GEN-LRA: TOWARDS A PRINCIPLED MEMBERSHIP INFERENCE ATTACK FOR GENERATIVE MODELS (SUPPLEMENTAL MATERIALS)

Anonymous authors
Paper under double-blind review

A APPENDIX

A.1 ADDITIONAL FIGURES

A.1.1 SAMPLE SIZE AND MIA EFFECTIVENESS

It is known that Membership Inference Attacks benefit from low sample sizes of T, R, and S. We explore the effect of the size of these samples across all models and datasets in figure 1. Here, we see that performance drops off between N=250 and N=1000; however it is relatively the same across all MIAs between N=1000 and N=4000. Across all N-sizes, Gen-LRA has a greater average AUC-ROC then all other MIAs. This further demonstrates that Gen-LRA is an excellent choice for a privacy auditing adversarial attack.

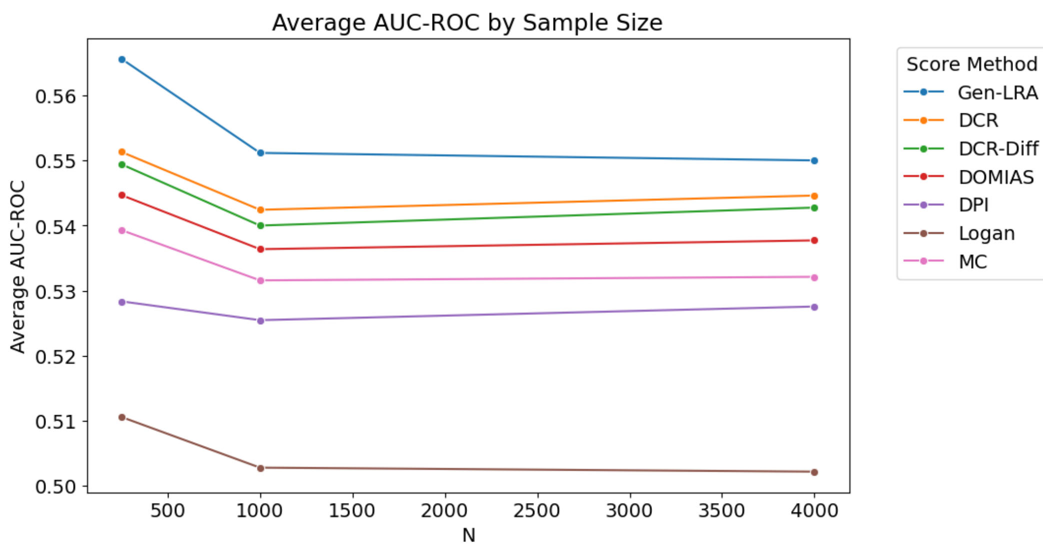

Figure 1: Average MIA AUC-ROC across different sample sizes. There is little decrease in performance after N=1000 and Gen-LRA has the highest global attack performance across N-sizes.

A.1.2 AVERAGE ACCURACY TABLE

A.1.3 MODEL UTILITY AND GEN-LRA EFFECTIVENESS

We benchmark various statistical metrics used to describe the quality of tabular synthetic data across architectures and datasets. We plot the mean Wasserstein distance and Maximum Mean Discrepancy between the corresponding training and synthetic data against the mean AUC-ROC of Gen-LRA in figure 2. Here, it seems there is some relationship between measures of statistical distance and Gen-LRA's global effectiveness. As these metrics are often used in utility benchmarks for tabular synthetic data, it is important to note that for practitioners, statistical fidelity in synthetic data can

Table 1: Average AUC-ROC for each Membership Inference Attack across model architectures and datasets.

Model	Gen-LRA (Ours)	MC	DCR	DCR-Diff	DPI	DOMIAS	LOGAN 2017
AdsGAN	**0.524 (0.02)**	0.513 (0.02)	0.513 (0.02)	0.513 (0.02)	0.515 (0.02)	0.513 (0.02)	0.503 (0.02)
ARF	**0.539 (0.02)**	0.524 (0.02)	0.524 (0.02)	0.529 (0.02)	0.526 (0.02)	0.524 (0.02)	0.503 (0.02)
Bayesian Network	0.619 (0.05)	**0.629 (0.05)**	0.629 (0.05)	0.621 (0.05)	0.538 (0.02)	0.599 (0.05)	0.504 (0.02)
CTGAN	**0.523 (0.02)**	0.509 (0.02)	0.509 (0.02)	0.511 (0.02)	0.513 (0.02)	0.511 (0.02)	0.504 (0.02)
Tab-DDPM	**0.58 (0.04)**	0.564 (0.05)	0.564 (0.05)	0.563 (0.05)	0.537 (0.02)	0.563 (0.04)	0.504 (0.02)
Normalizing Flows	**0.517 (0.02)**	0.504 (0.02)	0.504 (0.02)	0.504 (0.02)	0.505 (0.02)	0.504 (0.02)	0.501 (0.02)
PATEGAN	**0.514 (0.02)**	0.501 (0.02)	0.501 (0.02)	0.499 (0.02)	0.499 (0.02)	0.500 (0.02)	0.501 (0.02)
TVAE	**0.533 (0.02)**	0.520 (0.02)	0.520 (0.02)	0.522 (0.02)	0.517 (0.02)	0.518 (0.02)	0.503 (0.02)
Rank	**1.3**	3.2	3.4	3.6	3.6	3.9	5.5

come at a privacy cost. It also illustrates that measures of utility should include some kind of holdout testing method to consider overfitting.

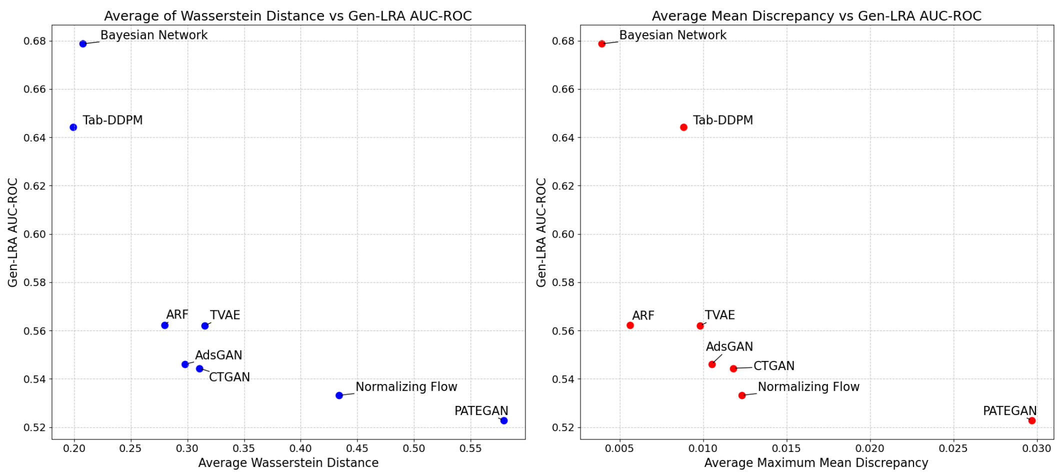

Figure 2: Average Wasserstein Distance and Average Maximum Mean Discrepancy plotted against Gen-LRA AUC-ROC for benchmarked models. Bayesian Network and Tab-DDPM outperform other models in these performance metrics but have higher privacy risk.

A.2 Experiment Details

A.2.1 Section 6.2

We conducted two experiments to evaluate the performance of DCR and Gen-LRA on different types of model failure, with the full results shown in table **??**. The experiments were carried out as follows:

Data Copying Simulation In this setup, we let T and R be random samples from a 2-dimensional standard multivariate Gaussian distribution; i.e., $T, R \overset{\text{iid}}{\sim} \mathcal{N}_2(\mathbf{0}, \mathbf{I})$. Here, we assume a model $\mathcal{M}$ that exactly reproduces the training examples in its output, meaning $S = T$.

Overfitting Simulation In this simulation, we again let $R \overset{\text{iid}}{\sim} \mathcal{N}_2(\mathbf{0}, \mathbf{I})$, but the sampling distribution of T is modified to slightly differ from R, potentially due to sampling variation or bias. In this case, the output S models T well, where $D, S \overset{\text{iid}}{\sim} \mathcal{N}_2(\mathbf{0}, \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix})$.

For both simulations, we set the sample size $n = 500$ for T, R, and S, and the AUC-ROC of DCR and Gen-LRA was compared over 10,000 iterations.

A.3 Ablation: Different k sizes

Gen-LRA targets local fitting by selecting a subset of S to evaluate likelihoods with. This is implemented using the k-nearest neighbors in S to x^*. In practice, this means that k must be selected as a hyperparameter for the attack. In order to understand how k impacts the quality of the attack, we replicate section **??** benchmarking with various k values. We report the average AUC-ROC and standard deviations in table 2. Overall, we find that empirically usually smaller values of k are better although it depends on the model. As stated in section **??**, a global attack over all S is unlikely to yield much membership signal. This is confirmed with $k = N$, where the AUC-ROC is always 0.5 and highlights that overfitting is a local phenomenon and that generative model adversarial attacks should focus on attacking locality to be successful.

Table 2: Average AUC-ROC at different k values for Gen-LRA.

Model	**k=1**	**k=3**	**k=5**	**k=10**	**k=15**	**k=20**	**k=N**
AdsGAN	0.514 (0.02)	0.518 (0.02)	0.519 (0.02)	0.520 (0.02)	0.521 (0.02)	0.521 (0.02)	0.500 (0.00)
ARF	0.532 (0.02)	0.538 (0.02)	0.540 (0.02)	0.540 (0.03)	0.540 (0.03)	0.539 (0.03)	0.500 (0.00)
Bayesian Network	0.650 (0.07)	0.645 (0.07)	0.640 (0.07)	0.634 (0.07)	0.631 (0.07)	0.629 (0.07)	0.500 (0.00)
CTGAN	0.514 (0.02)	0.516 (0.02)	0.517 (0.02)	0.517 (0.02)	0.518 (0.02)	0.518 (0.02)	0.500 (0.00)
Tab-DDPM	0.595 (0.07)	0.595 (0.07)	0.594 (0.07)	0.592 (0.06)	0.591 (0.06)	0.589 (0.06)	0.500 (0.00
Normalizing Flow	0.503 (0.02)	0.503 (0.02)	0.505 (0.02)	0.506 (0.02)	0.506 (0.02)	0.506 (0.02)	0.500 (0.00)
TVAE	0.527 (0.03)	0.531 (0.03)	0.531 (0.03)	0.531 (0.03)	0.530 (0.03)	0.529 (0.03)	0.500 (0.00)

A.4 MIAs for Generative Models Descriptions

The Membership Inference Attacks referenced in this paper is are described as follows:

- **LOGAN** Hayes et al. (2017): LOGAN consists of black box and shadow box attack. The black-box version involves training a Generative Adversarial Network (GAN) on the synthetic dataset and using the discriminator to score test data. A calibrated version improves upon this by training a binary classifier to distinguish between the synthetic and reference dataset. In this paper, we only benchmark the calibrated version.
- **Distance to Closest Record (DCR) / DCR Difference** Chen et al. (2020): DCR is a black-box attack that scores test data based on a sigmoid score of the distance to the nearest neighbor in the synthetic dataset. DCR Difference enhances this approach by incorporating a reference set, subtracting the distance to the closest record in the reference set from the synthetic set distance.
- **MC** Hilprecht et al. (2019): MC is based on counting the number of observations in the synthetic dataset that fall into the neighborhood of a test point (Monte Carlo Integration). However, this method does not consider a reference dataset, and the choice of distance metric for defining a neighborhood is a non-trivial hyperparameter to tune.
- **DOMIAS** van Breugel et al. (2023): DOMIAS is a calibrated attack which scores test data by performing density estimation on both the synthetic and reference datasets. It then calculates the density ratio of the test data between the learned synthetic and reference probability densities.
- **DPI** Ward et al. (2024): DPI computes the ratio of k-Nearest Neighbors of x^* in the synthetic and reference datasets. It then builds a scoring function by computing the ratio of the sum of data points from each class of neighbors from the respective sets.

A.5 Generative Model Architecture Descriptions

In all experiments, we use the implementations of these models from the Python package Synthcity Qian et al. (2023). For benchmarking purposes, we use the default hyperparameters for each model. A brief description of each model is as follows:

- **CTGAN** Xu et al. (2019): Conditional Tabular Generative Adversarial Network uses a GAN framework with conditional generator and discriminator to capture multi-modal distributions. It employs mode normalization to better learn mixed-type distributions.

- **TVAE** Xu et al. (2019): Tabular Variational Auto-Encoder is similar to CTGAN in its use of mode normalizing techniques, but instead of a GAN architecture, it employs a Variational Autoencoder.
- **Normalizing Flows (NFlows)** Durkan et al. (2019): Normalizing flows transform a simple base distribution (e.g., Gaussian) into a more complex one matching the data by applying a sequence of invertible, differentiable mappings.
- **Bayesian Network (BN)** Ankan & Panda (2015): Bayesian Networks use a Directed Acyclic Graph to represent the joint probability distribution over variables as a product of marginal and conditional distributions. It then samples the empirical distributions estimated from the training dataset.
- **Adversarial Random Forests (ARF)** Watson et al. (2023): ARFs extend the random forest model by adding an adversarial stage. Random forests generate synthetic samples which are scored against the real data by a discriminator network. This score is used to re-train the forests iteratively.
- **Tab-DDPM** Kotelnikov et al. (2022): Tabular Denoising Diffusion Probabilistic Model adapts the DDPM framework for image synthesis. It iteratively refines random noise into synthetic data by learning the data distribution through gradients of a classifier on partially corrupted samples with Gaussian noise.
- **PATEGAN** Yoon et al. (2019): The PATEGAN model uses a neural encoder to map discrete tabular data into a continuous latent representation which is sampled from during generation by the GAN discriminator and generator pair.
- **Ads-GAN** Yoon et al. (2020): Ads-GAN uses a GAN architecture for tabular synthesis but also adds an identifiability metric to increase its ability to not mimic training data.

A.6 Benchmarking Datasets References

We provide the URL for the sources of each dataset considered in the paper. We use datasets common in the tabular generative modeling literature Suh et al. (2023)

1. **Abalone** (OpenML): `https://www.openml.org/search?type=data&sort=runs&id=183&status=active`
2. **Adult** Becker & Kohavi (1996)
3. **Bean** (UCI): `https://archive.ics.uci.edu/dataset/602/dry+bean+dataset`
4. **Churn-Modeling** (Kaggle): `https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling`
5. **Faults** (UCI): `https://archive.ics.uci.edu/dataset/198/steel+plates+faults`
6. **HTRU** (UCI): `https://archive.ics.uci.edu/dataset/372/htru2`
7. **Indian Liver Patient** (Kaggle): `https://www.kaggle.com/datasets/uciml/indian-liver-patient-records?resource=download`
8. **Insurance** (Kaggle): `https://www.kaggle.com/datasets/mirichoi0218/insurance`
9. **Magic** (Kaggle): `https://www.kaggle.com/datasets/abhinand05/magic-gamma-telescope-dataset?resource=download`
10. **News** (UCI): `https://archive.ics.uci.edu/dataset/332/online+news+popularity`
11. **Nursery** (Kaggle): `https://www.kaggle.com/datasets/heitornunes/nursery`
12. **Obesity** (Kaggle): `https://www.kaggle.com/datasets/tathagatbanerjee/obesity-dataset-uci-ml`
13. **Shoppers** (Kaggle): `https://www.kaggle.com/datasets/henrysue/online-shoppers-intention`

14. **Titanic** (Kaggle): `https://www.kaggle.com/c/titanic/data`
15. **Wilt** (OpenML): `https://www.openml.org/search?type=data&sort=runs&id=40983&status=active`

REFERENCES

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the Python in Science Conference*, SciPy. SciPy, 2015. doi: 10.25080/majora-7b98e3ed-001. URL `http://dx.doi.org/10.25080/Majora-7b98e3ed-001`.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20. ACM, October 2020. doi: 10.1145/3372297.3417238. URL `http://dx.doi.org/10.1145/3372297.3417238`.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. *Neural spline flows*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019: 133 – 152, 2017. URL `https://api.semanticscholar.org/CorpusID:52211986`.

Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:232 – 249, 2019. URL `https://api.semanticscholar.org/CorpusID:199546273`.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models, 2022.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023. URL `https://arxiv.org/abs/2301.07573`.

Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Merhdad Honarkhah, and Guang Cheng. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing, 2023. URL `https://arxiv.org/abs/2310.15479`.

Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection, 2023.

Joshua Ward, Chi-Hua Wang, and Guang Cheng. Data plagiarism index: Characterizing the privacy risk of data-copying in tabular generative models. *KDD- Generative AI Evaluation Workshop*, 2024. URL `https://arxiv.org/abs/2406.13012`.

David S. Watson, Kristin Blesch, Jan Kapar, and Marvin N. Wright. Adversarial random forests for density estimation and generative modeling. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5357–5375. PMLR, 25–27 Apr 2023. URL `https://proceedings.mlr.press/v206/watson23a.html`.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Neural Information Processing Systems*, 2019. URL `https://api.semanticscholar.org/CorpusID:195767064`.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1zk9iRqF7`.

Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8):2378–2388, 2020.