data, leading to a differentially private reconstruction, X'.

We discuss some aspects of Algorithm 1 in Section 3.3. But, first, we provide the differential privacy guarantees of our reconstruction, as it remains to be shown that adding noise (M_1, M_2) to (P, X_C), and reconstructing X results in a differentially private output.

3.2 PRIVACY GUARANTEES OF DPRP

Before we state our main privacy guarantees, we start with two supporting Lemmas.

Lemma 2. [18] *For two neighbouring datasets X and X' that only differ in one observation, i, with $||X_i - X'_i|| \leq Z$, and a random Gaussian matrix P with entries drawn from $\mathcal{N}(0, \sigma_p^2)$, where $\sigma_p = 1/\sqrt{k_1}$. With probability at least $1 - \delta$, we have*

$$||XP - X'P||_F \leq Z\sigma_p \sqrt{k_1 + 2\sqrt{k_1 \log(1/\delta)} + 2\log(1/\delta)}$$

Proof. Proof is from [18], provided here for completeness[2]. Since X and X' only differ on one row i, we can write

$$(XP - X'P)_{mn} = 0, m \neq i \quad (1)$$

and

$$\begin{aligned}(XP - X'P)_{ij} &= < X_i, P_j > - < X'_i, P_j > \\ &= < X_i - X'_i, P_j > \quad (2)\end{aligned}$$

where P_j is the jth column of P and $< .,. >$ denotes the inner product. Let $z = X_i - X'_i$. We have

$$< z, P_j > \sim \mathcal{N}(0, ||z||^2\sigma_p^2) \quad (3)$$

using the scaling properties of Gaussians(i.e. for constants a, b; $X = \mathcal{N}(0, \sigma_x^2), Y = \mathcal{N}(0, \sigma_y^2)$, we have $aX + bY = \mathcal{N}(0, a^2\sigma_x^2 + b^2\sigma_y^2)$.

Let $Y_j = \mathcal{N}(0, 1)$ and $\chi^2_{k_1}$ denote a chi-squared random variable with k_1 degrees of freedom. We can bound the matrix norm as

$$\begin{aligned}||XP - X'P||_F &= \sqrt{\sum_{j=1}^{k_1} < z, P_j >^2} \\ &= \sqrt{\sum_{j=1}^{k_1} (||z||\sigma_p Y_j)^2} \\ &= ||z||\sigma_p \sqrt{\chi^2_{k_1}}\end{aligned} \quad (4)$$

second equality follows from $X \sim \mathcal{N}(0, \sigma^2), X/\sigma \sim \mathcal{N}(0, 1)$. Using Lemma 1 from [19], we can get the following tail bound on a random variable X, drawn from a χ^2 distribution with k_1 degrees of freedom

$$Pr[X \geq k_1 + 2\sqrt{k_1 x} + 2x] \leq \exp(-x) \quad (5)$$

setting $x = \log(1/\delta)$ completes the proof. □

Lemma 3. [20] *The mechanism $M(D) = f(D) + G$, where G is a random Gaussian matrix with entries drawn from $\mathcal{N}(0, \sigma_1^2)$, satisfies (ϵ, δ) - differential privacy, if $\delta < \frac{1}{2}$, where $\sigma_1^2 = 2\Delta_2(f)^2(\log(1/2\delta) + \epsilon)/\epsilon^2$ and $\Delta_2(f)$ is the sensitivity*

With the support of the two lemmas above, we are ready to state our main Theorem.

Theorem 2. *Algorithm 1 is (ϵ, δ)- differentially private, for $\epsilon > 0, 0 < \delta < 1/2$.*

Proof. DPRP has two main components where we add noise (to the random projection P and the covariance matrix X_C). We conduct the proof in two parts by sequentially proving the differential privacy of each of the parts. Starting with proving *user-level* differential privacy for the random projection, P.

[2] We reproduce the proof as the source document is unstable

Theorem A P' is (ϵ_1, δ_1)-differentially private if we add noise from $\mathcal{N}(0, \sigma_1^2)$; where

$$\sigma_1 = Z\sigma_p\sqrt{k_1 + 2\sqrt{k_1 \log(2/\delta_1)} + 2\log(2/\delta_1)}$$
$$\sqrt{2(\log(1/2\delta_1) + \epsilon_1)}/\epsilon_1$$

Proof. Proof is from [18], summarized next for completeness. Replacing $\Delta_2(f)$ in Lemma 3 with RHS from Lemma 2, and with $\delta/2$, we get

$$\sigma_1 = Z\sigma_p\sqrt{k_1 + 2\sqrt{k_1 \log(2/\delta_1)} + 2\log(2/\delta_1)}$$
$$\sqrt{2(\log(1/2\delta_1) + \epsilon_1)}/\epsilon_1 \quad (6)$$

which guarantees differential privacy of P' if we add a noise matrix M_1 with noise drawn from $\mathcal{N}(0, \sigma_1^2)$, i.e. $P' = P + \mathcal{N}(0, \sigma_1^2)$ (privacy guarantees follow from Lemma 3). Here k_1 are the number of dimensions for the random projection and Z is the bound on the L_2 sensitivity of the input. □

Theorem B $\hat{V}'$ is (ϵ_2, δ_2)- differentially private if we add noise to X_C from $\mathcal{N}(0, \mathcal{Z}^2\sqrt{2\ln 1.25/\delta_2}/\epsilon_2)$

Proof. For providing differential privacy for X_C and subsequently extending it to its singular value decomposition and hence $\hat{V}$, we follow the steps of [17], where we add Gaussian noise to each entry of X_C. Specifically,

$$X'_C = X_C + M_2 \quad (7)$$

where M_2 is a $d \times d$ symmetric matrix whose upper triangular values are chosen from

$$\mathcal{N}(0, \mathcal{Z}^2\sqrt{2\ln 1.25/\delta_2}/\epsilon_2) \quad (8)$$

and lower triangular entries are copied from their upper triangular counterparts. Here $\mathcal{Z}$ is the L_2 sensitivity required for the Gaussian mechanism[3]. Differential privacy guarantees for the above follow directly from [17] and as differential privacy is closed under post-processing [14], we can perform the decomposition on X'_C to get $\hat{V}'$ without any additional privacy loss. □

Using sequential composition[14], we get the Algorithm 1 as (ϵ, δ)- differentially private, where $\epsilon = \epsilon_1 + \epsilon_2$ and $\delta = \delta_1 + \delta_2$. □

3.3 DISCUSSION

In the light of the Algorithm 1 and Theorem 2, we use this section to highlight some important aspects of DPRP. We begin with re-emphasizing the significance of the simplicity of DPRP in the time of highly complex deep learning methods. Specifically, DPRP does not require any iterative procedure or any parameter estimation, DPRP offers a one-shot reconstruction with superior utility compared to the current state-of-the-art deep generative models for small datasets. Based on standard, simple matrix operations, it is easy to implement DPRP, making it readily accessible and deployable in real-world scenarios.

Compared to the extensive hyperparameter optimization required in deep generative models, DPRP requires minimal hyperparameter tuning. DPRP requires fine-tuning of the privacy budget allocation to the random projection and singular value decomposition; the number of dimensions in the random projection P; and the number of values from right singular vector $\hat{V}$ to be used for the reconstruction. DPRP's utility depends on the values of the hyperparameters above, most importantly on the value of k_1, as k_1 has a direct impact on our noise scale, where from Eqn. (6), we can observe that large k_1 can diminish noise ($\sigma_p = {}^1/_{\sqrt{k_1}}$) impact. Hence, for empirical evaluation, we use k_1 as our main hyperparameter. For the rest, we show that even using sensible defaults, DPRP outperforms current state-of-the-art generative models by a wide margin.

In regards to the utility, reconstruction by itself (without differential privacy) is near optimal with the reconstruction error proportional to the compression ratio (that is the number of values from the right singular vector $\hat{V}$) [13]. We incur additional utility loss by transitioning into the differential privacy paradigm, however, in our extensive empirical evaluation on seven diverse real-life datasets, we show that our method provides significantly better utility compared to the current state-of-the-art.

4 EXPERIMENTS

In this section, we present empirical evidence on seven real-life datasets to support our earlier claims that for small datasets, DPRP outperforms state-of-the-art differentially private generative models, both in terms of utility and privacy, while simultaneously being computationally cheap.

4.1 DATASETS

For evaluating the performance of DPRP and its comparison with state-of-the-art generative models, we use seven real-life, publicly available datasets from the UCI machine learning repository. Datasets are carefully selected to evaluate DPRP on small sized datasets of varying dimensionality, a utility bottleneck for current state-

[3] Based on the L_2 norm, can be ensured by appropriate scaling of input.

of-the-art differentially private generative models. Table 1 shows the basic dataset characteristics. Any missing values are replaced by their respective column-wise averages.

Dataset	Attributes	Instances	Type
Coimbra BC	10	116	Binary
Wisconsin BC	32	569	Binary
Indian Liver	10	583	Binary
Dermatology	33	366	Multiclass
Cervical Cancer	36	858	Binary
Caesarian	5	80	Binary
HCC	49	165	Binary

Table 1: Datasets used for DPRP evaluation. Type specifies the output type (binary classification vs multi-class classification).

4.2 PRIVACY PARAMETERS

For privacy parameters, δ is kept fixed at 0.0001 with ϵ varied as required and reported in the following sections. For DPRP, we have $(\epsilon_1, \epsilon_2, \delta_1, \delta_2)$, which compose for our overall privacy budget. We dedicate 80% of our privacy budget to (ϵ_1, δ_1), that is, the privacy parameters for the differentially private random projection P' and 15% for (ϵ_2, δ_2), that are the privacy parameters for differentially private right singular vector $\hat{V}'$ (obtained via differentially private SVD). We reserve 5% of the privacy budget to select *good* values for k_1, which can be done in more than one way, where one can directly account for hyperparameter search, similar to [8], or use the exponential mechanism to probabilistically select a best setting with score being the outcome, we use the former. Privacy budget allocation is an interesting property of DPRP, we study this in detail in Section 4.8.

4.3 DPRP

DPRP being a reconstruction based, model-free approach, does not have a very complicated setup, making it accessible to a wider audience. DPRP can be implemented using any statistical or machine learning workflow. Before inputting to DPRP, all datasets are normalized by their respective row-wise L_2 norm. For the initial parameters, in addition to selecting the best k_1, for choosing the number of values k_2 from the right singular vector, we use $k_2 = 0.6d$. That is, our initial reconstructions are based on optimal P' and constrained $\hat{V}'$. We further investigate this phenomenon in Section 4.7.

4.4 COMPETITORS

For comparison, we consider two state-of-the-art differentially private GANs. The DPGAN [6] and the DP-CGAN [21][4]. For the underlying GAN model, we use the Wasserstein GAN [22]. Minibatch size is kept fixed at 50 and 100 (50 for datasets with total sample size less than 200 and 100 for the rest, done to account for small dataset sizes). The discriminator and the generator are both fully connected neural networks with a depth of three. Generator's layers are of size $(d, d/2, d)$ respectively, where d is the input data's dimensionality. RMSProp is used as the optimizer with a learning rate for the discriminator and the generator set at 0.001. GANs are trained for 100 epochs.

4.5 COMPARISON

For comparison, we use the "train on synthetic-test on real" approach, where the real dataset is first split into train and test partitions ($^{80}/_{20}$ split). Train partition is used to train GANs and for the reconstruction in DPRP. Resulting differentially private datasets are then used to train a random forest model (built using Scikit Learn with default parameters), performance of which is tested using the hold-out test set (real data). For evaluation, we run the models 50 times and report the average results. For quantitative comparison, we carefully chose the metrics that best reflect the overall output quality. Specifically, we use the Area Under the Precision-Recall Curve (AUPRC) and Classification Accuracy (Acc). To provide an upper bound on the performance of DPRP and GANs, we also include performance metrics on the real, non-perturbed, non-noisy datasets.

4.6 MAIN COMPARISON

Table 2 shows our main comparison using Area Under the Precision-Recall Curve (AUPRC) and Classification Accuracy (Acc). With the aid of the results, we make several key observations. First, and the most important observation is that DPRP outperforms DPGAN and DP-CGAN on all datasets and for all privacy budgets by a significant margin, both in terms of AUPRC and Accuracy. This provides concrete evidence to our earlier claims that for small sized datasets, which are a common occurrence in high sensitivity domains such as healthcare, differentially private data generated using DPRP outperforms data generated by the state-of-the-art GANs in terms of utility while simultaneously providing tight privacy guarantees.

[4] Recently, PATE-GAN [7] has been shown to slightly outperform DPGAN on large binary outcome data. We were, however, unable to replicate the baselines reported in the PATE-GAN paper and no source code is made available at the time of this submission. Hence, PATE-GAN is not included as a direct competitor for binary outcome datasets. But, the margin of improvement offered by DPRP over DPGAN is greater than that reported in the PATE-GAN results (PATE-GAN vs DPGAN).

Data	Method	$\epsilon = 8$ (Acc,AUPRC)	$\epsilon = 6$ (Acc,AUPRC)	$\epsilon = 4$ (Acc,AUPRC)	$\epsilon = 2$ (Acc,AUPRC)	$\epsilon = 1$ (Acc,AUPRC)
Coimbra BC	DPRP	**0.56, 0.63**	**0.54, 0.62**	**0.52, 0.59**	**0.51, 0.59**	**0.51, 0.56**
	DPGAN	0.52, 0.53	0.50, 0.54	0.50, 0.54	0.48, 0.53	0.46, 0.53
	DP-CGAN	0.51, 0.51	0.51, 0.50	0.49, 0.50	0.45, 0.49	0.44, 0.46
	No Privacy	0.94, 0.69	0.94, 0.69	0.94, 0.69	0.94, 0.69	0.94, 0.69
Wisconsin BC	DPRP	**0.71, 0.74**	**0.69, 0.68**	**0.68, 0.65**	**0.63, 0.61**	**0.58, 0.62**
	DPGAN	0.53, 0.60	0.49, 0.60	0.48, 0.59	0.45, 0.58	0.45, 0.58
	DP-CGAN	0.55, 0.63	0.53, 0.61	0.48, 0.57	0.43, 0.57	0.42, 0.55
	No Privacy	0.98, 0.98	0.98, 0.98	0.98, 0.98	0.98, 0.98	0.98, 0.98
Indian Liver	DPRP	**0.79, 0.70**	**0.79, 0.66**	**0.77, 0.66**	**0.75, 0.65**	**0.72, 0.66**
	DPGAN	0.55, 0.59	0.53, 0.60	0.51, 0.59	0.49, 0.57	0.46, 0.54
	DP-CGAN	0.54, 0.60	0.52, 0.59	0.50, 0.54	0.47, 0.52	0.45, 0.51
	No Privacy	0.83, 0.74	0.83, 0.74	0.83, 0.74	0.83, 0.74	0.83, 0.74
Dermatology	DPRP	**0.35, 0.31**	**0.35, 0.31**	**0.32, 0.30**	**0.27, 0.29**	**0.28, 0.29**
	DPGAN	0.20, 0.23	0.19, 0.23	0.19, 0.22	0.19, 0.22	0.19, 0.21
	DP-CGAN	0.22, 0.25	0.21, 0.23	0.21, 0.22	0.20, 0.20	0.20, 0.19
	No Privacy	0.95, 0.99	0.95, 0.99	0.95, 0.99	0.95, 0.99	0.95, 0.99
Cervical Cancer	DPRP	**0.97, 0.93**	**0.97, 0.93**	**0.96, 0.92**	**0.95, 0.91**	**0.95, 0.91**
	DPGAN	0.37, 0.87	0.33, 0.87	0.29, 0.85	0.24, 0.83	0.21, 0.82
	DP-CGAN	0.39, 0.90	0.37, 0.89	0.33, 0.86	0.27, 0.85	0.23, 0.85
	No Privacy	0.98, 0.99	0.98, 0.99	0.98, 0.99	0.98, 0.99	0.98, 0.99
Caesarian	DPRP	**0.56, 0.71**	**0.56, 0.69**	**0.55, 0.66**	**0.55, 0.64**	**0.54, 0.59**
	DPGAN	0.53, 0.51	0.49, 0.51	0.49, 0.51	0.49, 0.50	0.49, 0.49
	DP-CGAN	0.49, 0.48	0.48, 0.48	0.46, 0.47	0.44, 0.47	0.42, 0.45
	No Privacy	0.78, 0.96	0.78, 0.96	0.78, 0.96	0.78, 0.96	0.78, 0.96
HCC	DPRP	**0.76, 0.67**	**0.76, 0.65**	**0.72, 0.62**	**0.66, 0.60**	**0.64, 0.58**
	DPGAN	0.52, 0.57	0.51, 0.53	0.48, 0.51	0.46, 0.51	0.45, 0.49
	DP-CGAN	0.54, 0.59	0.53, 0.56	0.50, 0.54	0.47, 0.52	0.45, 0.49
	No Privacy	0.79, 0.72	0.79, 0.72	0.79, 0.72	0.79, 0.72	0.79, 0.72

Table 2: Main comparison of DPRP with DPGAN and DP-CGAN for varying privacy budgets ($\epsilon \in [8, 6, 4, 2, 1]$). Comparison is made using Area under the Precision-Recall Curve (AUPRC) and Classification Accuracy (Acc). Higher is better for both metrics. Best performing model out of DPRP and GANs is presented in bold. We see that DPRP outperforms GANs on all datasets and for all privacy budgets. We also provide results on the non-private, non-noisy real data as an upper bound achievable (denoted in the table as "No Privacy").

For many scenarios, we observe that DPRP's performance is much closer to the non-noisy, no privacy model, compared to the DPGAN's and DP-CGAN's outcome. Where even with a large privacy budget ($\epsilon = 8$), both GANs struggle to learn the data distribution, which provides further evidence to our earlier claim that state-of-the-art generative models struggle when datasets are small, irrespective of the privacy budget. We also observe that as the privacy budget decreases, there is little variation in GANs performance. This is again due to the reason that GANs struggles to learn the data distribution, even with a larger privacy budget, resulting in similar poor performance, irrespective of the noise scale. For DPRP, for all datasets and for all privacy budgets, the optimal values for k_1 selected are always $> d$, that is our method exploits dimensionality explosion to provide good utility by reducing the noise impact (see Eqn. (6) and Section 3.3 for details).

4.7 IMPACT OF k_2

As we observe from Algorithm 1, one of the main components of our reconstruction, $\hat{V}'$, has it's own parameter, k_2, that can be tuned and has a direct impact on the reconstruction quality. k_2 controls the number of values from the right singular vector used in the reconstruction. Using few ($k_2 < d$) values result in a compressed reconstruction, which can potentially lead to poor outcomes in a differentially private regime (see our discussion on

Data	40% (Acc,AUPRC)	60% (Acc,AUPRC)	80% (Acc,AUPRC)	100% (Acc,AUPRC)
Coimbra BC	0.46, 0.59	0.52, 0.59	0.49, 0.61	0.56, 0.62
Wisconsin BC	0.57, 0.61	0.68, 0.65	0.65, 0.68	0.71, 0.64
Indian Liver	0.57, 0.65	0.77, 0.66	0.75, 0.67	0.61, 0.64
Dermatology	0.32, 0.29	0.32, 0.30	0.33, 0.28	0.29, 0.31
Cervical Cancer	0.95, 0.90	0.96, 0.92	0.95, 0.91	0.97, 0.94
Caesarian	0.49, 0.63	0.55, 0.66	0.59, 0.69	0.63, 0.64
HCC	0.67, 0.62	0.72, 0.62	0.72, 0.62	0.74, 0.65

Table 3: Evaluating the effect of k_2 on the outcome. $\%k_2$ signifies the reduction in the number of values from the right singular vector used for reconstruction relative to the original dataset dimensions d. Privacy budget is kept fixed at $\epsilon = 4, \delta = 0.0001$ with budget allocation kept fixed at 80%/15%.

the Algorithm 1 in Section 3.3, where we explicitly state the relationship between the values used from the right singular vector and the reconstruction quality). As we use the "optimal" values for k_1 in P', we use this section to investigate the impact of k_2 on the reconstruction of the differentially private dataset. For the comparison, we keep the privacy budget constant at $\epsilon = 4, \delta = 0.0001$ with a similar privacy budget split as earlier.

Table 3 shows the results. We observe that as we increase the number of values (k_2) from the right singular vector used in the reconstruction, on average, we see a utility boost. Which is intuitive, as we have explained earlier, $k_2 < d$ leads to a compressed reconstruction, which might not be desirable with differential privacy. However, we advocate for using k_2 as a hyperparameter and choosing the best value for a given dataset as in a few cases (Indian Liver, Dermatology), we observe that using *fewer* ($< 100\%$) noisy singular values provides less distorted reconstruction compared to using all values from the right singular vector (some datasets are more "compressible" than others).

4.8 PRIVACY BUDGET ALLOCATION

For all previous experiments, we have used a constant privacy budget split of $80\%/15\%$, that is 80% of the privacy budget is allocated to making the random projection, P, differentially private while 15% is allocated to the differentially private SVD for getting the right singular vector required for the reconstruction. But it remains to see how the variation in the privacy budget allocation affects the overall model utility. This is what we investigate in this section. For the comparison, we keep the core setup the same as the main results, with the total privacy budget kept constant at $\epsilon = 4, \delta = 0.0001$, and the values for k_2 kept constant at $0.6d$. For space constraints, detailed results are provided in the supplementary material. We observe that the best results are obtained as we increase the privacy budget allocation for the random projection, especially $\geq 40\%$, leading to a less noisy random projection. Signaling that random projection plays a *larger* role in the reconstruction compared to the right singular vector.

4.9 COMPUTATIONAL COMPLEXITY

We briefly mentioned in the introduction that compared to the state-of-the-art generative models, DPRP has an added computational advantage, making the implementation highly "usable" in real-life scenarios for researchers working in resource-constrained environments. Here we briefly justify the claim. Specifically, we compare the average run time of DPRP and GANs over the 50 runs per dataset using percent reduction in computational time as a comparison metric. Due to space constraints, detailed results are provided in the supplementary material. In summary, DPRP offers a reduction in computational time greater than 65% on all datasets, with gains close to 90% on some datasets. The average reduction in computational time across all datasets is close to 80%, a significant decrease compared to GANs.

4.10 COMPARISON WITH "OPTIMAL" GANS

As we have used hyperparameter tuning for DPRP to select the optimal values of k_1, it is only fair that we do the same for DPGAN and DP-CGAN. That is, we use the hyperparameter search to find the best parameters for both GANs to compare with DPRP. Specifically, we use the number of units in the hidden layers, the learning rate, and the number of epochs as the hyperparameters with the total privacy cost adjusted using the method similar to [8]. Even with the use of best hyperparameters, we fail to see any performance benefits for both GANs, hence the detailed results are not displayed (results are worse than reported in Table 2 with DP-CGAN suffering worse deterioration).

There are two main reasons for this phenomenon. First, as we have to charge privacy budget for hyperparameter tuning, doing so can consume *larger* budget for GANs compared to DPRP as in our method we only have a single tuning parameter, compared to the combinatorial explosion in deep learning models. Second, as we have already stated, deep generative models struggle to learn the data generating distribution for small datasets, hence, any setting of hyperparameters fails to provide added benefit.

5 RELATED WORK

The idea of non-private reconstruction is based on the groundwork of Vempala et al. [12], first used by Zhang et al. [13] for image compression. Most of the related work to DPRP can be divided into two main categories, that is the differentially private generative models and differential privacy in random projections.

5.1 GANS FOR DIFFERENTIAL PRIVATE DATA RELEASE

Following limited successes of prior privacy-preserving data sharing methods [11, 17], that are either limited by the computational complexity, constrained output space (such as [11] only works with binary outcomes), etc. GANs [4] have established themselves as the current state-of-the-art in differentially private synthetic data release [6, 7]. With the exception of PATE-GAN [7], all other differentially private GAN based models are trained using differentially private stochastic gradient decent [8]. Both approaches, however, fall severely short on the utility front when dealing with small datasets, as we discussed at length in Section 1.3.

5.2 RANDOM PROJECTIONS AND DIFFERENTIAL PRIVACY

Releasing differentially private random projections is studied in [20, 23]. For both, in addition to not releasing the data from the *original* data distribution, albeit only a random projection, limiting the inferential utility and prohibiting domain experts to do a *qualitative* assessment by comparing the reconstructed data to the original data per-observation, former only provides privacy at the *attribute-level*, not the often desired *user-level*. Subsequent works using differentially private random projections follow a similar approach and hence suffer from similar drawbacks, such as Ahmed et al. [24] uses random projections to release social network graph data, and the work by Xu et al. [25] uses differentially private random projection for releasing a low dimensional projection of original high dimensional data. As mentioned, these works are limited to providing privacy at the attribute level and releasing *just* random projections of the original data, severely limiting the inferential utility.

6 CONCLUSION, LIMITATIONS, AND FUTURE WORK

We have presented DPRP, a new method to release differentially private reconstruction of small-sized datasets. Based on random projections and right singular vectors, DPRP constitutes a model-free approach, which is easy to implement and computationally cheap, while providing strong privacy and utility guarantees. With the aid of our extensive empirical evaluation on seven real-life datasets, we have shown that DPRP outperforms state-of-the-art generative models for all privacy budgets and for all datasets. One of the main limitations of DPRP in comparison to generative models is that DPRP is limited to reconstruction of same sized datasets as input, whereas generative models can generate variable-sized output.

For our future work, we would like to push the limits of DPRP to extend it to large and high dimensional datasets. Some other research directions worth exploring are the use of efficient decompositions to get the right singular vectors, whereby further reducing the required noise for differential privacy; using an ensemble of differentially private random projections for reconstruction, where we can reduce the noise by averaging multiple “noisier” random projections; investigating the impact of the input rank, and evaluating the use of the matrix-based noise adding mechanisms.

ACKNOWLEDGEMENTS

We thank Ricardo Silva Carvalho for valuable feedback. This research is in part supported by a CGS-D award and a discovery grant from Natural Sciences and Engineering Research Council of Canada.

References

[1] B. Lo, L. Dornbrand, and N. N. Dubler, “Hipaa and patient care: the role for professional judgment,” *Jama*, vol. 293, no. 14, pp. 1766–1771, 2005.

[2] L. Sweeney, “Foundations of privacy protection from a computer science perspective,” in *Proceedings of the Joint Statistical Meeting, AAAS*, 2000.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proceedings of the Third Conference on Theory of Cryptography*, TCC’06, (Berlin, Heidelberg), pp. 265–284, Springer-Verlag, 2006.

[4] 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*, pp. 2672–2680, 2014.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 3–18, IEEE, 2017.

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

[7] J. Yoon, J. Jordon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2019.

[8] 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 SIGSAC Conference on Computer and Communications Security*, pp. 308–318, ACM, 2016.

[9] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "Near-optimal algorithms for differentially-private principal components," *arXiv preprint arXiv:1207.2812*, 2012.

[10] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett, "Privgene: differentially private model fitting using genetic algorithms," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 665–676, ACM, 2013.

[11] 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.

[12] S. S. Vempala, *The random projection method.* American Mathematical Soc., 2005.

[13] Q. Zhang and R. J. Plemmons, "Image reconstruction from double random projection," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2501–2513, 2014.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, Aug. 2014.

[15] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[16] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.

[17] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: optimal bounds for privacy-preserving principal component analysis," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 11–20, ACM, 2014.

[18] S. Tu, *Differentially private random projections*, Accessed Nov 2019. https://people.eecs.berkeley.edu/~sltu/writeups/dp-rp.pdf.

[19] B. Laurent and P. Massart, "Adaptive estimation of a quadratic functional by model selection," *Annals of Statistics*, pp. 1302–1338, 2000.

[20] K. Kenthapadi, A. Korolova, I. Mironov, and N. Mishra, "Privacy via the johnson-lindenstrauss transform," *Journal of Privacy and Confidentiality*, vol. 5, no. 1, pp. 39–71, 2013.

[21] R. Torkzadehmahani, P. Kairouz, and B. Paten, "Dp-cgan: Differentially private synthetic data and label generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, pp. 214–223, 2017.

[23] J. Blocki, A. Blum, A. Datta, and O. Sheffet, "The johnson-lindenstrauss transform itself preserves differential privacy," in *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 410–419, IEEE, 2012.

[24] F. Ahmed, R. Jin, and A. X. Liu, "A random matrix approach to differential privacy and structure preserved social network graph publishing," *arXiv preprint arXiv:1307.0475*, 2013.

[25] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "Dp-pro: Differentially private high-dimensional data release via random projection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.