UNDERSTANDING THE IMPACT OF DATA DOMAIN EXTRACTION ON SYNTHETIC DATA PRIVACY

Georgi Ganev[1,2] **Meenatchi Sundaram Muthu Selva Annamalai**[1]
Sofiane Mahiou[2] **Emiliano De Cristofaro**[3]
[1]UCL [2]SAS [3]UC Riverside
georgi.ganev.16@ucl.ac.uk

ABSTRACT

Privacy attacks, particularly membership inference attacks (MIAs), are widely used to assess the privacy of generative models for tabular synthetic data, including those with Differential Privacy (DP) guarantees. These attacks often exploit outliers, which are especially vulnerable due to their position at the boundaries of the data domain (e.g., at the minimum and maximum values). However, the role of data domain extraction in generative models and its impact on privacy attacks have been overlooked. In this paper, we examine three strategies for defining the data domain: assuming it is externally provided (ideally from public data), extracting it directly from the input data, and extracting it with DP mechanisms. While common in popular implementations and libraries, we show that the second approach breaks end-to-end DP guarantees and leaves models vulnerable. While using a provided domain (if representative) is preferable, extracting it with DP can also defend against popular MIAs, even at high privacy budgets.

1 INTRODUCTION

Differentially Private (DP) synthetic tabular data promises to support the safe release of sensitive data by training generative machine learning models while limiting individual-level information leakage. This approach is gaining significant traction (Jordon et al., 2022; Hu et al., 2024; De Cristofaro, 2024) and is increasingly being deployed in real-world applications, from public releases of census data (NASEM, 2020; ONS, 2023; Hod & Canetti, 2024) to data sharing in financial and healthcare contexts (UK ICO, 2023b; Microsoft, 2022). Synthetic data has also attracted interest from regulators (UK ICO, 2023a;b; FCA, 2024), who are shifting focus from assessing the anonymity of released datasets (A29WP, 2014) to evaluating generative models (EDPB, 2024).

In this context, membership inference attacks (MIAs) (Shokri et al., 2017; Hayes et al., 2019), are used as a measuring stick for privacy leakage. MIAs are typically evaluated using a privacy game that entails identifying (or crafting) a vulnerable record, training a generative model with it and without it, generating synthetic data, and having the adversary distinguish whether or not that record was used to train the model. In this game, outliers located at the boundaries of the data domain (e.g., at each column's min or max values) are particularly vulnerable (Stadler et al., 2022; Annamalai et al., 2024). However, adding/removing outliers can significantly impact the training of DP generative models, especially the initial pre-processing steps (e.g., scaling, normalization, discretization, encoding, etc.) common in DP synthetic tabular data algorithms. This presents a unique challenge for tabular data, unlike, e.g., for images or text, where input pixels and tokens have clearly defined domains (e.g., $[0, 255]$ or ASCII characters).

Nonetheless, many implementations and libraries for DP synthetic tabular data have overlooked this issue, as they directly extract data domain from the input data (Zhang et al., 2017; Ping et al., 2017; Vietri et al., 2020; McKenna et al., 2021; Qian et al., 2023; Mahiou et al., 2022; Du & Li, 2024). In this paper, we examine how different strategies for extracting the data domain affect the privacy of DP generative models for tabular synthetic data. Specifically, we compare a publicly available data domain to extracting the domain directly from the input data—denoted, respectively, as *provided* and *extracted* domain. We do so both with and without DP and for two generative models, PrivBayes (Zhang et al., 2017) and MST (McKenna et al., 2021). Since both models require discretized data, we adapt and assess four DP discretization strategies: uniform, quantile, k-means, and PrivTree (Zhang et al., 2016). For the MIA, we use the GroundHog attack (Stadler et al., 2022).

In short, our experiments show that:

- Extracting the data domain directly from the input data, which is the common practice, breaks the end-to-end DP guarantees of generative models and exposes outliers to MIAs.
- Assuming that a representative data domain is provided and extracting it with DP (up to $\epsilon = 100$) successfully protects outliers from specific MIAs. In particular, adopting a DP domain extraction strategy could address many previously identified DP vulnerabilities in open-source implementations and libraries.
- The GroundHog attack (Stadler et al., 2022) may be more effective at detecting issues with data domain extraction than with vulnerabilities of the generative models themselves.

From Domain Extraction to Discretization. In separate work (Ganev et al., 2025b), we examine the broader question of discretization in end-to-end DP generative models, primarily focusing on utility. In contrast, this paper focuses specifically on data domain extraction strategies and their privacy implications, which is closely related to Research Question 4 in (Ganev et al., 2025b).

2 Experimental Framework

As mentioned, we aim to evaluate the impact of the domain extraction strategy on privacy leakage in (end-to-end) DP generative models using an MIA. We experiment with three strategies: 1) assuming a provided data domain set to the full dataset's range, regardless of the target record's inclusion/exclusion, 2) extracting it directly from the input data (without DP), as done in numerous publicly available implementations and libraries (Zhang et al., 2017; Ping et al., 2017; Vietri et al., 2020; McKenna et al., 2021; Qian et al., 2023; Mahiou et al., 2022; Du & Li, 2024), or 3) extracting the data domain with DP (Desfontaines, 2020).

MIA Instantiation. To evaluate the privacy of the resulting synthetic data, we use GroundHog (Stadler et al., 2022), one of the most widely used MIAs for synthetic tabular data, on the Wine dataset (Dua & Graff, 2017). First, we select a vulnerable record as the target, picking the data point furthest from all others in the training set (Meeus et al., 2023) and ensuring it lies outside their domain. Then, we train two sets of 200 shadow models: one trained on the full dataset, including the target record, and the other excluding it. To do so, for each model, we extract the domain, discretize the data, and train the generative model. We generate synthetic datasets and extract statistical features (minimum, maximum, mean, median, and standard deviation, corresponding to the naive feature set F_{naive} in (Stadler et al., 2022)) from each column of the synthetic datasets. Half of these datasets are used to train a classifier, and the adversary's success in distinguishing between the two scenarios is measured using Area Under the Curve (AUC), reported on the remaining data.

Settings. We choose PrivBayes (Zhang et al., 2017) and MST (McKenna et al., 2021) as our DP generative models, using $\epsilon = 1$ for pre-processing (split evenly between domain extraction, when applicable, and discretization) and $\epsilon = 1$ for the model (with $\delta = 1\text{e}{-5}$ for MST). We use 20 bins for all discretization strategies and the default hyperparameters for both models. The selected target record represents a worst-case scenario, consistent with prior work (Stadler et al., 2022; Annamalai et al., 2024), given two columns with values significantly larger than for the remaining records (289 and 440 vs. 146.5 and 366.5). Due to space limitations, we defer additional details, including the DP data extraction method, discretization strategies, DP generative models, and dataset, to Appendix A.

3 Experimental Evaluation

Figure 1 provides an overview of our experiments, quantifying the impact of the three domain extraction strategies on privacy leakage as measured by GroundHog (Stadler et al., 2022)'s success with four discretizers and two generative models.

Direct Domain Extraction. Regardless of discretization, extracting the domain directly from the input data (bars with horizontal lines) without a proper privacy mechanism breaks the end-to-end DP guarantees, providing highly informative features. This enables the adversary to achieve near-perfect accuracy in all cases, rendering the synthetic data non-private regardless of the discretizer or generative model. This performance is equivalent to that of using the default domain extraction/discretization strategy (grey bars), i.e., uniform discretization with direct domain extraction.

Provided Domain/DP Domain Extraction. By contrast, using methods that respect the end-to-end DP pipeline, i.e., either assuming a provided domain (bars with crosses) or extracting the domain

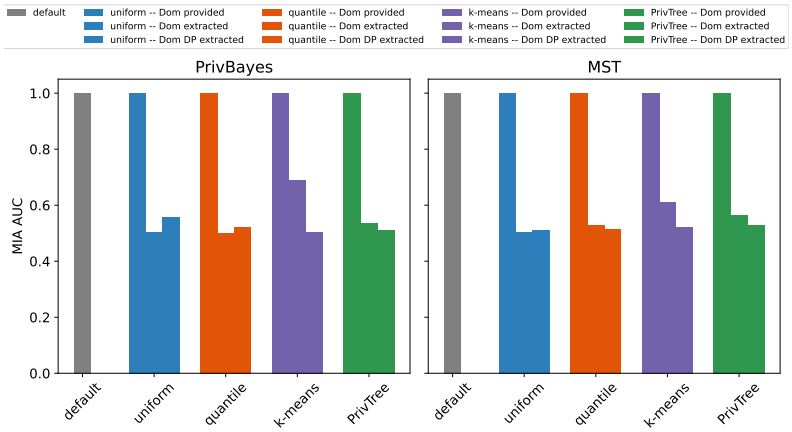


Figure 1: Privacy leakage with *provided* and *extracted* domain (w/ and w/o DP) for the four DP discretizers ($\epsilon = 1$) and two DP generative models ($\epsilon = 1$) on a target record *outside* the domain of the remaining data.

with DP (bars with circles), substantially reduces the adversary's success rate. With only one exception (when using the k-means discretizer with provided domain), the attack success rate is no better than random guessing. This has several important implications, as discussed next.

First, extracting the domain in a DP-compliant way is sufficient to protect against GroundHog (Stadler et al., 2022)'s adversary. Their success drops significantly even in settings that could be considered non-private, i.e., (discretizer, generator)-ϵ values of (1, 100), (100, 1), (100, 100), and even (1, 1,000), as shown in Figure 2 and 3a (see Appendix B). The attack becomes effective at higher discretizer ϵ values, i.e., (1,000, 1) and (1,000, 1,000) (see Figure 3b and 3c); also, recall that it achieves nearly 100% success when the domain is directly extracted from the data. This suggests that the effectiveness of the GroundHog attack may primarily be due to domain extraction rather than inherent vulnerabilities in the model.[1] To further validate this, we run additional experiments on a target record that is farther away from the others but still within their domain (see Figure 4 in Appendix B) and observe that the attack's success remains close to random across all ϵ values.

Second, adopting a DP domain extraction strategy could help address privacy vulnerabilities identified by prior research (Annamalai et al., 2024; Ganev et al., 2025a) in popular model implementations and libraries (Ping et al., 2017; Qian et al., 2023) that directly extract the domain from the input data. In other words, incorporating such techniques could make DP generative model implementations more robust and better align them with end-to-end DP guarantees.

Finally, while extracting the domain in a DP way slightly reduces, on average, the adversary's success compared to using a provided data domain, this may come at the cost of utility. Therefore, practitioners should prefer using a trusted, provided data domain when available (e.g., codebooks for census data), rather than spending additional privacy budget to extract it. However, further research is needed to explore these trade-offs and investigate enhanced methods for DP domain extraction.

4 Conclusion

This paper focused on an important yet overlooked issue in implementations of DP generative models: how to extract data domain We show that extracting the data domain directly from the input, which is unfortunately common in the wild (Ping et al., 2017; Qian et al., 2023), breaks DP guarantees and leaves models vulnerable. We also find that, while using a provided domain (e.g., from public data) is preferable, extracting it with DP can also defend against MIAs, even at large ϵ values.

Overall, we are confident that our research will shed light on the importance of the integrity of end-to-end DP pipelines when developing and releasing DP generative models. Our work also highlights the need for further analysis of membership inference attacks against DP generative models, e.g., understanding the extent to which the privacy leakage they exploit may be due to issues like domain extraction rather than inherent vulnerabilities in the models.

[1] These results are specific to GroundHog (Stadler et al., 2022). Studying the impact of DP domain extraction on other MIAs as well as other models besides PrivBayes and MST, is left to future work.

REFERENCES

A29WP. Opinion on anonymisation techniques. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf, 2014.

Meenatchi Sundaram Muthu Selva Annamalai, Georgi Ganev, and Emiliano De Cristofaro. "What do you want from theory alone?" Experimenting with Tight Auditing of Differentially Private Synthetic Data Generation. In *USENIX Security*, 2024.

Emiliano De Cristofaro. Synthetic Data: Methods, Use Cases, and Risks. *IEEE S&P Magazine*, 2024.

Damien Desfontaines. *Lowering the cost of anonymization*. PhD thesis, ETH Zurich, 2020.

Yuntao Du and Ninghui Li. Towards Principled Assessment of Tabular Data Synthesis Algorithms. *arXiv:2402.06806*, 2024.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/186/wine+quality, 2017.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EuroCrypt*, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006b.

EDPB. Opinion 28/2024 on certain data protection aspects related to the processing of personal data in the context of AI models. https://www.edpb.europa.eu/system/files/2024-12/edpb_opinion_202428_ai-models_en.pdf, 2024.

FCA. Using Synthetic Data in Financial Services. https://www.fca.org.uk/publication/corporate/report-using-synthetic-data-in-financial-services.pdf, 2024.

Georgi Ganev, Meenatchi Sundaram Muthu Selva Annamalai, and Emiliano De Cristofaro. The Elusive Pursuit of Reproducing PATE-GAN: Benchmarking, Auditing, Debugging. *TMLR*, 2025a.

Georgi Ganev, Meenatchi Sundaram Muthu Selva Annamalai, Sofiane Mahiou, and Emiliano De Cristofaro. The Importance of Being Discrete: Measuring the Impact of Discretization in End-to-End Differentially Private Synthetic Data. *arXiv:2504.06923*, 2025b.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *STOC*, 2009.

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Membership Inference Attacks against Generative Models. In *PoPETs*, 2019.

Shlomi Hod and Ran Canetti. Differentially Private Release of Israel's National Registry of Live Births. *arXiv:2405.00267*, 2024.

Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the IBM differential privacy library. https://github.com/IBM/differential-privacy-library, 2019.

Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David Forsyth, Bo Li, and Dawn Song. SoK: Privacy-Preserving Data Synthesis. In *IEEE S&P*, 2024.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic Data–what, why and how? *arXiv:2205.03257*, 2022.

Sofiane Mahiou, Kai Xu, and Georgi Ganev. dpart: Differentially Private Autoregressive Tabular, a General Framework for Synthetic Data Generation. *TPDP*, 2022.

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

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. *JPC*, 2021.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, 2007.

Matthieu Meeus, Florent Guepin, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Achilles' Heels: vulnerable record identification in synthetic data publishing. *arXiv:2306.10308*, 2023.

Microsoft. IOM and Microsoft release first-ever differentially private synthetic dataset to counter human trafficking. https://www.microsoft.com/en-us/research/blog/iom-and-microsoft-release-first-ever-differentially-private-synthetic-dataset-to-counter-human-trafficking/, 2022.

NASEM. *2020 Census Data Products: Data Needs and Privacy Considerations: Proceedings of a Workshop*. The National Academies Press, 2020.

ONS. Synthesising the linked 2011 Census and deaths dataset while preserving its confidentiality. https://datasciencecampus.ons.gov.uk/synthesising-the-linked-2011-census-and-deaths-dataset-while-preserving-its-confidentiality/, 2023.

OpenDP. SmartNoise SDK: Tools for Differential Privacy on Tabular Data. https://github.com/opendp/smartnoise-sdk, 2021.

Haoyue Ping, Julia Stoyanovich, and Bill Howe. DataSynthesizer: Privacy-Preserving Synthetic Datasets. In *SSDBM*, 2017.

Zhaozhi Qian, Rob Davis, and Mihaela Van Der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. In *NeurIPS Datasets and Benchmarks Track*, 2023.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models. In *IEEE S&P*, 2017.

Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *STOC*, 2011.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic Data – Anonymization Groundhog Day. In *USENIX Security*, 2022.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *CODASPY*, 2016.

UK ICO. Privacy-enhancing technologies (PETs). https://ico.org.uk/media/for-organisations/uk-gdpr-guidance-and-resources/data-sharing/privacy-enhancing-technologies-1-0.pdf, 2023a.

UK ICO. Synthetic data to test the effectiveness of a vulnerable person's detection system in financial services. https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-sharing/privacy-enhancing-technologies/case-studies/synthetic-data-to-test-the-effectiveness-of-a-vulnerable-persons-detection-system-in-financial-services/, 2023b.

Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Steven Wu. New oracle-efficient algorithms for private synthetic data release. In *ICML*, 2020.

Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *SIGMOD*, 2016.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM TODS*, 2017.

A More Details on the Experimental Framework

We now provide details of the experimental framework introduced in Section 2, specifically, DP data extraction methods, DP discretization strategies, DP generative models, and dataset.

DP Domain Extraction. To estimate the domain of numerical data given a privacy budget, ϵ, we implement the algorithm by Desfontaines (2020), also included in the popular OpenDP library (OpenDP, 2021). It derives bounds using a noisy histogram over an exponential range $[-2^m, 2^m]$ (with m typically set to 32), determined by iteratively reducing a threshold until at least one bin exceeds it, using the highest and lowest bin edges above the threshold as the domain bounds.

DP Discretizers. We use the following four methods to make the four discretizers satisfy DP (note that the data domain, privacy budget, and number of bins b are provided as input to all discretizers). For the DP implementation, we use primitives of two well-known open-source libraries, namely, Harvard's OpenDP (OpenDP, 2021) and IBM's Diffprivlib (Holohan et al., 2019).

- *Uniform* divides the data domain into b intervals of equal width. It does not consume any privacy budget and relies solely on the provided data domain to determine bin edges.
- *Quantile* distributes data such that each bin contains approximately an equal fraction of data points, specifically $1/b$. The privacy budget ϵ is split evenly across a given number of bins, with each quantile calculated using ϵ/b. We use the method proposed by Smith (2011), which samples quantile values from a discrete distribution. Each q_i is computed as $(x_{i+1} - x_i) \cdot \exp(-\epsilon|i - \alpha n|)$, where x_i is the value at index i in the sorted dataset, α is the target quantile.
- *K-means* employs a standard k-means clustering algorithm to group the data into clusters and then splits them into non-overlapping intervals. It is based on (Su et al., 2016), which adds Geometric noise (Ghosh et al., 2009) to the counts of the nearest neighbors for cluster centers and Laplace to the sum of values per dimension. However, some clusters may occasionally be empty, resulting in fewer than b bins.
- *PrivTree* (Zhang et al., 2016) is a tree-based method that recursively splits the data domain into subdomains. It ensures DP by adding Laplace noise (Dwork et al., 2006b) to the count at each step. Subdomains are further split if the noisy count exceeds a threshold, τ; otherwise, they form leaves, with bin edges corresponding to the domains of all leaves. The threshold parameter τ is set to $1/b$, making b an upper limit for the actual number of bins produced.

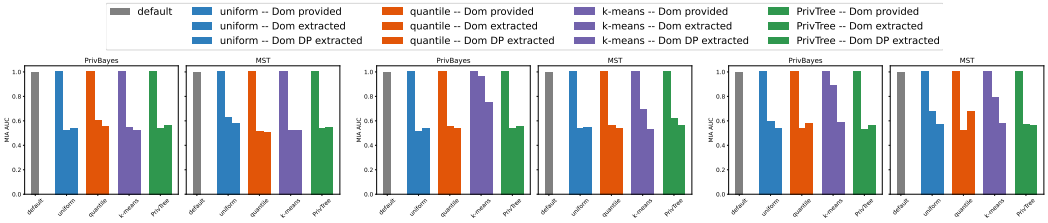
DP Generative Models. The two DP generative models we use, PrivBayes (Zhang et al., 2017) and MST (McKenna et al., 2021), rely on the *select–measure–generate* paradigm (McKenna et al., 2021), as they: 1) select a collection of (low-dimensional) marginals, 2) measure them privately with a noise-addition mechanism, and 3) generate synthetic data consistent with the measurements.

PrivBayes (Zhang et al., 2017) uses a Bayesian network to select k-degree marginals by optimizing the mutual information between them, using the Exponential mechanism (Dwork et al., 2006a). Then, propagating though the network, the model relies on the Laplace mechanism (Dwork et al., 2006b) to measure noisy counts and translate them to conditional marginals, which could later be sampled to generate synthetic data. MST (McKenna et al., 2021) forms a maximum spanning tree (an undirected graph) of the underlying correlation graph by selecting all one-way marginals and a collection of two-way marginals. These marginals are noisily measured via the Gaussian mechanism (McSherry & Talwar, 2007). Finally, to create new data, the measurements are processed through Private-PGM (McKenna et al., 2019).

Wine Dataset (Dua & Graff, 2017). As mentioned, our experiments are run on the Wine dataset, which consists of 4,898 wine samples, each described by 11 continuous physicochemical attributes, with the goal of modeling wine quality.

B Additional Plots

In Figure 2 and 3, we present additional results related to running the GroundHog attack (Stadler et al., 2022) on PrivBayes and MST – specifically, with a target record outside the domain of the remaining data with $\epsilon = 1/100$ and $1/1,000$, respectively. Figure 4 also shows results with a target record inside the domain with $\epsilon = 1,100$ and $1,000$. We discuss these results in Section 3.

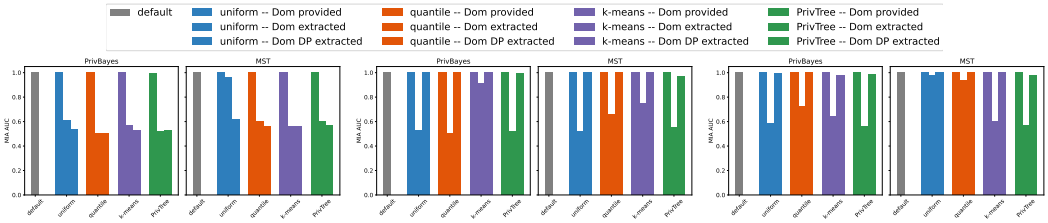


(a) Discretizer ($\epsilon = 1$), Generator ($\epsilon = 100$)

(b) Discretizer ($\epsilon = 100$), Generator ($\epsilon = 1$)

(c) Discretizer ($\epsilon = 100$), Generator ($\epsilon = 100$)

Figure 2: Privacy leakage with *provided* domain and *extracted* domain (w/ and w/o DP) of the four DP discretizers ($\epsilon = 1$ or 100) and two DP generative models ($\epsilon = 1$ or 100) on a target record *outside* the domain of the remaining data.

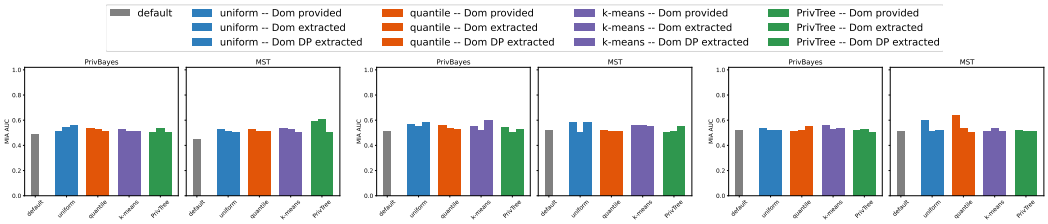


(a) Discretizer ($\epsilon = 1$), Generator ($\epsilon = 1,000$)

(b) Discretizer ($\epsilon = 1,000$), Generator ($\epsilon = 1$)

(c) Discretizer ($\epsilon = 1,000$), Generator ($\epsilon = 1,000$)

Figure 3: Privacy leakage with *provided* domain and *extracted* domain (w/ and w/o DP) of the four DP discretizers ($\epsilon = 1$ or $1,000$) and two DP generative models ($\epsilon = 1$ or $1,000$) on a target record *outside* the domain of the remaining data.



(a) Discretizer ($\epsilon = 1$), Generator ($\epsilon = 1$)

(b) Discretizer ($\epsilon = 100$), Generator ($\epsilon = 100$)

(c) Discretizer ($\epsilon = 1,000$), Generator ($\epsilon = 1,000$)

Figure 4: Privacy leakage with *provided* domain and *extracted* domain (w/ and w/o DP) of the four DP discretizers ($\epsilon = 1,100$ or $1,000$) and two DP generative models ($\epsilon = 1,100$ or $1,000$) on a target record *inside* the domain of the remaining data.