Noise-Aware Statistical Inference with Differentially Private Synthetic Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing work has shown that analysing differentially private (DP) synthetic data as if it were real does not produce valid uncertainty estimates. We tackle this problem by combining synthetic data analysis techniques from the field of multiple imputation (MI), and synthetic data generation using a novel noise-aware (NA) synthetic data generation algorithm NAPSU-MQ into a pipeline NA+MI that allows computing accurate uncertainty estimates for population-level quantities from DP synthetic data. Our experiments demonstrate that the pipeline is able to produce accurate confidence intervals from DP synthetic data.

1 Introduction

Availability of data for research is constrained by the dilemma between privacy preservation and potential gains obtained from sharing. One solution to the dilemma is to have the *data holder* release a synthetic dataset based on a real dataset. *Data analysts* can then use the synthetic data for their *downstream analysis*. Privacy-protection of the synthetic data can be guaranteed by employing differential privacy (DP) [10], which offers provable protection.

While many works have studied synthetic data generation under DP [2, 4, 6, 7, 8, 13, 14, 17, 20, 21, 22, 24, 25, 26, 27, 29, 35, 37, 38, 39], the analysis of synthetic data has received far less attention. This is especially apparent for uncertainty estimates, like p-values and confidence intervals. It is clear that synthetic data generation adds additional uncertainty [30], especially under DP, which requires adding extra noise. However, only two of the mentioned works [26, 29] consider uncertainty estimation of query values on the real data, and none of them consider uncertainty estimation of population parameters.

We illustrate the issue with treating synthetic data as if it were real using a simple toy data experiment. We generate 3-dimensional binary data, where one variable is generated from logistic regression on the other two, serving as the original dataset. Then, we generate synthetic data from the original data and compute confidence intervals for the coefficients of the logistic regression from the synthetic data. We describe the setting in more detail in Supplementa Section F. As shown by Figure 1, the existing algorithms PGM [27], PEP [21], and RAP [2] do not provide valid confidence intervals, as they treat the synthetic data as real data. Only our method NA+MI and PrivLCM [29] are able provide valid confidence intervals. However, PrivLCM is too conservative and produces much wider intervals than NA+MI, as shown by Figure 3a.

2 The NA+MI Pipeline

To compute accurate uncertainty estimates from DP synthetic data, we build on the framework of Rubin [33]. He proposed generating multiple datasets from a posterior predictive distribution, and

Submitted to NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research. Do not distribute.

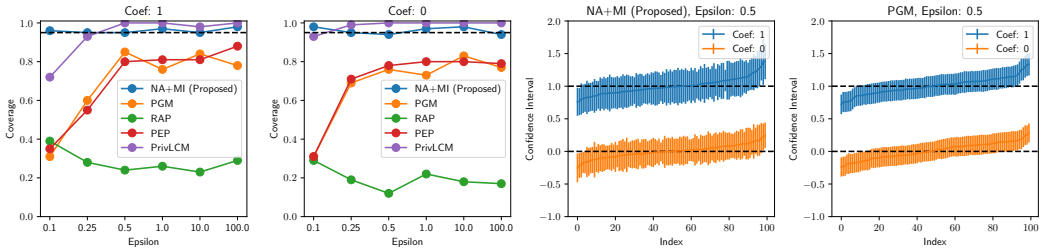


Figure 1: Toy data experiment results of logistic regression on 3 binary variables, showing that all algorithms apart from ours and PrivLCM are overconfident, even with almost no privacy ($\epsilon = 100$). The first two panels from the left show the fraction of the 95% confidence intervals that contain the true parameter value in 100 repeats, with the target confidence level of 95% highlighted by a black line. The third panel shows the confidence intervals from synthetic data generated by our mechanism for $\epsilon = 0.5$, and the fourth panel shows the confidence intervals from PGM. The third and fourth panels show that the overconfidence stems from intervals that are too narrow, a result of not accounting for all uncertainty.

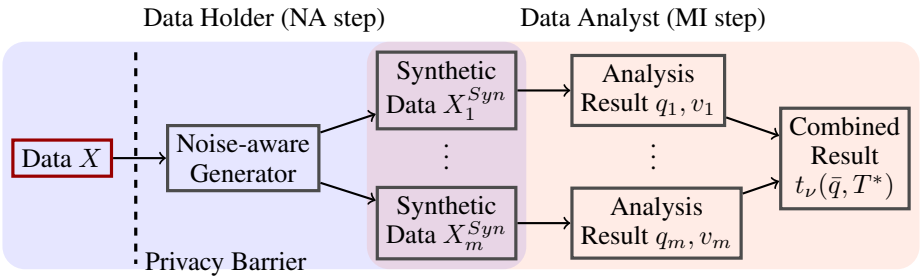


Figure 2: NA+MI pipeline for noise-aware DP synthetic data generation and statistical inference. The nodes shaded in blue are computed by the data holder, and the nodes shaded in orange are computed by the data analyst. All nodes except Data (with red border) can be released to the public. The synthetic datasets can be generated by either party because the Generator is also released by the data holder.

using simple combining rules called Rubin's rules to account for the extra uncertainty that synthetic data generation introduces. This workflow is modeled after *multiple imputation* (MI), where it is used to deal with missing data.

Rubin's rules are able to account for the extra noise added by DP when the posterior predictive distribution that generates the synthetic data accurately reflects the noise addition process. We call Bayesian inference algorithms that account for DP noise *noise-aware* (NA).

Figure 2 summarises our proposed pipeline, which we call NA+MI. First, the data holder runs inference on a noise-aware Bayesian model using the private data, which we call the NA step. After the inference, the data holder generates multiple synthetic datasets. The data holder can also release the posterior distribution in addition to the synthetic datasets, so that the analyst can also generate synthetic datasets if needed.

For each synthetic dataset, the analyst runs their analysis, producing a point estimates q_i and a variance estimate v_i. These are combined with Rubin's rules, which produce a t-distribution that the analyst can use to compute confidence intervals or hypothesis tests.

3 Noise-aware Synthetic Data Generation

To implement the NA step, we develop an algorithm called *Noise-Aware Private Synthetic data Using Marginal Queries* (NAPSU-MQ), that generates synthetic data from discrete tabular datasets using the noisy values of preselected marginal queries.

Data and Marginal Queries NAPSU-MQ uses tabular datasets of d discrete variables, where the domains of the discrete variables, as well as the number datapoints n are known. We denote the set of possible datapoints by $\mathcal{X}$, and the set of possible datasets by $\mathcal{X}^n$.

Definition 3.1. *A marginal query of variables I and value x is a function $a\colon \mathcal{X} \to \{0,1\}$ that takes a datapoint as input and returns 1 if the variables in I in the datapoint have the value x, and 0 otherwise. For a dataset $X \in \mathcal{X}^n$, we define $a(X) = \sum_{i=1}^{n} a(X_i)$, where X_i is the i:th datapoint in X.*

When I has k variables, a is called a k-way marginal query.

When evaluating multiple marginal queries $a_1, \ldots, a_{n_q}$, we concatenate their values to a vector-valued function $a\colon \mathcal{X} \to \{0,1\}^{n_q}$. We call the concatenation of marginal queries for all possible values of variables I the *full set of marginals on I* [1].

Implemetation In order to implement the NA step, the data holder needs to generate synthetic data from the posterior of a noise-aware Bayesian model. Bernstein and Sheldon [5] develop noise-aware Bayesian inference under DP for simple exponential family models. We implement the NA step by generalising their algorithm to arbitrary marginal queries, using the maximum entropy distribution of the marginal queries as our exponential family model.

The distribution with maximal entropy with the correct expected marginal query values is

$$P(x) = \exp(\theta^T a(x) - \theta_0(\theta)) \tag{1}$$

for some parameters $\theta \in \mathbb{R}^k$ [36], where k is the number of queries. $\theta_0(\theta)$ is the normalising constant of the distribution. We denote this distribution by MED_θ, and use MED_θ^n to denote the distribution of n i.i.d. samples from MED_θ.

The Bayesian model we consider is derived from the generative process of the noisy query values, which are observed. Assuming that the data generating process is MED_θ, and knowing that the Gaussian mechanism adds noise with variance σ_{DP}^2, we get the probabilistic model

$$\theta \sim \mathrm{Prior}, \quad X \sim \mathrm{MED}_\theta^n, \quad s = a(X), \quad \tilde{s} \sim \mathcal{N}(s, \sigma_{DP}^2 I). \tag{2}$$

In principle, we could now sample from the posterior $p(\theta \mid \tilde{s})$, with s marginalised out. In practice, the marginalisation is not feasible, as s is a discrete variable with a very large domain.

However, s is a sum of the query values for individual datapoints, so asymptotically s has a normal distribution according to the central limit theorem. We can substitute the normal approximation for s into the model, which allows us to easily marginalise s out, resulting in

$$\theta \sim \mathrm{Prior}, \quad \tilde{s} \sim \mathcal{N}(n\mu(\theta), n\Sigma(\theta) + \sigma_{DP}^2 I), \tag{3}$$

where $\mu(\theta)$ and $\Sigma(\theta)$ denote the mean and covariance of $a(x)$ for a single sample $x \sim \mathrm{MED}_\theta$.

$\mu(\theta)$ and $\Sigma(\theta)$ can be computed efficiently using techniques from probabilistic graphical models and autodifferentiation when the graph corresponding to the selected queries is sparse. We can then compute the approximate unnormalised density, which allows us to use off-the-shelf sampling algorithms, like the Laplace approximation or the No-U-Turn sampler (NUTS) [15].

For the prior, we choose another Gaussian distribution with mean 0 and standard deviation 10, which is a simple and weak prior, but other priors could be used.

We can make MED_θ identifiable by pruning the queries using the *canonical parameterisation* [19] of MED_θ, which we describe in more detail in Supplemental Section C. The algorithm is summarised in Algorithm 1.

4 Privacy of NAPSU-MQ

To quantify privacy, we use differential privacy (DP) [9, 10], which aims to rigorously quantify the privacy loss resulting from releasing the results of an algorithm. DP algorithms are also called *mechanisms*.

[1] Some existing works [27] use the term marginal query for the full set of marginal queries. We chose this terminology because we deal with individual marginal queries in Supplemental Section C.

Algorithm 1: NAPSU-MQ

Input: Real data X, marginal queries a, number of synthetic datasets m, size of synthetic datasets n_{Syn}, privacy bounds ϵ, δ.
Output: Posterior distribution $p(\theta|\tilde{s})$, synthetic datasets $X_1^{Syn}, \ldots, X_m^{Syn}$.
$a^* \leftarrow$ Canonical queries for a (Section C);
$s \leftarrow a^*(X)$;
$\Delta_2 \leftarrow$ Sensitivity of s (Theorem A.2);
$\sigma_{DP}^2 \leftarrow$ Required noise variance for (ϵ, δ)-DP with sensitivity Δ_2 (Theorem A.3);
Sample $\tilde{s} \sim \mathcal{N}(s, \sigma_{DP}^2)$;
Run Bayesian inference algorithm to find $p(\theta|\tilde{s})$ (Section 3);
Sample $\theta_i \sim p(\theta|\tilde{s})$ and $X_i^{Syn} \sim \mathrm{MED}_{\theta_i}^{n_{Syn}}$ for $1 \leq i \leq m$;
return $p(\theta|\tilde{s})$, $X_1^{Syn}, \ldots, X_m^{Syn}$

Definition 4.1. *A mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for all neighbouring datasets X, X' and all measurable output sets S*

$$P(\mathcal{M}(X) \in S) \leq e^{\epsilon} P(\mathcal{M}(X') \in S) + \delta. \tag{4}$$

The neighbourhood relation in the definition is domain specific. We use the *substitute* neighbourhood relation for tabular datasets, where datasets are neighbouring if they differ in at most one datapoint.

The mechanism we use to release marginal query values under DP is the *Gaussian mechanism* [9].

Definition 4.2. *The Gaussian mechanism with noise variance σ^2 adds Gaussian noise to the value of a function $f\colon \mathcal{X}^n \to \mathbb{R}^k$ for input data X: $\mathcal{M}(X) = f(X) + \mathcal{N}(0, \sigma^2 I)$.*

An important property of DP is post-processing immunity: post-processing the result of a DP-algorithm does not weaken the privacy bounds.

Theorem 4.3 (Dwork and Roth [11])**.** *Let $\mathcal{M}$ be an (ϵ, δ)-DP mechanism, and let f be any algorithm. Then the composition $f \circ \mathcal{M}$ is (ϵ, δ)-DP.*

Because of post-processing immunity, we can release marginal query values with the Gaussian mechanism to obtain privacy bounds, and make arbitrary use of the noisy query values without weakening the privacy bounds. This makes the privacy proof of NAPSU-MQ very simple:

Theorem 4.4. *NAPSU-MQ (Algorithm 1) is (ϵ, δ)-DP with regards to the real data X.*

Proof. The returned values of Algorithm 1 only depend on the real data X through $\tilde{s}$. Releasing $\tilde{s}$ is (ϵ, δ)-DP due to the selection of σ_{DP}^2 with Theorem A.3. Computing the returned values from $\tilde{s}$ is post-processing, so by Theorem 4.3, NAPSU-MQ is (ϵ, δ)-DP. □

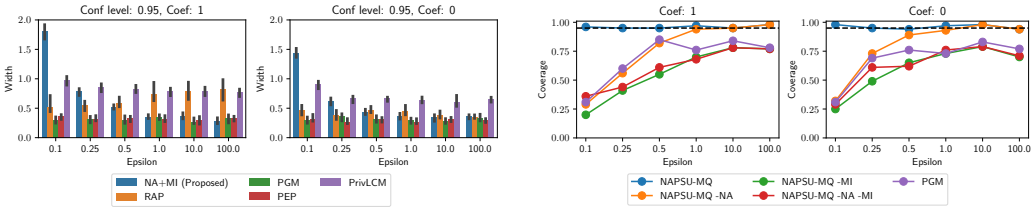
5 Experiments

In this section, we summarise our experiments and their results. We give detailed descriptions of hyperparameters in Supplemental Section E, of the toy data experiment in Supplemental Section F, and of the Adult experiment in Supplemental Section G. We describe an additional experiment with the UCI US Census (1990) dataset in Supplemental Section I.

Ablation Study In addition to the comparison presented in Figure 1, we conducted an ablation study on the same toy data setting to show that both multiple imputation and noise-awareness are necessary for accurate confidence intervals. The results are presented in Figure 3b. Without both multiple imputation and noise-awareness, NA+MI is overconfident like PGM, except for $\epsilon \geq 1$, where noise-awareness is not required.

5.1 Adult Dataset

Our main experiment evaluates the performance of NAPSU-MQ on the UCI Adult dataset [18]. We include 10 of the original 15 columns to remove redundant columns and keep runtimes manageable,



(a) (b)

Figure 3: (a) Toy data confidence intervals widths. NA+MI produces slightly wider intervals than PGM, or PEP for $\epsilon > 0.1$, which is necessary to account for DP noise. PrivLCM produces much wider intervals. (b) Ablation study on the toy data. "-NA" refers to removing noise-awareness, and "-MI" refers to removing multiple imputation. Unless both are included, NAPSU-MQ is overconfident like PGM except for $\epsilon \geq 1$ where noise-awareness is not necessary.

and discretise the continuous columns. After dropping rows with missing values, there are $n = 46\,043$ rows. The discretised domain has $1\,792\,000$ distinct values.

We compare NAPSU-MQ against PGM [27], RAP [2] and PEP [21]. We used the published implementations of their authors for all of them, with small modifications to ensure compatibility with new library versions and our experiments. The published implementation of PrivLCM only supports binary data, and does not scale to datasets of this size, so it was not included in this experiment. We also include a naive noise-aware baseline that runs m completely independent repeats of PGM, splitting the privacy budget appropriately, and uses Rubin's rules with the m generated synthetic datasets. The number of repeats for repeated PGM and the number of generated synthetic datasets for NAPSU-MQ were selected with a preliminary experiment.

As our downstream task, we use logistic regression with income as the dependent variable, and a subset of the columns as independent variables, which allows us to include all the relevant marginals for synthetic data generation. Note that the synthetic dataset was still generated with all 10 columns.

NAPSU-MQ and PGM-repeat sometimes generate synthetic datasets with no people of some race with high income. This causes the downstream logistic regression to produce an extremely wide confidence interval for the coefficient of that race. As Rubin's rules average over estimates and estimated variances, even a single one of these bad estimates makes the combined confidence interval extremely wide. To fix this issue, we can either remove the estimates with estimated standard deviations greater than $10^{1.5}$ before applying Rubin's rules, or add a small regularisation term to the logistic regression, and use bootstrapping for variance estimates. Both methods produce almost identical results, but bootstrapping significantly increases the computational cost of the downstream analysis.

As the input queries, we pick 2-way marginals that are relevant for the downstream task, and select the rest of the queries with the MST algorithm [25]. This selection was kept constant throughout the experiment. For the privacy budget, we use $\delta = n^{-2}$ for all runs, and vary ϵ.

Reiter [31] discusses the choice of n_{Syn} and n for non-DP synthetic data generation in detail. Based on his results, choosing $n_{Syn} = n$, is very safe, and we use it for all algorithm except RAP, where n_{Syn} is a significant hyperparameter. For NAPSU-MQ and PGM-repeat, we choose the number of generated synthetic datasets with a preliminary experiment.

The Laplace approximation for NAPSU-MQ does not work well for this setting because many of the queries have small values, so we use NUTS [15] for posterior inference. To speed up NUTS, we normalise the posterior before running the inference using the mean and covariance of the Laplace approximation

Results from 20 repeats of the experiment are shown in Figure 4. PGM, RAP and PEP produce overconfident confidence intervals that do not meet the given confidence levels. The bottom plots show the cause of the overconfidence: the confidence intervals have similar widths in the original data, while they should be wider in synthetic data because of DP noise.

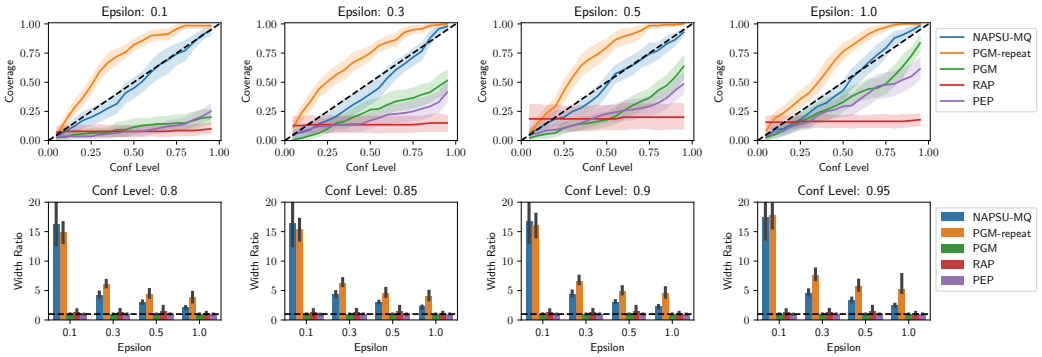


Figure 4: Top row: the fraction of downstream coefficients where the synthetic confidence interval contains the real data coefficient, averaged over 20 repeated runs. NAPSU-MQ is the only algorithm that is consistently around the diagonal, showing good calibration. The error bands are bootstrap 95% confidence intervals of the average. Bottom row: confidence intervals widths divided by real data confidence interval widths. Each bar is a median of medians over the different coefficients and repeats, and the black lines are 95% bootstrapped confidence intervals. The dashed line is at $y = 1$, showing where synthetic confidence intervals have the same width as original confidence intervals.

With the repeats, PGM becomes overly conservative, and produces confidence intervals that are too wide. NAPSU-MQ is the only algorithm that produces properly calibrated confidence intervals, although repeated PGM is able to produce narrower intervals than NAPSU-MQ with $\epsilon = 0.1$.

Noise-awareness, especially with the increased accuracy from NUTS, comes with a steep computational cost, as PGM ran in 15s, while the Laplace approximation took several minutes, and NUTS took up to ten hours. All of the algorithms were run on 4 CPU cores of a cluster.

6 Limitations and Conclusion

Limitations While our general pipeline NA+MI is applicable to all kinds of datasets in principle, the data generation algorithm NAPSU-MQ is currently only applicable to discrete tabular data and only supports sparse marginal queries perturbed with the Gaussian mechanism as input. We aim to generalise NAPSU-MQ to more general query classes, such as linear queries, in the future, but supporting other types of noise is likely much harder.

The runtime of NAPSU-MQ, especially when using NUTS, is another major limitation. As NAPSU-MQ is compatible with any non-DP posterior sampling method, recent [16] and future advances in MCMC and other sampling techniques are likely able to cut down on the runtime.

Conclusion The analysis of DP synthetic data has not received much attention in existing research. Our work patches a major hole in the current generation and analysis methods by developing the NA+MI pipeline that allows computing accurate confidence intervals and p-values from DP synthetic data. We develop the NAPSU-MQ algorithm in order to implement NA+MI on nontrivial discrete datasets. While NAPSU-MQ has several limitations, NA+MI only depends on noise-aware posterior inference, not NAPSU-MQ specifically, and can thus be extended to other settings in the future. With the noise-aware inference algorithm, NA+MI allows conducting valid statistical analyses that include uncertainty estimates with DP synthetic data, potentially unlocking existing privacy-sensitive datasets for widespread analysis.

References

[1] P. Abbeel, D. Koller, and A. Y. Ng. “Learning Factor Graphs in Polynomial Time and Sample Complexity”. In: *Journal of Machine Learning Research* 7 (2006), pp. 1743–1788.

[2] S. Aydore, W. Brown, M. Kearns, K. Kenthapadi, L. Melis, A. Roth, and A. A. Siva. "Differentially Private Query Release Through Adaptive Projection". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 457–467.

[3] B. Balle and Y.-X. Wang. "Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 394–403.

[4] G. Bernstein, R. McKenna, T. Sun, D. Sheldon, M. Hay, and G. Miklau. "Differentially Private Learning of Undirected Graphical Models Using Collective Graphical Models". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 478–487.

[5] G. Bernstein and D. R. Sheldon. "Differentially Private Bayesian Inference for Exponential Families". In: *Advances in Neural Information Processing Systems*. 31, 2018, pp. 2924–2934.

[6] K. Cai, X. Lei, J. Wei, and X. Xiao. "Data Synthesis via Differentially Private Markov Random Fields". In: *Proceedings of the VLDB Endowment* 14.11 (2021), pp. 2190–2202.

[7] D. Chen, T. Orekondy, and M. Fritz. "GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 12673–12684.

[8] R. Chen, Q. Xiao, Y. Zhang, and J. Xu. "Differentially Private High-Dimensional Data Publication via Sampling-Based Inference". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 129–138.

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. "Our Data, Ourselves: Privacy Via Distributed Noise Generation". In: *Advances in Cryptology - EUROCRYPT*. Vol. 4004. Lecture Notes in Computer Science. Springer, 2006, pp. 486–503.

[10] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Third Theory of Cryptography Conference*. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 265–284.

[11] C. Dwork and A. Roth. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014), pp. 211–407.

[12] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Third edition. Chapman & Hall/CRC Texts in Statistical Science Series. Boca Raton: CRC Press, 2014.

[13] F. Harder, K. Adamczewski, and M. Park. "DP-MERF: Differentially Private Mean Embeddings with RandomFeatures for Practical Privacy-preserving Data Generation". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, 2021, pp. 1819–1827.

[14] M. Hardt, K. Ligett, and F. McSherry. "A Simple and Practical Algorithm for Differentially Private Data Release". In: *Advances in Neural Information Processing Systems*. Vol. 25. 2012, pp. 2348–2356.

[15] M. D. Hoffman and A. Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[16] M. Hoffman, A. Radul, and P. Sountsov. "An Adaptive-MCMC Scheme for Setting Trajectory Lengths in Hamiltonian Monte Carlo". In: *The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, 2021, pp. 3907–3915.

[17] J. Jälkö, E. Lagerspetz, J. Haukka, S. Tarkoma, A. Honkela, and S. Kaski. "Privacy-Preserving Data Sharing via Probabilistic Modeling". In: *Patterns* (2021), p. 100271.

[18] R. Kohavi and B. Becker. *Adult*. UCI Machine Learning Repository. 1996.

[19] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[20] S. P. Liew, T. Takahashi, and M. Ueno. "PEARL: Data Synthesis via Private Embeddings and Adversarial Reconstruction Learning". In: *International Conference on Learning Representations*. 2022.

[21] T. Liu, G. Vietri, and S. Wu. "Iterative Methods for Private Synthetic Data: Unifying Framework and New Methods". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[22] Y. Long, B. Wang, Z. Yang, B. Kailkhura, A. Zhang, C. A. Gunter, and B. Li. "G-PATE: Scalable Differentially Private Data Generator via Private Aggregation of Teacher Discriminators". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[23] A. Marshall, D. G. Altman, R. L. Holder, and P. Royston. "Combining Estimates of Interest in Prognostic Modelling Studies after Multiple Imputation: Current Practice and Guidelines". In: *BMC Medical Research Methodology* 9.1 (2009), p. 57.

[24] R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala. "Optimizing Error of High-Dimensional Statistical Queries under Differential Privacy". In: *Proceedings of the VLDB Endowment* 11.10 (2018).

[25] R. McKenna, G. Miklau, and D. Sheldon. "Winning the NIST Contest: A Scalable and General Approach to Differentially Private Synthetic Data". In: *Journal of Privacy and Confidentiality* 11.3 (2021).

[26] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau. *AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data*. arXiv: 2201.12677. 2022.

[27] R. McKenna, D. Sheldon, and G. Miklau. "Graphical-Model Based Estimation and Inference for Differential Privacy". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4435–4444.

[28] C. Meek, B. Thiesson, and D. Heckerman. *US Census Data (1990)*. UCI Machine Learning Repository.

[29] M. P. Nixon, A. F. Barrientos, J. P. Reiter, and A. Slavković. *A Latent Class Modeling Approach for Generating Synthetic Data and Making Posterior Inferences from Differentially Private Counts*. arXiv: 2201.10545. 2022.

[30] T. E. Raghunathan, J. P. Reiter, and D. B. Rubin. "Multiple Imputation for Statistical Disclosure Limitation". In: *Journal of Official Statistics* 19.1 (2003), p. 1.

[31] J. P. Reiter. "Satisfying Disclosure Restrictions with Synthetic Data Sets". In: *Journal of Official Statistics* 18.4 (2002), p. 531.

[32] J. P. Reiter. "Significance Tests for Multi-Component Estimands from Multiply Imputed, Synthetic Microdata". In: *Journal of Statistical Planning and Inference* 131.2 (2005), pp. 365–377.

[33] D. B. Rubin. "Discussion: Statistical Disclosure Limitation". In: *Journal of Official Statistics* 9.2 (1993), pp. 461–468.

[34] Y. Si and J. P. Reiter. "A Comparison of Posterior Simulation and Inference by Combining Rules for Multiple Imputation". In: *Journal of Statistical Theory and Practice* 5.2 (2011), pp. 335–347.

[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*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 9765–9774.

[36] M. J. Wainwright and M. I. Jordan. "Graphical Models, Exponential Families, and Variational Inference". In: *Foundations and Trends in Machine Learning* 1.1-2 (2008), pp. 1–305.

[37] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. *Differentially Private Generative Adversarial Network*. arXiv:1802.06739. 2018.

[38] 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.

[39] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. "PrivBayes: Private Data Release via Bayesian Networks". In: *ACM Transactions on Database Systems* 42.4 (2017), 25:1–25:41.

[40] X. Zhou and J. P. Reiter. "A Note on Bayesian Inference After Multiple Imputation". In: *The American Statistician* 64.2 (2010), pp. 159–163.