Benchmarking Differentially Private Synthetic Data Generation Algorithms*

Yuchao Tao, [1,3] **Ryan McKenna,** [1,2]
Michael Hay, [1,4] **Ashwin Machanavajjhala,** [1,3] **Gerome Miklau** [1,2]
[1] Tumult Labs, USA, [2] University of Massachusetts Amherst, USA
[3] Duke University, USA, [4] Colgate University, USA
science@tmlt.io

Abstract

This work presents a systematic benchmark of differentially private synthetic data generation algorithms that can generate tabular data. Utility of the synthetic data is evaluated by measuring whether the synthetic data preserve the distribution of individual and pairs of attributes, pairwise correlation as well as on the accuracy of an ML classification model. In a comprehensive empirical evaluation we identify the top performing algorithms and those that consistently fail to beat baseline approaches.

Introduction

While there are many compelling reasons to share data about individuals, such sharing is often prevented due to privacy concerns. Differentially private synthetic data generation stands out as an appealing solution to this problem: it provides strong formal privacy guarantees, while producing a synthetic data set that "looks like" the real data from the perspective of an analyst. This problem has received considerable attention from the research community, with a wide variety of approaches available in the literature. [27, 26, 38, 40, 36, 29, 18, 29, 15, 35, 5, 10, 2, 20, 19, 21, 41, 22, 23, 32, 16, 33, 11].

Despite the variety of mechanisms available for this task, the community is lacking a systematic empirical study that compares a variety of mechanisms on different datasets, tasks, and privacy levels. Prior work in this space includes [29], which focuses on GAN-based algorithms in terms of the machine learning classification accuracy; [13], which focuses on GAN-based algorithms in terms of the utility of classification, clustering, aggregation queries and privacy protection; [9], which focuses on algorithms from the NIST 19 synthetic data challenge in terms of the marginal distribution, joint distribution and correlation; [8], which focuses on algorithms before 2016 in terms of the statistical utility; [4], which proposes a general framework for evaluating the quality of private synthetic data; and [37], which proposes a framework SDGym to benchmark the performance of synthetic data. However, none of these works both include a representative set of state-of-the-art algorithms and cover a representative set of metrics.

*Work done while at Tumult Labs.

Inspired by DPBench [17], we focus on benchmarking differentially private synthetic data generation algorithms selected from a specific set of inclusion criteria. Most DP synthetic data generation algorithms learn a model over the data from which synthetic data records are sampled. We categorize the algorithms included in our study into three broad classes: **GAN-based** methods learn a generative adversarial network (GAN) privately, mainly by adding noise to the gradient calculation; **Marginal-based** methods measure a subset of the low-order marginals and use them to fit a graphical model; and **Workload-based** algorithms iteratively improve their model to reduce approximation error on workload queries.

We evaluate these mechanisms across different datasets and privacy budgets on whether the synthetic data can preserve the distribution of individual and pairs of attributes, pairwise correlation and on the accuracy of an ML classification model. Our experiments reveal a number of findings:

1. Many mechanisms, especially GAN-based mechanisms, often fail to preserve the most basic statistics of the data distribution — their one way marginals. Moreover, these mechanisms fail to beat simple baseline mechanisms on other more interesting metrics.
2. No single mechanism is best on every dataset and task, and privacy budget considered. However, marginal-based mechanisms consistently rank among the best.
3. Marginal-based methods expect discrete data, and proper discretization is essential to get good performance on numerical attributes. We found that using PrivTree [39] to discretize numerical attributes is far more effective than equal-width discretization.

Methodology

In this section, we describe the mechanisms included in this study (and the justification for inclusion), the tasks considered, the datasets evaluated, as well as any modifications necessary to run the mechanisms on our datasets.

Mechanisms

We consider five inclusion criteria for selecting the mechanisms for this benchmark study, enumerated below:

1. **End-to-End DP**: It is claimed to be an end-to-end differential private algorithm that takes a tabular dataset as

input and generates a synthetic data of the same schema.

2. **Tabular Data**: It supports tabular data that could have numerical and/or categorical columns. The associated publication includes experiments on tabular data.
3. **Publication Venue**: It is published in a top conference/journal or included in a well known library. For example, we consider academic venues of SIGMOD, VLDB, CCS, NeurIPS, ICML, PETS and JPC and the open-source libraries of SmartNoise and Gretel. Algorithms from other conferences/journals and libraries are left for future work.
4. **Publicly Available Source Code**: Its source code is accessible to the public (e.g. either available on GitHub or linked in the paper describing the work).
5. **No Public Data**: It requires no public data.

Table 1 lists the mechanisms included in this benchmark, categorizing them by type. Included in this table is `GretelRNN`, which satisfied the inclusion criteria but is not shown in the experimental results because we found that even when the epsilon that it claims is over one million, it generates an empty dataset after one hour of the generation stage due to the high rejection rate of invalid samples.

Algorithm	Code	Type
`MST` [27]	[24]	Marginal
`MWEM-PGM` [26]	[25]	Marginal
`PrivBayes` [38, 40]	[30]	Marginal
`DPGAN` [36]	[31]	GAN
`DPCTGAN` [29]	[31]	GAN
`PATEGAN` [18]	[31]	GAN
`PATECTGAN` [29]	[31]	GAN
`FEM` [35]	[34]	Workload
`RAP` [5]	[28]	Workload
`Kamino` [15]	[14]	Other
`RON-GAUSS` [10]	[7]	Other
`GretelRNN` [2]	[2]	Other

Table 1: Mechanisms included in our study.

While we expect all the mechanisms to be able to take a dataset with mixed-type columns as input, some of them only accept categorical datasets or numerical datasets. For mechanisms that expect numerical data, we one-hot encode all categorical features. For mechanisms that expect categorical data, we discretize all numerical features. We considered two approaches for discretization: an equal-width binning strategy, and a strategy based on PrivTree [39]. We find that PrivTree binning was never worse than equi-width binning in most of the cases and lead to significant improvements for some metrics. Details omitted due to space.

Three mechanisms, `MWEM-PGM`, `FEM` and `RAP`, also require a workload as input. For datasets that include a classification label (see next section), we set the workload to be all 2- and 3-way marginals that include the label as one of the attributes. For other datasets, we set the workload to be all 2-way marginals. For all algorithms except Kamino, we use default hyper-parameters. For Kamino, the search algorithm (Algorithm 6 from [15]) was not included in the available implementation, so we implemented a variant of it.

Datasets

We consider seven datasets with different characteristics, numbers of records, and column types. Datasets *Car* and *Mushroom* contain only categorical attributes; *Scooter* contains only numerical attributes; all other datasets contain a mix of attribute types. Most datasets have a classification label. All datasets are from the UCI machine learning repository [12] except *Scooter* which is from Gretel [1].

Name	Records	Cat.	Numeric	Label
Shopping	12330	9	10	Yes
Adult	32561	9	6	Yes
Bank	45211	13	8	Yes
Census	299285	29	12	Yes
Car	1728	7	0	Yes
Mushroom	8124	23	0	Yes
Scooter	27715	0	5	No

Table 2: Summary of datasets.

Metrics

We consider four groups of metrics to measure the goodness-of-fit of the synthetic data generated by each algorithm. Each group might include more than one metric but with similar goals.[1] These metrics are inspired by SDGym [3]. For the first three metric groups, numerical attributes are discretized into 19 bins of equal-depth (based on the original data). Since the algorithms may generate synthetic data that lies outside of this range, an additional bin is added to each end of the range.

1. **Individual Attribute Distribution Similarity (Ind)** This group of metrics measures the similarity of one-way marginals between the synthetic data and the original data. We use total variation distance (TVD), to measure the distance between two one-dimensional distributions, and use 1-TVD as the score. We report the average score over all one-way marginal as the final score.
2. **Pairwise Attribute Distribution Similarity (Pair)** Similar to the Individual Distribution Similarity, this group of metrics measures the TVD for each two-way marginal and we average across all attribute pairs.
3. **Pairwise Correlation Similarity (Corr)** We use Cramer's V with bias correction [6] to measure the correlation between two attributes, and, following convention, discretize it into one of four levels: V in [0, .1) is low, [.1, .3) is weak, [.3, .5) is middle and [.5, 1) is strong. The metric *CorAcc* measures the accuracy of correlation levels, reporting the fraction of pairs where the synthetic and original data assign the same correlation level.
4. **Classification Accuracy (F1)** We use the synthetic dataset to train an XGBoost classifier and use it to make predictions on the original data. The score is reported by the f1 score using macro average. This metric category only applies to datasets that have a class label.

[1]For brevity, we include a single metric per group. In the full version of this paper, multiple metrics per group are considered.

Mechanism	GT	Ind	Pair	Corr	F1
MST	69%	95%	81%	52%	44%
MWEM-PGM	19%	0%	14%	29%	33%
PrivBayes	9%	0%	0%	19%	17%
Kamino	1%	0%	0%	0%	6%
FEM	0%	0%	0%	0%	0%
RAP	1%	0%	0%	5%	0%
PATECTGAN	4%	5%	5%	5%	0%
DPCTGAN	1%	0%	0%	5%	0%
RonGauss	0%	0%	0%	0%	0%
DPGAN	0%	0%	0%	0%	0%
PATEGAN	0%	0%	0%	0%	0%

(a) **Optimal Rate**

Mechanism	GT	Ind	Pair	Corr	F1
MST	1.56	1.05	1.24	2.00	2.00
MWEM-PGM	2.88	2.76	2.62	3.86	2.17
PrivBayes	4.54	5.43	5.67	3.29	3.67
Kamino	5.26	4.27	4.93	7.87	3.67
FEM	4.91	4.30	4.35	5.95	5.06
RAP	5.94	5.83	5.39	7.17	5.27
PATECTGAN	6.17	6.45	5.90	4.90	7.65
DPCTGAN	6.56	6.84	6.68	5.16	7.75
RonGauss	7.35	7.06	7.11	7.61	7.61
DPGAN	8.46	9.06	9.44	6.78	8.60
PATEGAN	8.99	9.85	9.70	7.05	9.41

(b) **Average Ranking**

Figure 1: Overview of mechanisms in terms of optimal rate and average ranking across datasets, epsilons, and metrics stratified by metric groups. GT means "Grand Total."

Dataset	Indep..	MST	PrivBa..	MWE..	RAP	Kamino	FEM	DPGAN	PATEC..	RonGa..	DPCT..	PATE..
Adult	0.98	0.98	0.74	0.95	0.70	0.85	0.76	0.59	0.57	0.59	0.59	0.46
Mushroom	0.99	0.99	0.97	0.95	0.88	0.78	0.78	0.70	0.68	0.68	0.67	0.58

(a) 1-TVD for individual attribute distributions

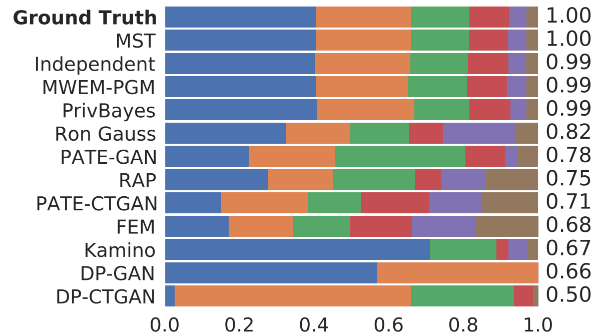
Dataset	MST	MWE..	PrivBa..	Indep..	PATEC..	DPCT..	FEM	RonGa..	DPGAN	Kamino	PATE..	RAP
Adult	0.71	0.66	0.60	0.53	0.53	0.50	0.49	0.42	0.35	0.09	0.38	0.32
Mushroom	0.36	0.42	0.15	0.13	0.13	0.13	0.18	0.36	0.40	0.38	0.34	0.34

(b) Correlation accuracy (*CorAcc*)

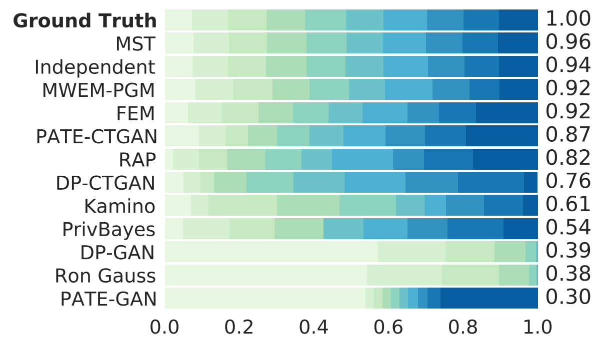
Dataset	MST	MWE..	FEM	PrivBa..	RonGa..	Kamino	RAP	DPGAN	Indep..	PATE..	PATEC..	DPCT..
Adult	0.63	0.74	0.44	0.66	0.39	0.66	0.55	0.33	0.45	0.35	0.43	0.39
Mushroom	0.98	0.97	0.90	0.77	0.77	0.76	0.70	0.69	0.50	0.50	0.43	0.36

(c) Classification accuracy, measured by f1 score, of an XGBoost classifier. The F1 score by training on the original data is 0.86 for Adult and 1.0 for Mushroom.

Figure 2: Performance metrics for synthetic data algorithms at $\epsilon = 1.0$



(a) Distributions of the *relationship* attribute; a categorical attribute with six possible values (shown as different colors).



(b) Distributions of the *age* attribute; a numerical attribute discretized by quantiles (which is why the *Ground Truth* appears uniform).

Figure 3: One-way marginal distributions for the original *Adult* dataset and for a sample synthetic dataset generated by each algorithm ($\epsilon = 1.0$) in descending order by similarity (1-TVD, shown to the right of each row).

Findings

We use SDGym [37] as the platform for all the experiments. The privacy parameter ϵ varies within $\{0.1, 1.0, 10.0\}$. In this section, we briefly summarize our main findings.

F1: No algorithm dominates. We consider a mechanism "optimal" for a particular combination of dataset, epsilon, and metric if that mechanism achieves the best performance (averaged over trials) according to the metric. The optimal rate, shown in Fig. 1a, is the frequency at which a mechanism is optimal for a particular category of metric. Any algorithm that has a non-zero optimal rate means that the algorithm performs the best on at least one combination of dataset, epsilon, and metric. Over half of the algorithms have a non-zero optimal rate.

F2: While no algorithm dominates, marginal-based approaches are highly ranked and `MST`*, in particular, is the top-ranked algorithm across all metrics*. To get a sense of the overall best performing algorithm, we rank the algorithms according to each metric and then average the rankings. In Fig. 1b, we report the average ranking, stratified by category of metric; we also report the average ranking across all metrics ("GT"). The overall average rank of `MST` is 1.56 indicating that it is frequently the best algorithm, which is also consistent with the results of Fig. 1a.

F3: Many mechanisms fail to accurately preserve the distributions of individual attributes (1-way marginals). Fig. 2a reports the average similarity (1-TVD) of individual attribute distributions for two of the datasets in our benchmark, *Adult* and *Mushroom*. Several algorithms have an average similarity of less than 0.75. `PrivBayes` has uneven performance, doing well on *Mushroom* and worse on *Adult*; we hypothesize this is due to how PrivBayes discretizes numerical attribues (*Mushroom* has no numerical attributes).

To gain some intuition for how well algorithms are preserving attribute distributions, we display some representative examples in Fig. 3 from the *Adult* dataset. Fig. 3a uses a stacked bar chart to compactly display the distribution of the *relationship* attribute. The first row is the distribution in the original data (ground truth) and the remaining rows are the distributions in the synthetic datasets produced by the algorithms, ordered by their similarity to the ground truth (1-TVD is reported to the right of each row). When 1-TVD is below 0.75, the distortion is visually apparent. Some algorithms (`DP-GAN`, `DP-CTGAN`) have highly skewed distributions; others appear uniform (`FEM`) even though the original data is non-uniform. Fig. 3b shows the distributions of the numerical attribute *age* (after it was discretized).

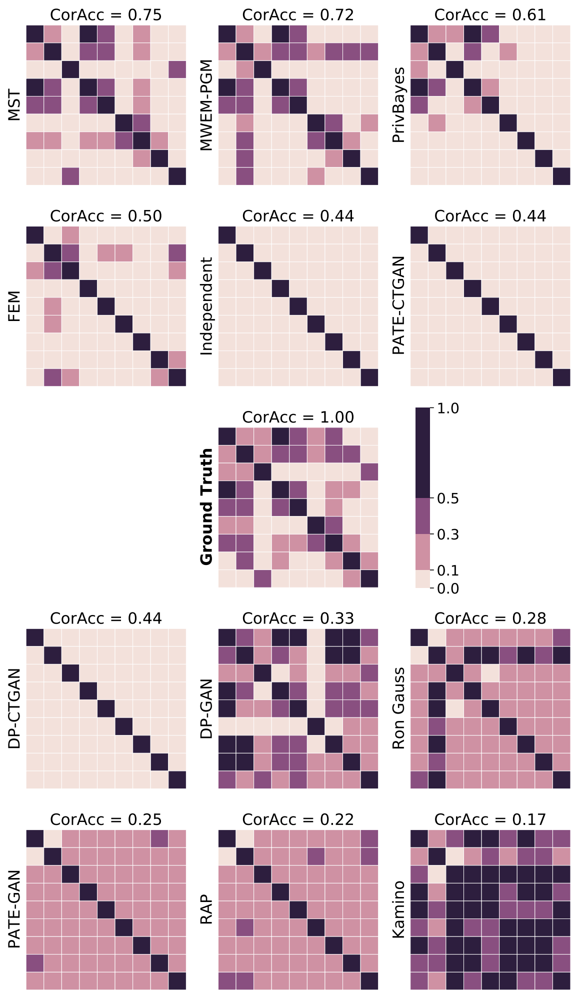


Figure 4: Correlation heatmaps for all pairwise categorical attributes from the *Adult* dataset. A heatmap is shown for the original data, *Ground Truth* (center), and for a sample synthetic dataset generated by each algorithm at $\epsilon = 1.0$. Attributes are sorted by domain sizes.

In addition to looking at individual attribution distributions, we also evaluate pairwise attribute correlations. Fig. 2b reveals the extent to which correlations are accurately preserved. It reports the *CoreAcc* metric for datasets *Adult* and *Mushroom*. As a baseline for comparison, we include `Independent`, an algorithm that assumes all columns are statistically independent (uncorrelated) and generates synthetic data by sampling attribute values from distributions estimated from 1-way marginals perturbed with Laplace noise. We use a divergent color scheme to indicate whether it is above (orange) or below (blue) the baseline.

The results in Fig. 2b give us two main findings. *F4: In terms of preserving attribute correlations, Marginal-based algorithms consistently obtain the highest correlation accuracy.* And, *F5: Many algorithms fail to preserve correlations more accurately than independent, a simple baseline that generates uncorrelated data.*

To gain some intuition about correlations, we look more closely at the correlation accuracy on the *Adult* dataset. Fig. 4 shows correlation heatmaps for the original data (*Ground Truth*, center plot) and for synthetic datasets generated by the algorithms. In each heatmap, a cell corresponds to an attribute pair and darker cells indicate stronger correlation (the colors are discretized to the four correlation levels described earlier). The figure shows that marginal-based algorithms (top row) do fairly well (*CorAcc*=0.75 means 75% of the colored cells match the ground truth figure) though some correlations are not captured. Several algorithms (`FEM`, `PATE-CTGAN`, `DP-CTGAN`) have accuracy matching the baseline `Independent`. The correlation plots show why: the synthetic data generated by these algorithms has attributes that appear to be statistically independent (uncorrelated), matching the independent baseline. The remaining algorithms have accuracy that is *lower* than the baseline and it appears that this is due to those techniques introducing *spurious* correlation.

F6: The synthetic data produced by marginal-based approaches `MST` and `MWEM-PGM` is of sufficient quality that it can be used to train an accurate classifier, nearly matching the performance of a classifier trained on the original data. In Fig. 2c, we report how well the synthetic data preserves the ability to train a classifier. It shows the f1 score of an XGBoost classifier trained on the synthetic data. On *Mushroom*, the classifiers trained on the synthetic data from `MST` and `MWEM-PGM` achieve nearly perfect accuracy; on *Adult*, `MWEM-PGM` achieves the highest f1 score of 0.74, which approaches the f1 on the original data of 0.86. The relatively strong performance of `MWEM-PGM` may reflect the fact that its strategy is tuned to support classifier learning by favoring marginals that include the class label.

F7: The synthetic data produced by GAN-based approaches yields classifers that are generally no more accurate than a simple majority classifier. In Fig. 2c, we include baseline algorithm `Independent`. Since this algorithm models each attribute independently, a classifier trained on its synthetic data can be no more accurate than a classifier that always predicts the majority label. In this figure, we again use a divergent color scheme to compare performance to this basline and we see the GAN-based approaches often have an f1 score below the baseline.

Conclusion

We presented a systematic benchmark study of differentially private synthetic data generation algorithms that can generate tabular data. We considered a variety of algorithms including GAN-based, Marginal-based and Workload-based methods and evaluated their utility in terms of how well they preserve low dimensional statistics, pairwise correlations and ML classification accuracy. We found that Marginal-based methods consistently outperformed other methods, and GAN-based methods were unable to preserve the 1-dimensional statistics of tabular data. Our research motivates future research directions that include developing better GAN methods for tabular data, methods for pre-processing categorical/numeric data types, and identifying methods to choose the best synthetic data algorithms given a dataset.

References

[1] Scooter Dataset. https://github.com/gretelai/gretel-synthetics/blob/master/examples/data/uber_scooter_rides_1day.csv, 2020. [Online; accessed 2-Mar-2021].

[2] Gretel Synthetic. https://github.com/gretelai/gretel-synthetics/tree/v0.15.10, 2021. [v 0.15.10; released 20-May-2021].

[3] SDGym. https://github.com/sdv-dev/SDGym, 2021. [Online; accessed 17-June-2021].

[4] C. Arnold and M. Neunhoeffer. Really useful synthetic data - A framework to evaluate the quality of differentially private synthetic data. *CoRR*, abs/2004.07740, 2020. URL https://arxiv.org/abs/2004.07740.

[5] S. Aydöre, W. Brown, M. Kearns, K. Kenthapadi, L. Melis, A. Roth, and A. A. Siva. Differentially private query release through adaptive projection. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 457–467. PMLR, 2021. URL http://proceedings.mlr.press/v139/aydore21a.html.

[6] W. Bergsma. A bias-correction for cramér's v and tschuprow's t. *Journal of the Korean Statistical Society*, 42(3):323–328, 2013.

[7] BorealisAI. Impelementation of RonGauss. https://github.com/BorealisAI/private-data-generation/tree/737df84e3f1ee521190cc2b62ce408ad708206e6, 2021. [released 21-Apr-2021].

[8] C. M. Bowen and F. Liu. Comparative study of differentially private data synthesis methods. *Statistical Science*, 35(2):280–307, 2020.

[9] C. M. Bowen and J. Snoke. Comparative study of differentially private synthetic data algorithms from the nist pscr differential privacy synthetic data challenge. *arXiv preprint arXiv:1911.12704*, 2019.

[10] T. Chanyaswad, C. Liu, and P. Mittal. Ron-gauss: Enhancing utility in non-interactive private data release. *Proc. Priv. Enhancing Technol.*, 2019(1):26–46, 2019. doi: 10.2478/popets-2019-0003. URL https://doi.org/10.2478/popets-2019-0003.

[11] D. Chen, T. Orekondy, and M. Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[12] D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[13] J. Fan, T. Liu, G. Li, J. Chen, Y. Shen, and X. Du. Relational data synthesis using generative adversarial networks: A design space exploration. *Proc. VLDB Endow.*, 13(11):1962–1975, 2020. URL http://www.vldb.org/pvldb/vol13/p1962-fan.pdf.

[14] C. Ge. Impelementation of Kamino. https://github.com/cgebest/kamino/tree/0e4f63a2199fa140fb050b50516582f616d0583f, 2021. [released 19-May-2019].

[15] C. Ge, S. Mohapatra, X. He, and I. F. Ilyas. Kamino: Constraint-aware differentially private data synthesis. *Proc. VLDB Endow.*, 14(10):1886–1899, 2021. URL http://www.vldb.org/pvldb/vol14/p1886-ge.pdf.

[16] F. Harder, K. Adamczewski, and M. Park. DP-MERF: differentially private mean embeddings with randomfeatures for practical privacy-preserving data generation. In A. Banerjee and K. Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1819–1827. PMLR, 2021. URL http://proceedings.mlr.press/v130/harder21a.html.

[17] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, and D. Zhang. Principled evaluation of differentially private algorithms using dpbench. In F. Özcan, G. Koutrika, and S. Madden, editors, *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 139–154. ACM, 2016. doi: 10.1145/2882903.2882931. URL https://doi.org/10.1145/2882903.2882931.

[18] J. Jordon, J. Yoon, and M. van der Schaar. PATE-GAN: generating synthetic data with differential privacy guarantees. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1zk9iRqF7.

[19] H. Li, L. Xiong, L. Zhang, and X. Jiang. Dpsynthesizer: Differentially private data synthesizer for privacy preserving data sharing. *Proc. VLDB Endow.*, 7(13): 1677–1680, 2014. doi: 10.14778/2733004.2733059. URL http://www.vldb.org/pvldb/vol7/p1677-li.pdf.

[20] K. Li and J. Malik. Implicit maximum likelihood estimation. *CoRR*, abs/1809.09087, 2018. URL http://arxiv.org/abs/1809.09087.

[21] N. Li, Z. Zhang, and T. Wang. Dpsyn: Experiences in the NIST differential privacy data synthesis challenges. *CoRR*, abs/2106.12949, 2021. URL https://arxiv.org/abs/2106.12949.

[22] T. Liu, G. Vietri, T. Steinke, J. R. Ullman, and Z. S. Wu. Leveraging public data for practical private query release. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6968–6977. PMLR, 2021. URL http://proceedings.mlr.press/v139/liu21w.html.

[23] T. Liu, G. Vietri, and Z. S. Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *CoRR*, abs/2106.07153, 2021. URL https://arxiv.org/abs/2106.07153.

[24] R. McKenna. Impelementation of MST. https://github.com/ryan112358/private-pgm/blob/aae58df3dc27b9d7ceb9eeab75a02549b3bc870e/mechanisms/mst.py, 2021. [released 19-Oct-2021].

[25] R. McKenna. Impelementation of MWEM-PGM. https://github.com/ryan112358/private-pgm/blob/aae58df3dc27b9d7ceb9eeab75a02549b3bc870e/mechanisms/mwem%2Bpgm.py, 2021. [released 19-Oct-2021].

[26] R. McKenna, D. Sheldon, and G. Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

[27] R. McKenna, G. Miklau, and D. Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *CoRR*, abs/2108.04978, 2021. URL https://arxiv.org/abs/2108.04978.

[28] A. Research. Impelementation of RAP. https://github.com/amazon-research/relaxed-adaptive-projection/tree/1960dc66a28ad2a6b1a5670ec1d7102bde4fd034, 2021. [released 29-Jun-2021].

[29] L. Rosenblatt, X. Liu, S. Pouyanfar, E. de Leon, A. Desai, and J. Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

[30] SDGym. Impelementation of PrivBayes. https://github.com/sdv-dev/SDGym/tree/59cf1b4007661943aa3283473156cfd44c5fc527/privbayes, 2019. [released 6-May-2019].

[31] SmartNoise. Impelementation of DPGAN, DPCTGAN, PATEGAN and PATECTGAN. https://github.com/opendp/smartnoise-sdk/tree/a99f004732d7779f082a09037c5204165a94e81e/sdk/opendp/smartnoise/synthesizers/pytorch/nn, 2021. [released 13-Jul-2021].

[32] J. Snoke and A. B. Slavkovic. pmse mechanism: Differentially private synthetic data with maximal distributional similarity. In J. Domingo-Ferrer and F. Montes, editors, *Privacy in Statistical Databases - UNESCO Chair in Data Privacy, International Conference, PSD 2018, Valencia, Spain, September 26-28, 2018, Proceedings*, volume 11126 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2018. doi: 10.1007/978-3-319-99771-1_10. URL https://doi.org/10.1007/978-3-319-99771-1_10.

[33] R. Torkzadehmahani, P. Kairouz, and B. Paten. DP-CGAN: differentially private synthetic data and label generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 98–104. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPRW.2019.00018. URL http://openaccess.thecvf.com/content_CVPRW_2019/html/CV-COPS/Torkzadehmahani_DP-CGAN_Differentially_Private_Synthetic_Data_and_Label_Generation_CVPRW_2019_paper.html.

[34] G. Vietri. Impelementation of FEM. https://github.com/giusevtr/fem/tree/f538f61c564dbc38c2fc8e63b8af0a6f7bb5ba0a, 2021. [released 10-Jun-2021].

[35] G. Vietri, G. Tian, M. Bun, T. Steinke, and Z. S. Wu. New oracle-efficient algorithms for private synthetic data release. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9765–9774. PMLR, 2020. URL http://proceedings.mlr.press/v119/vietri20b.html.

[36] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018. URL http://arxiv.org/abs/1802.06739.

[37] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: private data release via bayesian networks. In C. E. Dyreson, F. Li, and M. T. Özsu, editors, *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 1423–1434. ACM, 2014. doi: 10.1145/2588555.2588573. URL https://doi.org/10.1145/2588555.2588573.

[39] J. Zhang, X. Xiao, and X. Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 International Conference on Management of Data*, pages 155–170, 2016.

[40] 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)*, 42(4):1–41, 2017.

[41] Z. Zhang, T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, and Y. Zhang. Privsyn: Differentially private data synthesis. *CoRR*, abs/2012.15128, 2020. URL https://arxiv.org/abs/2012.15128.