Leveraging Vertical Public-Private Split for Improved Synthetic Data Generation

Samuel Maddock, Shripad Gade, Graham Cormode & Will Bullock
Meta Platforms, Inc.
{smaddock,shripadgade,gcormode,bullock}@meta.com

Abstract

Differentially Private Synthetic Data Generation (DP-SDG) is a key enabler of private and secure tabular-data sharing, producing artificial data that carries through the underlying statistical properties of the input data. This typically involves adding carefully calibrated statistical noise to guarantee individual privacy, at the cost of synthetic data quality. Recent literature has explored scenarios where a small amount of public data is used to help enhance the quality of synthetic data. These methods study a horizontal public-private partitioning which assumes access to a small number of public rows that can be used for model initialization, providing a small utility gain. However, realistic datasets often naturally consist of public and private attributes, making a *vertical public-private partitioning* relevant for practical synthetic data deployments. We propose a novel framework that adapts horizontal public-assisted methods into the vertical setting. We compare this framework against our alternative approach that uses conditional generation, highlighting initial limitations of public-data assisted methods and proposing future research directions to address these challenges.

1 Introduction

Due to increasing demand for privacy-preserving data sharing, differentially private synthetic data generation (DP-SDG) has emerged as a powerful tool for sharing sensitive tabular data. By generating "artificial" data that retains the statistical properties of the real data and adhering to formal privacy constraints, DP-SDG allows organizations to share, analyze and train models on data without exposing sensitive information (Assefa et al., 2020; van Breugel & van der Schaar, 2023). Differential Privacy (DP) outlines a formal framework for individual privacy, providing SDG methods with quantifiable privacy guarantees, often at the expense of synthetic data quality (Dwork et al., 2006; Hardt et al., 2012). State-of-the art (SOTA) methods for private tabular synthetic data use DP to offer strong privacy guarantees. These methods typically involve taking low-dimensional measurements of the dataset, such as marginal queries, and adding calibrated statistical noise to ensure DP. A model is trained over these noisy measurements to learn a synthetic data representation that can be sampled from, such as a graphical model (Zhang et al., 2017; McKenna et al., 2019) or generator neural networks (Liu et al., 2021b).

Many real-world datasets naturally contain a mix of public and private information which can be leveraged to improve synthetic data quality (utility). In the context of differentially private learning, the concept of public-data assisted algorithms is well-studied (Bie et al., 2022; Ganesh et al., 2023; Ullah et al., 2024). Most existing DP-SDG algorithms treat the input data as entirely private (Zhang et al., 2017; 2021; McKenna et al., 2022). However, real-world tabular datasets often contains a natural partition of public and private information. For example, demographic statistics like age may be publicly available from census information, while sensitive attributes related to financial or medical data require strict privacy protections. In many industry applications, companies that collect sensitive information have subsets of data that can be freely used, e.g., information users have explicitly consented to be used for improving products which can be used by the SDG algorithm.

Recent research has introduced methods for synthetic data generation in public-data assisted settings, primarily focusing on a *horizontal* partitioning (Liu et al., 2021a;b). In this setting, a subset of rows over the entire dataset is considered public while the remainder is private. These methods adapt existing marginal-based DP-SDG algorithms to make use of the public rows in order to improve synthetic data quality (Wang et al., 2023). In several cases, the essence of these methods is to simply initialize the synthetic data model with public data, then proceed with training on the private dataset under DP. Leveraging the public subset in this way has been shown to enhance the utility of DP synthetic data (Liu et al., 2021b). While this approach can improve synthetic data quality, it ignores the public data after initialization (risking forgetting), and so there is potential for more closely integrating public data into the private algorithm. Furthermore, it is often unrealistic to assume that the public subset is identical in distribution to the private dataset: in many scenarios the public subset will be small and from a biased population, so may result in a poor model initialization. Meanwhile, if it were the case that the public subset does represent the private distribution accurately, then there is little benefit to be gained from fine tuning on the private data.

In this work, we study an equally important but unexplored problem of a *vertical* public-private partitioning. Here, some subset of columns in the dataset are considered public, while the rest are treated as private. This vertical partitioning is common in practice, as organizations frequently handle datasets with well-defined public and private features. In addition, these applications often involve scenarios where the number of private columns are much smaller than the number of public columns, necessitating the use of public-data assisted DP-SDG in order to obtain high utility. Despite its practical relevance, the vertical setting remains under-explored in the literature. Our contributions are as follows:

- **Vertical Public-Private Setting**: We propose a framework for the adaptation of horizontal public-assisted DP-SDG methods to the vertical setting. We find our methods achieve good utility whilst existing baselines often fail to improve over fully private methods, even in settings where the percentage of public columns is large.
- **Conditional Generation:** We propose an alternative approach based on conditional generation, adapting marginal-based methods that use Private-PGM. We find this approach greatly improves synthetic data quality in a vertical setting and achieves best utility overall.
- **Re-thinking public-private SDG:** We enumerate the current limitations and propose future directions to help advance public-assisted private synthetic data generation in the vertical setting.

2 Related Work

A key class of methods for generating private synthetic data are marginal-based methods (Liu et al., 2021b). These methods primarily focus on learning a synthetic data distribution by measuring low-dimensional statistics, such as marginal queries, under DP noise. Examples include PrivBayes (Zhang et al., 2017), which learns a Bayesian network from noised mutual information, and Private-PGM, which trains a graphical model over noisy marginals (McKenna et al., 2019). Approaches using neural networks have gained popularity with methods like DP-CTGAN (Fang et al., 2022), PATE-GAN (Jordon et al., 2018), and DP-VAE (Weggenmann et al., 2022) which employ deep learning techniques to directly model synthetic data and are trained privately via DP-SGD (Abadi et al., 2016). Our focus in this work is on private marginal-based methods, as they are shown to outperform their neural network counterparts on tabular data (Liu et al., 2022; Ganev et al., 2023).

Recent advances in marginal-based methods follow the “select-measure-generate” paradigm (Liu et al., 2021b). The goal is to generate synthetic data that preserves the answers to a given workload of queries. The approach is iterative and involves a number of steps. In each iteration, the query that is worst-approximated under the current synthetic model is selected under DP. This marginal is measured and perturbed with DP noise, and is used to update the synthetic model. The MWEM (Hardt et al., 2012) algorithm was the first to adopt such an approach, but its synthetic data representation directly models the entire joint distribution leading to poor scalability as the number of columns increase. Alternative methods like GEM (Liu et al., 2021b) replace the inefficient representation in MWEM with a generator neural network that is trained directly on noisy measurements. The current SOTA is AIM (McKenna et al., 2022), which combines Private-PGM (McKenna et al., 2019), a graphical model inference procedure on noisy measurements, with a more sophisticated selection algorithm in order to achieve high utility and scale to a larger number of columns.

The integration of public data for improving the utility in DP-SDG algorithms has been explored under a horizontal partitioning. In such settings, a small subset of rows is assumed to be public. These methods either pretrain the synthetic data model on public data before using the (fully) private algorithm or augment the training process of the private SDG algorithm with statistics from the public subset. Liu et al. (2021a) propose PMW, a public-data assisted version of MWEM. They initialize both the synthetic data distribution and its support using public data, then proceed as normal with MWEM. They show the use of public data can help bridge the gap between the fully private setting and perfectly fitting the training data. However, their focus is restricted to a horizontal partitioning and uses MWEM, which is a method that suffers on high-dimensional data. Liu et al. (2021b) propose a public-data assisted version of GEM. The idea is analogous to PMW: the generator network in GEM^{Pub} is pretrained on all marginals in the public dataset, then the GEM^{Pub} algorithm proceeds as normal on the private dataset, using the pretrained model as initialization. This brings both added utility and scalability over PMW. Wang et al. (2023) propose a post-processing scheme that can be applied to the output of any SDG to help improve utility, including the public-data assisted setting. Finally, Fuentes et al. (2024) propose JAM-PGM, a public-data assisted version of the AIM algorithm. The selection step of AIM is extended to decide at each iteration whether to measure a private marginal (with DP noise) or use a public (noise-free) marginal. This more closely involves the public data in the training process and helps achieve better utility than GEM^{Pub} in practice.

The only other work we are aware of that addresses a vertical partitioning for public-assisted DP-SDG is due to Liu et al. (2021b), who outline extensions to GEM^{Pub} that allow a vertical partitioning. In essence, since GEM^{Pub} trains a generator network over marginals, it can be pretrained on any set of public measurements, computed from either a horizontal or vertical partitioning.

3 Methods

We assume access to a dataset D over n individuals, vertically partitioned into a private dataset $D_{\text{priv}} \in \mathbb{N}^{n \times d_{\text{priv}}}$ and a public dataset $D_{\text{pub}} \in \mathbb{N}^{n \times d_{\text{pub}}}$ with $d := d_{\text{priv}} + d_{\text{pub}}$ being the total number of columns in D[1]. We are mainly interested in the setting where $d_{\text{priv}} < d_{\text{pub}}$. The goal is to generate a synthetic dataset $\hat{D} \in \mathbb{N}^{n \times d}$ under differential privacy. In this work we are concerned with guaranteeing (ε, δ)-DP.

Definition 3.1 (**(ε, δ)-DP**) *A randomized algorithm $\mathcal{M}$ is (ε, δ)-DP if for any neighboring datasets D, D' and any subset of outputs S we have $\mathbb{P}(M(D) \in S) \leq e^{\varepsilon}\mathbb{P}(M(D') \in S) + \delta$*

The parameter ε is the privacy budget and determines how strong the privacy guarantee is. Using a large ε will decrease the noise in "Select-Measure-Generate" algorithms but in turn reduces the formal privacy guarantees. The parameter δ determines a probability that the DP guarantee fails and is usually set to be cryptographically small. Alternative definitions seeks to improve composition results such as zero-Concentrated Differential Privacy (zCDP) (Bun & Steinke, 2016). All methods in our experiments use zCDP. See Appendix A for full details. In the vertical public-assisted setting we consider in this work, the DP guarantee of $\mathcal{M}$ is applied to D_{priv} and we assume $\mathcal{M}$ has access to D_{pub} for no additional privacy cost (as it is public information).

The general vertical public-assisted DP-SDG framework is summarised in Algorithm 1. Here we adapt the adaptive measurements framework proposed by Liu et al. (2021b) to our vertical public-assisted setting. There are two key areas that differ. Firstly, model initialization may depend on D_{pub} i.e., for public pretraining. Secondly, the measurement step may also depend on D_{pub} e.g., for measuring a public marginal which requires no DP noise.

3.1 Adapting Pretraining Methods to the Vertical Setting

Existing DP-SDG methods can be adapted to the vertical public-assisted setting by fitting them into the framework outlined in Algorithm 1.

vGEM^{Pub} (Liu et al., 2021b): The GEM method uses a generator neural network to model the synthetic data distribution. No pre-processing of the workload is used, so $W^* = W$. The model

[1] We assume the dataset is discrete as in prior work. In practice, numerical features can be binned.

Algorithm 1 Vertical Public-assisted Adaptive Measurements (vPAM)

```
Input: Private dataset $D_{\text{priv}}$, public dataset $D_{\text{pub}}$, workload of queries $W$, training steps $T$, privacy parameters $(\varepsilon, \delta)$
Output: Synthetic data $\hat{D}$
Pre-process the workload $W^* :=$ PROCESS-WORKLOAD($W$)
$\theta_0 :=$ MODEL-INIT($D_{\text{pub}}$), $\hat{D}_0 \sim \theta_0$
for $t = 0, \dots, T-1$ do
    Select: via the Exponential mechanism $q_{t+1} \in W^*$ using SCORE($q; D_{\text{priv}}, D_{\text{pub}}, \hat{D}_t$)
    Measure: selected marginal query $q_{t+1}$ i.e., $\tilde{M}_{t+1} :=$ MEASURE($q_{t+1}; D_{\text{priv}}, D_{\text{pub}}, \sigma^2$)
    Update: synthetic model $\theta_{t+1} :=$ TRAIN-MODEL($\theta_t, \{\tilde{M}_1, \dots, \tilde{M}_{t+1}\}$)
    Generate: $\hat{D}_{t+1} \sim \theta_{t+1}$
end for
Output $\hat{D} \sim f(\{\theta_t\}_{t=1}^{T})$
```

initialization in the vertical setting pre-trains the generator neural network on all 3-way marginals in W that contain only public features in D_{pub}. Next, the error scores for query selection simply measure the error between the current synthetic data model and the private dataset at step t, with $\text{SCORE}(q; D, \hat{D}_t) := \|M_{q_t}(D) - M_{q_t}(\hat{D}_t)\|$. TRAIN-MODEL$(\cdot)$ performs a number of SGD steps on the current GAN θ_t to produce θ_{t+1}. The gradients for these updates are computed from the average L_1 loss between the current marginals produced by θ_t and the observed noisy measurements $\{\tilde{M}_1, \dots, \tilde{M}_t\}$. Finally, the post-processing function $f(\cdot)$ performs an Exponential Moving Average (EMA) over the last $T/2$ generator networks, and the final synthetic dataset $\hat{D}$ is sampled from this.

vPMW: PMW can be adapted to the vertical setting in a similar way to vGEM$^{\text{Pub}}$. The pretraining process for MODEL-INIT is the same as in vGEM$^{\text{Pub}}$, where the initial model θ_0 is initialized over all 3-way marginals in W^* that only contain columns in D_{pub}. All other steps remain the same as vGEM$^{\text{Pub}}$ except for TRAIN-MODEL$(\cdot)$, which replaces the generator neural network via direct modelling using multiplicative weights i.e., $\theta_{t+1} := \theta_t \cdot \exp(q_{t+1}(x) \cdot (\tilde{M}_{t+1} - q_{t+1}(\theta_t))/2n)$. Finally, the post-processing function averages all T synthetic distributions, $f(\cdot) := \frac{1}{T}\sum_t \theta_t$.

3.2 Adapting JAM-PGM to the Vertical setting

JAM-PGM (Fuentes et al., 2024) : The original algorithm extends AIM to the horizontal public-assisted setting. Firstly, JAM-PGM initializes a workload W^* that has two additional properties: it contains the downward closure of W (i.e., all lower order marginals that can be formed from queries in W are also added to W^*) and all public-marginals that can be measured from D_{pub} are added to W^* separately from those that can also be measured on D. The scoring functions are adapted from PMW to take into account the predicted error of selecting a public or private marginal. More specifically, the score for a private marginal is $\text{SCORE}(q; D, \hat{D}_t) := \|M_{q_t}(D) - M_{q_t}(\hat{D}_t)\| - \sqrt{2/\pi}\sigma n_{q_t}$ which adapts the PMW score to include an expected error term based on measuring q_t under Gaussian noise. For the case of a public marginal, the score is adapted to $\|M_{q_t}(D) - M_{q_t}(\hat{D}_t)\| - \frac{|D|}{|D_{\text{pub}}|}\|M_{q_t}(D) - M_{q_t}(D_{\text{pub}})\|$ which replaces the expected error of Gaussian noise with the expected error of measuring from the smaller public dataset. The measurement step is changed to adapt to W^*, where if q_t is public, then the marginal is measured without noise on D_{pub} i.e., $\tilde{M}_t := q_t(D_{\text{pub}})$ otherwise the measurement happens under Gaussian noise as normal, $\tilde{M}_t := q_t(D) + N(0, \sigma^2)$. The initialization step for JAM-PGM uses a random graphical model and does no public pretraining. Finally, the post-processing function $f(\cdot)$ returns the last model θ_T, since a new graphical model is estimated at each step with (potentially) different structures.

vJAM-PGM: When adapting JAM-PGM to the vertical setting, the augmented workload W^* still includes both private and public marginals, except now the same marginal is only added to the workload once as it is either deemed private (contains at least one private column) or public (contained within D_{pub}). This necessitates changing the score functions, since now the expected error of measuring a public marginal is zero, as it is exact. When measuring a public marginal, we use the original MWEM scoring i.e., $\text{SCORE}(q_{\text{public}}; D_{\text{pub}}, \hat{D}_t) := \|M_{q_t}(D_{\text{pub}}) - M_{q_t}(\hat{D}_t)\|$. Furthermore, we initialize the graphical model on all 1-way marginals as in the original AIM algorithm (McKenna et al., 2022), except the public columns from D_{pub} are used as-is without any Gaussian noise.

3.3 Conditional Generation

In the vertical setting we have access to the public columns (D_{pub}) of the underlying dataset. We can naturally use this to improve the synthetic data sampling process. We consider conditional generation for marginal-based methods that utilize the Private-PGM algorithm (McKenna et al., 2019), whose synthetic data representation is a graphical model. In this process, the graphical model forms a factorisation of the joint probability distribution from which, given an elimination order, we can sample synthetic data. To adapt this to a vertical partitioning, during the generation/sampling phase we simply use the raw data for public columns (i.e., exact marginals) and only sample private columns via the graphical model.

4 Experiments

Methods. We compare (fully) private AIM (McKenna et al., 2022) against the closest baseline GEM^{Pub} (Liu et al., 2021b) which we denote vGEM^{Pub} and our two vertical approaches: vPMW, vJAM-PGM. We train these methods on a workload of all 3-way marginals. For methods using Private-PGM (AIM, vJAM-PGM) we apply our conditional generation approach. For conditional methods, since the public columns are generated as is, we remove them from the training workload and instead use a workload of 3-way marginals that contain at least one private column.

Datasets. We focus on two public datasets. The first is Adult (Kohavi & Becker, 1996), used in prior horizontal public-private work (Liu et al., 2021b; Fuentes et al., 2024) and also by GEM^{Pub} in a vertical setting (Liu et al., 2021b). As PMW cannot scale to more than a few columns, we also consider Adult (red.), a smaller version containing only the first 8 columns. The second dataset we use is the Census-Income KDD dataset (cen, 2000) which we use as a proxy for a more practical large-scale dataset allowing us to evaluate our methods across a larger range of public-private splits. We filter out rows with missing values to obtain 95,130 rows and 40 columns.

Evaluation. We repeat experiments 3 times and present the average L1 error over all k-way marginals with $k \leq 3$. See Appendix B.1 for full experiment hyperparameters and open-source implementations used.

4.1 Results

Comparison of methods on Adult (reduced). We start by comparing all methods on Adult (red.), taking 75% of the columns to be public whilst varying the privacy budget ε, shown in Figure 1. We observe, like in prior horizontal work, that PMW has generally poor utility across all values of ε. We also find vGEM^{Pub} struggles to achieve good utility, performing worse than (fully) private AIM across all settings. For methods based on Private-PGM, we see that when $\varepsilon < 1$, vJAM-PGM achieves consistently lower error than AIM but that as ε grows large this gap starts to decrease, since the benefit of using public columns diminishes. We omit vPMW from further experiments, due to poor scalability and utility.

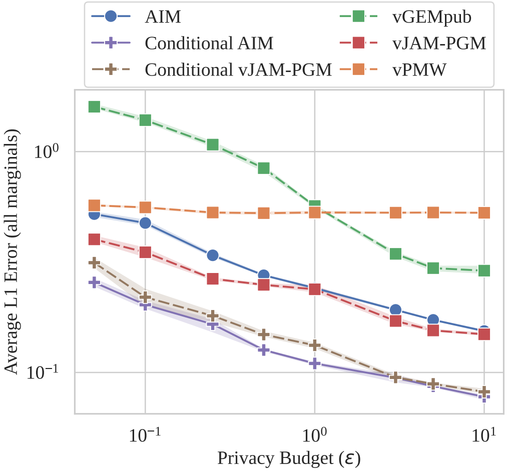

Figure 1: Varying ε on Adult (red.). with $d_{\text{pub}} = 6$

Varying ε. In Figure 2, we vary ε on the full Adult dataset with 15 columns, taking [25%, 50%, 75%] of the columns to be public. When 25% of the columns are public, none of the vertical public-assisted methods provide better utility than using (fully) private AIM except when $\varepsilon < 0.1$

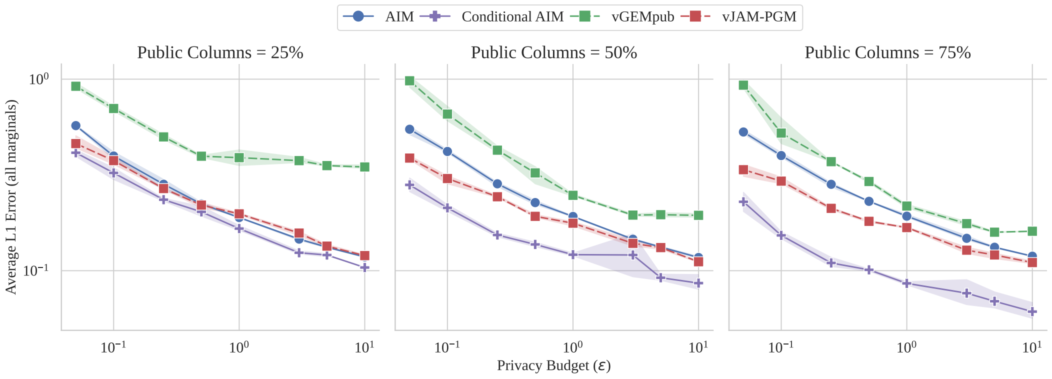

Figure 2: Varying ε on Adult with $[25\%, 50\%, 75\%]$ of the columns being public.

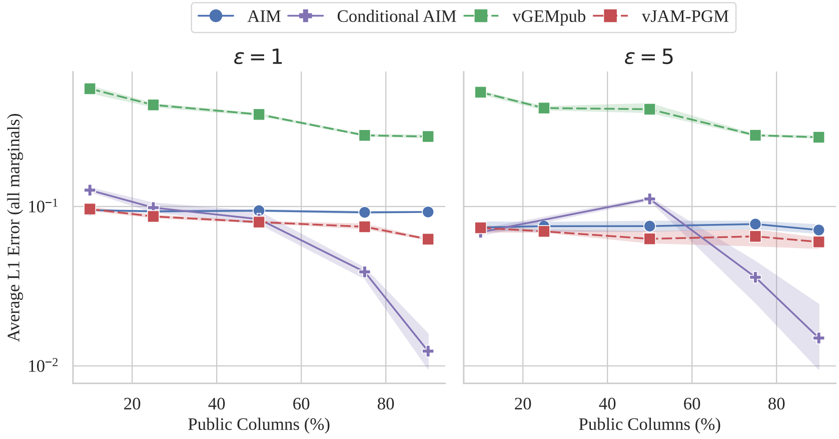

Figure 3: Varying the percentage of public columns on Census data with $p \in \{10\%, 25\%, 50\%, 75\%, 90\%\}$ and $\varepsilon = 1, 5$

for vJAM-PGM. This changes as the number of public columns increases, and at 50% with $\varepsilon < 1$ vJAM-PGM achieves lower error than AIM. For 75% public, vJAM-PGM achieves lower error than AIM across all values of ε. We note vGEM$^{\text{Pub}}$ has consistently worse error than (fully private) AIM across all settings.

Conditional Generation. In Figure 1 we plot conditional variations of AIM and vJAM-PGM. We find both of these methods achieve superior utility, outperforming all other vertical methods. We omit Conditional vJAM-PGM from further experiments, since it performs consistently worse than Conditional AIM. Figure 2 and 3, further shows Conditional AIM achieving lowest error across all settings, even beating AIM in scenarios where the number of public columns is small. The advantage of conditional generation not only helps achieve zero error across all public-marginals but consistently lowers private error as well.

Varying the public-private split (p). In Figure 3, we consider a high-privacy ($\varepsilon = 1$) and low-privacy ($\varepsilon = 5$) setting on the Census dataset where we vary the percentage of public columns. We find, as on Adult, that vertical methods only achieve lower utility than (fully) private AIM when the number of public columns is $> 50\%$. We continue to observe the benefit of conditional generation, where the gap against vJAM-PGM is most striking when $p \geq 75\%$.

5 Discussion

Utility. A consistent finding from our experiments is that vertical public-assisted methods struggle to remain competitive against (fully) private AIM. That is to say, the additional pretraining on public data currently has little effect on the overall utility. The exception is vJAM-PGM, which integrates public data more closely into the training of the model, but only beats AIM when either ε is small or d_{pub} is large. The only approach with consistently low error is Conditional AIM.

Scalability. Although Conditional AIM achieves best utility, we have found scenarios where conditional sampling can require an intractable amount of memory when conditioning on a large number of public columns. We have found that heuristic approaches based on restricting the size of public marginals used in conditional sampling can alleviate these issues with no significant changes to utility. We note these problems are inherent to methods based on Private-PGM which struggle to scale to a large number of columns in both the fitting and generation step. This is prohibitive in industry applications where high-dimensional data is ubiquitous and is an area where we hope to make improvements in the future.

Future Directions. We have found that adaptations of existing methods do not always provide sufficient benefit over (fully) private AIM, even when there are more public than private columns. Instead, we observe that moving away from pretraining approaches and towards conditional generation can help achieve the best synthetic data quality in the vertical setting. We believe there are many future directions for improving public-assisted methods. One such focus is enhancing both the scalability and utility of conditional methods, by improving the elimination order for Private-PGM based methods or adapting the generator networks in GEM to allow for conditional generation. Another direction is to focus the design of future vertical public-assisted algorithms away from pretraining and to instead make more use of public data during training such as in vJAM-PGM.

References

Census-Income (KDD). UCI Machine Learning Repository, 2000. DOI: https://doi.org/10.24432/C5N30T.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC CCS*, pp. 308–318, 2016.

Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1–8. ACM, 2020.

Alex Bie, Gautam Kamath, and Vikrant Singhal. Private estimation with public data. *Advances in neural information processing systems*, 35:18653–18666, 2022.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658, Berlin, 2016. Springer, Springer.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Mei Ling Fang, Devendra Singh Dhami, and Kristian Kersting. DP-CTGAN: Differentially private medical data generation using CTGANs. In *International Conference on Artificial Intelligence in Medicine*, pp. 178–188. Springer, 2022.

Miguel Fuentes, Brett C Mullins, Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Joint selection: Adaptively incorporating public information for private synthetic data. In *International Conference on Artificial Intelligence and Statistics*, pp. 2404–2412. PMLR, 2024.

Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Guha Thakurta, and Lun Wang. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, pp. 10611–10627. PMLR, 2023.

Georgi Ganev, Kai Xu, and Emiliano De Cristofaro. Understanding how differentially private generative models spend their privacy budget, 2023.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 34:11631–11642, 2021.

Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25, 2012.

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

Ronny Kohavi and Barry Becker. Adult dataset. UCI machine learning repository, 1996. URL `http://archive.ics.uci.edu/ml/nomao`.

Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *International Conference on Machine Learning*, pp. 6968–6977. PMLR, 2021a.

Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34: 690–702, 2021b.

Yucong Liu, Chi-Hua Wang, and Guang Cheng. On the utility recovery incapability of neural net-based differential private tabular training data synthesizer under privacy deregulation, 2022.

Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *ICML*, pp. 4435–4444. PMLR, 2019.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Enayat Ullah, Michael Menart, Raef Bassily, Cristóbal Guzmán, and Raman Arora. Public-data assisted private stochastic optimization: Power and limitations. *arXiv preprint arXiv:2403.03856*, 2024.

Boris van Breugel and Mihaela van der Schaar. Beyond privacy: Navigating the opportunities and challenges of synthetic data. *arXiv preprint arXiv:2304.03722*, 2023.

Hao Wang, Shivchander Sudalairaj, John Henning, Kristjan Greenewald, and Akash Srivastava. Post-processing private synthetic data for improving utility on selected measures. *Advances in Neural Information Processing Systems*, 36:64139–64154, 2023.

Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. DP-VAE: Human-readable text anonymization for online reviews with differentially private variational autoencoders. In *Proceedings of the ACM Web Conference*, pp. 721–731, 2022.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. PrivSyn: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 929–946, 2021.

A Differential Privacy

In this work we are concerned with synthetic data methods that guarantee (ε, δ)-DP.

Definition A.1 (**(ε, δ)-DP**) *A randomized algorithm $\mathcal{M}$ is (ε, δ)-differentially private if for any neighboring datasets D, D' and any subset of outputs S we have $\mathbb{P}(M(D) \in S) \leq e^{\varepsilon}\mathbb{P}(M(D') \in S) + \delta$*

We consider neighboring datasets to mean that D' is formed from the addition or removal of a single data-point in D. One useful property of DP is composition, that is the output of two DP algorithms on D with privacy parameters $\varepsilon_1, \varepsilon_2$ is $(\varepsilon_1 + \varepsilon_2)$-DP. Alternative DP definitions seek to improve these composition results. One such approach is zero-Concentrated Differential Privacy (zCDP)(Bun & Steinke, 2016).

Definition A.2 (**ρ-zCDP**) *An algorithm $\mathcal{M}$ is ρ-zCDP if for any two neighbouring datasets D, D' and all $\alpha \in (1, \infty)$ we have $D_\alpha(\mathcal{M}(D)|\mathcal{M}(D') \leq \rho \cdot \alpha$, where D_α is Renyi divergence of order α.*

We can convert an associated ρ-zCDP guarantee into (ε, δ)-DP via the following lemma.

Lemma A.3 (zCDP to DP (Canonne et al., 2020)) *If an algorithm $\mathcal{M}$ satisfies ρ-zCDP then it satisfies (ε, δ)-DP for all $\varepsilon > 0$ with*

$$\delta = \min_{\alpha > 1} \frac{\exp((\alpha - 1)(\alpha\rho - \varepsilon))}{\alpha - 1} \left(1 - \frac{1}{\alpha}\right)^{\alpha}$$

Existing methods like AIM(McKenna et al., 2022) utilize zCDP accounting. We implement all vertical "Select-Measure-Generate" methods in our framework with zCDP accounting, composing the Exponential ("Select" step) and Gaussian ("Measure" step) mechanisms over T rounds. We note the privacy accounting of vJAM-PGM following that of JAM-PGM spends some extra privacy budget to decide whether to pick a public or private marginal at a given round under the framework of the Exponential mechanism. We refer to the original work for full privacy details (Fuentes et al., 2024).

Definition A.4 (Gaussian Mechanism) *Let $q : \mathcal{D} \to \mathbb{R}^d$ be a sensitivity 1 query, the Gaussian mechanism releases $q(D) + \Delta_2(f) \cdot \mathcal{N}(0, \sigma^2 I_d)$ and satisfies $\frac{1}{2\sigma^2}$-zCDP.*

Definition A.5 (Exponential Mechanism) *Let $s(c) : \mathcal{D} \to \mathbb{R}$ be a score function defined over a set of candidates C. The exponential mechanism releases c with probability $\mathbb{P}[\mathcal{M}(D) = c] \propto \exp(\frac{\varepsilon}{2\Delta} \cdot s(c))$, with $\Delta := \max_q \Delta_1(s(c))$. This satisfies $\frac{\varepsilon^2}{8}$-zCDP.*

Finally, we note that further accounting improvements can be made to improve composition when then number of rounds T is large, using numerical accounting methods as in DP-ML (Mironov, 2017; Gopi et al., 2021) but is not a setting we consider.

B Experiment Details

B.1 Implementation Details

In our experiments we use the following methods:

- **vPMW and v$\text{GEM}^{\textbf{Pub}}$:** We use the open-source implementations of PMW and GEM^{Pub} by the original authors[2]. We modify these methods to change the pretraining to only pretrain on marginals that contain public columns as discussed in Section 3.1.
- **vJAM-PGM:** We use the open-source implementation of JAM-PGM[3] by Fuentes et al. (2024) with the modifications discussed in Section 3.2.

[2] https://github.com/terranceliu/iterative-dp

[3] https://github.com/Miguel-Fuentes/JAM_AiStats

- **AIM / Conditional AIM:** For AIM, we use the original implementation by the original authors[4]. For conditional AIM, we modify the Private-PGM algorithm to conditionally sample based on the public vertical columns as discussed in Section 3.3.

B.2 Datasets and Hyperparameters

We use the following datasets:

- **Adult** (Kohavi & Becker, 1996): We use the Adult dataset with $32,561$ rows and 15 features. For Figure 1, we use Adult (red.), a reduced version we form by using only the first 8 columns.
- **Census-Income KDD** (cen, 2000): We preprocess it by dropping all rows with missing values and any columns that are constant values. This gives a final dataset with $95,130$ rows and 40 columns. We discretize numerical columns via (non-private) quantiles with a maximum of 50 distinct values. We note that this process can be done privately but is not the focus of our work.

Our experiments in Section 4 use the following hyperparameters:

- **Privacy parameters** (ε, δ): Whilst we vary ε in different experiments the privacy accounting is kept constant among methods in line with the discussion in Appendix A. We fix $\delta = 1e - 6$ in all experiments.
- **Number of Rounds** (T): We fix this to $T = 100$ in all experiments. We varied T and did not see any significant change to results.
- **Pretraining Rounds**: For vGEM$^{\text{Pub}}$, we pretrain on all public marginals for 10 rounds. For vPMW the synthetic data is initialized directly from the public marginals.

For other method specific hyperparameters (e.g., GEM learning rates or the number of iterations to optimize the graphical model in AIM) we use the same parameter defaults as the open-source implementations noted in Appendix B.1. This applies to vGEM$^{\text{Pub}}$, vPMW, AIM (also Conditional AIM) and vJAM-PGM.

[4] `https://github.com/ryan112358/private-pgm`