A Comparative Study of Fidelity Metric Variability in Synthetic Tabular Data Generated With and Without Differential Privacy Guarantees

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Data Analytics
June 2026
Ghufran Ullah

UNIVERSITY OF TURKU
Department of Computing

Ghufran Ullah: A Comparative Study of Fidelity Metric Variability in Synthetic Tabular Data Generated With and Without Differential Privacy Guarantees

Master of Science (Tech) Thesis, 87 p., 7 app. p.
Data Analytics
June 2026

Synthetic tabular data generation is increasingly used when real datasets cannot be shared directly because of privacy, legal, or ethical concerns. However, synthetic data is useful when it preserves important statistical properties of the real data and produces reliable results across repeated generations. This thesis investigates the fidelity metric variability of differentially private (DP) and Non-Differential Private (Non-DP) synthetic tabular data generators. The main objective is to examine how closely synthetic data reproduces the univariate distributions of real data and how much run-to-run dispersion these fidelity measurements show.

The study compares six synthetic data generation methods using three datasets: Adult, Bank Marketing, and Cardiovascular Disease. The Non-DP synthesizers include Gaussian Copula, CTGAN, TVAE, and RTVAE, while the DP synthesizers include AIM and PrivBayes. The experiments are conducted across three sample sizes, $n = 500$, $n = 5000$, and $n = 10000$. For DP methods, three privacy budget values are used: $\epsilon = 1$, $\epsilon = 5$, and $\epsilon = 10$. Hellinger distance is used to capture the geometric difference between probability distributions, while Jensen Shannon distance is used to capture the information theoretic difference between each distribution and their shared midpoint distribution.

The results show that the Gaussian Copula model generally achieves high synthetic data fidelity based on Hellinger and Jensen Shannon distances for categorical variables, especially for the Adult and Bank datasets. CTGAN and TVAE show moderate performance, while RTVAE shows higher run-to-run variability for several complex variables. The Cardio dataset is more difficult for all synthesizers, mainly because of variables with irregular numerical and clinical distributions. Among the differentially private synthesizers, AIM generally outperforms PrivBayes by producing lower distance values and lower-dispersion fidelity scores. Increasing the sample size usually improves stability, especially from $n = 500$ to $n = 5000$. Increasing the privacy budget from $\epsilon = 1$ to $\epsilon = 5$ generally improves fidelity and stability, although the improvement from $\epsilon = 5$ to $\epsilon = 10$ did not produce a uniform improvement across models and variables.

Keywords: Synthetic data, differential privacy, data fidelity, Hellinger distance, Jensen Shannon distance, privacy budget

Contents

1 Introduction **1**

2 Literature Review **7**

2.1 Synthetic Tabular Data Generation 7

2.1.1 Non-DP Data Generators 10

2.1.2 DP Synthetic Data Generators 14

2.1.3 Evaluation of Synthetic Data Quality 16

2.1.4 Fidelity Metrics for Synthetic Data Evaluation 18

3 Methodology **21**

3.1 Experimental Design Overview 21

3.2 Datasets 22

3.3 Data Preprocessing 23

3.4 Experimentation 25

3.5 Evaluation Procedure 27

4 Results and Discussion **30**

4.1 Non-DP Synthesizers 30

4.1.1 Adult Dataset 31

4.1.2 Bank Dataset 35

4.1.3 Cardio Dataset 41

4.2 Discussion on Non-DP Synthesizers 48
4.3 DP Synthesizers 52
4.3.1 Differential Privacy Budget $\epsilon = 1$ 52
4.3.2 Differential Privacy Budget $\epsilon = 5$ 63
4.3.3 Differential Privacy Budget $\epsilon = 10$ 72
4.4 Discussion on DP Synthesizers 77

5 Conclusion **83**
5.1 Answering the Research Questions 83
5.2 Conclusions, Limitations and Future Work 85

6 Disclosure of AI Assistance in the Thesis **87**

References **88**

Appendices

A Complete Non-Private Synthesizers Variable-Level Boxplots **A-1**

List of Figures

4.1 Hellinger distance results for selected variables in the Adult dataset. . 34

4.2 Jensen–Shannon distance results for selected variables in the Adult dataset. 36

4.3 Hellinger distance results for selected variables in the Bank dataset. . 40

4.4 Jensen–Shannon distance results for selected variables in the Bank dataset. 42

4.5 Hellinger distance results for selected variables in the Cardio dataset. 45

4.6 Jensen–Shannon distance results for selected variables in the Cardio dataset. 47

4.7 Hellinger distance results for selected variables in the Adult dataset with $\epsilon = 1$. 54

4.8 Jensen–Shannon distance results for selected variables in the Adult dataset with $\epsilon = 1$. 56

4.9 Hellinger distance results for selected variables in the Bank dataset with $\epsilon = 1$. 58

4.10 Jensen–Shannon distance results for selected variables in the Bank dataset with $\epsilon = 1$. 60

4.11 Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 1$. 62

4.12 Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 1$. 64
4.13 Hellinger distance results for selected variables in the Adult dataset with $\epsilon = 5$. 66
4.14 Jensen–Shannon distance results for selected variables in the Adult dataset with $\epsilon = 5$. 68
4.15 Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 5$. 71
4.16 Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 5$. 73
4.17 Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 10$. 74
4.18 Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 10$. 76

A.1 Hellinger distance results for all variables in the Adult dataset using non-private synthesizers. A-2
A.2 Jensen Shannon distance results for all variables in the Adult dataset using non-private synthesizers. A-3
A.3 Hellinger distance results for all variables in the Bank dataset using non-private synthesizers. A-4
A.4 Jensen–Shannon distance results for all variables in the Bank dataset using non-private synthesizers. A-5
A.5 Hellinger distance results for all variables in the Cardio dataset using non-private synthesizers. A-6
A.6 Jensen Shannon distance results for all variables in the Cardio dataset using non-private synthesizers. A-7

List of Tables

3.1 Version of libraries used in the experiments 26

4.1 Hellinger distance summary for selected Adult variables using CTGAN and Gaussian Copula 31

4.2 Jensen–Shannon distance summary for selected Adult variables using CTGAN and Gaussian Copula 32

4.3 Hellinger distance summary for selected Adult variables using TVAE and RTVAE 33

4.4 Jensen–Shannon distance summary for selected Adult variables using TVAE and RTVAE 33

4.5 Hellinger distance summary for selected Bank variables using CTGAN and Gaussian Copula 37

4.6 Jensen–Shannon distance summary for selected Bank variables using CTGAN and Gaussian Copula 37

4.7 Hellinger distance summary for selected Bank variables using TVAE and RTVAE 38

4.8 Jensen–Shannon distance summary for selected Bank variables using TVAE and RTVAE 39

4.9 Hellinger distance summary for selected Cardio variables using CTGAN and Gaussian Copula 43

4.10 Jensen–Shannon distance summary for selected Cardio variables using CTGAN and Gaussian Copula 43
4.11 Hellinger distance summary for selected Cardio variables using TVAE and RTVAE 44
4.12 Jensen–Shannon distance summary for selected Cardio variables using TVAE and RTVAE 46
4.13 Hellinger distance results for selected Adult with Epsilon 1 53
4.14 Jensen–Shannon distance results for selected Adult with Epsilon 1 . . 53
4.15 Hellinger distance results for selected Bank with Epsilon 1 57
4.16 Jensen–Shannon distance results for selected Bank with Epsilon 1 . . 59
4.17 Hellinger distance results for selected Cardio with Epsilon 1 61
4.18 Jensen–Shannon distance results for selected Cardio with Epsilon 1 . 63
4.19 Hellinger distance results for selected Adult with Epsilon 5 65
4.20 Jensen–Shannon distance results for selected Adult with Epsilon 5 . . 67
4.21 Hellinger distance results for selected Cardio with Epsilon 5 69
4.22 Jensen–Shannon distance results for selected Cardio with Epsilon 5 . 70
4.23 Hellinger distance results for selected Cardio with Epsilon 10 75
4.24 Jensen–Shannon distance results for selected Cardio with Epsilon 10 . 77

1 Introduction

Data has become a central resource in modern research, digital services, and decision-making. In fields such as healthcare, finance, public administration, and business, data is used to identify patterns, develop machine learning models, evaluate the systems, and support evidence-based decision making. However, many real-world datasets contain sensitive information, which includes demographic, medical, financial, or behavioural records. High-dimensional, frequently sparse datasets are intrinsically susceptible to privacy attacks, as presented by Arvind et.al. where they were successfully able to locate known users' Netflix data, revealing their apparent political inclinations as well as other potentially private information [1]. Another study found a formula measuring the uniqueness for human mobility by coarsening the data both physically and temporally, and it revealed that even coarse datasets offer minimal anonymity [2].

The protection offered by current anonymization methods such as k-anonymity [3], l-diversity [4], t-closeness [5] are insufficient. This makes it more difficult for us to share these high variety, high volume, and high velocity datasets, which limits the advancement and application of data science and machine learning techniques. As a result, the use and sharing of such data often involves privacy, ethical, legal, and institutional concerns. This creates an important challenge for researchers and organisations, they need access to useful data, but this access must only be balanced with the responsibility to protect the individuals represented in the data [6], [7].

Synthetic data generation has emerged as one possible response to this challenge. Instead of directly sharing original records, synthetic data aims to generate artificial records that are able to preserve useful statistical patterns from the real data. In case of tabular datasets, this means producing rows and columns that resemble the original structure, distributions, and relationships of the dataset [6]. However, synthetic data is not automatically useful or safe simply just because it is artificially created. It is not intrinsically private to train an off-the-shelf generative algorithm on real data and then use the trained model to produce synthetic data. Private or objective data is not produced by standard GANs [8]. Its quality must be evaluated in terms of how closely it represents the real data, whether it remains useful for analysis, and whether it reduces privacy risks [7]. Therefore, the evaluation of synthetic tabular data has become an important area of study, especially when such data is considered for privacy-aware data sharing and analysis.

Synthetic tabular data generation has been studied as a way to support data access when real datasets cannot be shared directly. The main idea behind it is to create artificial records that preserve useful statistical patterns from the original data, while reducing the need to expose real individual-level records [6], [9]. In case of tabular data, this task is not very simple because datasets often contain a mixture of categorical, numerical, binary, ordinal, and imbalanced variables. Therefore, in any synthetic data generation method, a model must not only be able to keep the individual variable distributions, but also the structure and relationships with other variables within the dataset.

Several methods have been proposed for generating synthetic tabular data. Statistical approaches, such as the Synthetic Data Vault framework and Gaussian Copula-based models, attempt to learn the distributional structure of tabular data and generate new records from the estimated model [9]. Classifiers like random forests, support vector machines, and inverted decision trees were used by early

generators. Nevertheless, these techniques found it difficult to find a compromise between the risk of data leakage and classifier accuracy [10], [11]. More recent deep learning-based methods, such as CTGAN and TVAE, were developed to better handle the challenges of mixed-type tabular data, including multimodal continuous variables and imbalanced categorical variables [12]. Using GANs necessitated transforming discrete and categorical columns into a continuous form, either by breaking them down into a variable number of Gaussian modes or using auto-encoders [12], [13]. These methods have expanded the practical use of synthetic data, but their performance can differ depending on dataset structure, sample size, and variable complexity.

Differential privacy has also become an important direction in synthetic data generation because it provides a formal privacy framework for limiting the influence of any single individual in the data [14].The ability to integrate private synthetic data into the processing of data workflow intended for the real data makes it attractive. In DP methods such as Private Data Release via Bayesian Networks (PrivBayes) and Adaptive and Iterative Mechanism (AIM), they generate synthetic records while adding privacy protection during the modelling process. A key parameter in this setting is the privacy budget, epsilon, where smaller values usually provide stronger privacy protection but may reduce the accuracy of the generated data. The precision that synthetic data may offer under differential privacy, or in any other plausible definition of privacy, is known to have limitations [15].

The evaluation of synthetic data is commonly discussed through three connected dimensions: fidelity, utility, and privacy [7], [16], [17]. Fidelity measures how closely the synthetic data follows the statistical properties of the real data, while utility examines whether the synthetic data remains useful for analysis or machine learning tasks. Privacy focuses on the risk that sensitive information from real individuals may be exposed. In this thesis, the main focus is on fidelity in univariate distributional

evaluation. In this thesis, run-to-run stability refers to low statistical dispersion of fidelity scores across repeated synthetic data generations under the same experimental setting. It is measured using standard deviation, interquartile range, and boxplot spread. The term is not used in the sense of algorithmic stability or differential privacy sensitivity. Therefore, variability is not treated as a separate evaluation concept, but as the statistical dispersion of fidelity scores when the same synthetic data generation process is repeated under fixed experimental settings.

Although synthetic tabular data generation has received growing attention, existing studies often focus on whether a generated dataset is close to the real dataset in terms of fidelity, utility, or privacy risk [6], [7]. Many studies compare different synthesizers using distributional similarity measures, machine learning utility, or privacy-related evaluations. However, less attention is given to how much run-to-run dispersion these fidelity measurements show when the same synthesizer is run repeatedly under the same experimental setting. This is important because many synthetic data generation methods are stochastic, meaning that repeated runs may produce different synthetic datasets and therefore different fidelity scores.

This creates a practical research gap. A synthesizer may achieve a low average distance from the real data, but if its fidelity scores vary greatly across repeated runs, the method may be less reliable in practice. In contrast, a method with slightly higher average distance but lower run-to-run variation may produce more reproducible results. Therefore, it is not enough to only report a single fidelity score or compare models based on one generated dataset. The variation of fidelity scores across repeated runs should also be examined to understand whether the synthesizer produces stable and reproducible outcomes.

This study addresses this gap by evaluating the run-to-run stability of fidelity metrics in Differentially Private (DP) and Non-Differentially Private (Non-DP) synthetic data generation. The distance metrics used in this thesis, including Hellinger

distance, and Jensen Shannon distance which are treated as fidelity measures. Their variation across repeated runs is then used to assess the stability of the synthesizers. By comparing multiple synthesizers across different datasets, sample sizes, and privacy budget values, this thesis provides a more detailed view of how model type, dataset structure, sample size, and differential privacy affect the run-to-run dispersion of synthetic data fidelity.

The aim of this thesis is to compare the univariate fidelity of DP and Non-DP synthetic tabular data generation methods and to examine how stable these fidelity measurements are across repeated runs.

The study uses Hellinger distance which is measuring the overlap between the probability distributions of each variables, while Jensen–Shannon distance is measuring their information theoretic divergence in a symmetric and bounded form. The variation of these scores across repeated runs is used to assess run-to-run stability measured through variability. Based on this, the thesis addresses the following research questions: 1. how much statistical dispersion is observed in the fidelity scores of synthetic data generation methods across repeated runs, and how do variable types affect the observed fidelity scores?; 2. how do DP and Non-DP synthesizers differ in terms of fidelity and run-to-run stability stability?; 3. how the privacy budget affect the fidelity and run-to-run stability of DP synthesizers? and 4. how does increasing sample size influence the consistency of fidelity measurements?

This thesis contributes to the evaluation of synthetic tabular data by examining both the fidelity and run-to-run stability of generated data across repeated runs. Using both fidelity metrics, the study evaluates how different synthesizers are able to reproduce data quality under fixed experimental settings. It also compares the DP and Non-DP synthetic data generation methods across three datasets, three sample sizes, and three different privacy budgets, including statistical, deep learning-based, and DP approaches. By analyzing the effects, the study provides insights that can

help the researchers in future, while selecting synthetic data generation methods that balance data quality, privacy, and reproducibility.

This thesis consists of five chapters, followed by a Use of AI statement. Chapter 1 introduces the research problem, motivation, research gap, objectives, research questions, and contributions of the study. Chapter 2 provides the theoretical background and literature review, covering synthetic tabular data generation, statistical and deep learning-based synthesizers, differential privacy concepts, and the evaluation dimensions of fidelity, utility, privacy, and stability. Chapter 3 describes the research methodology, which incudes the datasets used, preprocessing procedures, experimental setup, privacy budget configurations, sample sizes, and evaluation metrics. The chapter also explains the repeated-run experimental design used to evaluate the stability of synthetic data generation.

Chapter 4 presents and discusses the experimental results for both Non-DP and DP synthesizers, analysing the effects of model type, dataset characteristics, sample size, variable structure, and privacy budget on fidelity and stability. Finally, Chapter 5 summarises the key findings, answers the research questions, highlights the contributions and limitations of the study, and provides recommendations for future research. Additional experimental details and supplementary results are included in the appendices.

2 Literature Review

The literature on synthetic tabular data generation mainly focuses on three connected goals: preserving data fidelity, maintaining practical utility, and reducing privacy risk. A model that produces synthetic data with strong similarity to the original data may also increase privacy risk if it memorises individual records. On the other hand, a model that adds strong privacy protection may reduce the statistical quality of the generated data. In this chapter we are going to discuss existing approaches for generating synthetic data.

2.1 Synthetic Tabular Data Generation

Synthetic data generation can be defined as the process of creating artificial data that is designed to preserve useful statistical patterns from an original dataset [6]. Instead of directly sharing real records, a synthetic data generator learns patterns from the real data and produces new records that follow similar distributions and relationships. In the case of tabular data, this means generating rows and columns that resemble the structure of the original dataset, including categorical variables, numerical variables, binary variables, ordinal variables, and target variables. Synthetic tabular data has gained attention because many real-world datasets cannot be freely shared due to privacy, ethical, legal, or organisational restrictions [7], [9].

The main motivation for synthetic tabular data generation is to support data access while reducing direct dependence on sensitive individual-level records. In do-

mains such as healthcare, finance, public administration, and social science, datasets often contain personal or sensitive information. These datasets may be useful for model development, statistical analysis, benchmarking, teaching, or reproducibility studies, but access to them is often restricted. Synthetic data provides one possible way to make data-driven work easier by creating artificial records that preserve important properties of the real data without directly publishing the original records [9], [18]. However, synthetic data should not be understood as a complete replacement for real data. Its value depends on how well it represents the real data for the intended purpose and how well it manages possible privacy risks.

Generating synthetic tabular data is a difficult task because tabular datasets are usually heterogeneous. Unlike image or text data, where the structure is often more regular, tabular data may contain a mix of continuous, categorical, binary, ordinal, and highly imbalanced variables. For example, one column may represent age, another may represent education level, another may represent income group, and another may represent a disease outcome. A useful synthetic data generator must therefore learn both the distribution of individual variables and the relationships between variables. This is important because preserving each column separately does not guarantee that the generated dataset will preserve the real structure of the data. For instance, a synthetic dataset may correctly reproduce the distribution of age and income separately, but still fail to preserve the relationship between age, education, occupation, and income.

The difficulty of tabular data generation has also been discussed in the context of deep generative models. Xu et al. [12] explain that tabular data commonly contains mixed discrete and continuous columns, multimodal continuous variables, and imbalanced categorical variables. These features make synthetic tabular data generation more complex than simply sampling from a single distribution. Continuous variables may not follow a normal distribution and may contain several modes, while

categorical variables may contain rare categories that are difficult to learn. If these rare categories are ignored, the synthetic data may look reasonable overall but still fail to represent important minority groups or less frequent outcomes. Therefore, both common and rare patterns need to be considered when evaluating synthetic tabular data.

Synthetic tabular data generation methods can be broadly grouped into statistical, deep learning-based, and DP approaches. Statistical methods aim to estimate the distributional structure of the data using probabilistic assumptions. A common example is the Gaussian Copula-based approach used in the Synthetic Data Vault framework, where marginal distributions and dependency structures are modelled before generating new records [9]. Deep learning-based methods, such as CTGAN and TVAE, attempt to learn more complex patterns from mixed-type tabular datasets [12].

DP methods add formal privacy protection during the generation process, usually by introducing controlled noise or limiting the influence of individual records [16], [19]. These categories are not identical in their goals: Non-DP models often focus more directly on data fidelity, while DP models try to balance data fidelity with privacy protection.

A model that produces synthetic data with very high resemblance to the original data may also increase privacy risk if it memorises training records. In contrast, a model that adds strong privacy protection may reduce the statistical quality or analytical usefulness of the synthetic data.

It is therefore important to avoid assuming that synthetic data is automatically safe just because it is artificial. Stadler et al. [20] show that synthetic data should not be treated as a simple privacy solution without evaluation. A Non DP generator trained on sensitive data may still reveal information about the original records, especially if the model overfits or if the synthetic data contains records that are too

close to real individuals. This is why synthetic data generation should be evaluated not only from the perspective of statistical similarity, but also from the perspective of privacy risk and reproducibility. In privacy-sensitive settings, formal privacy methods such as differential privacy provide a clearer privacy guarantee, but they may also reduce fidelity because of the additional noise required during the generation process.

2.1.1 Non-DP Data Generators

The Non-DP synthesizers used in this study represent both statistical and deep learning-based approaches.

Statistical Methods

- **Gaussian Copula:** It is a statistical synthetic data generation method used in the Synthetic Data Vault framework. The main idea behind copula-based modelling is to separate the marginal distribution of each variable from the dependency structure between variables. In simple terms, the model first learns the distribution of each column and then uses a copula to capture how the columns are related to each other. This makes the method suitable for tabular data where variables may have different data types and different distributions [21], [22].

  In Data Vault framework, it is used as a practical synthesizer for tabular data. It estimates the statistical properties of the original table and then samples new records from the fitted model [9]. Compared with deep learning-based approaches, Gaussian Copula is generally simpler and less computationally demanding. However, Gaussian Copula also has limitations. Since it is based on a statistical approximation of the data distribution, it may struggle when the real data contains complex non-linear relationships, irregular numerical distributions, or high-dimensional interactions that are not captured well by

the estimated dependency structure.

Deep Learning-Based Methods

- **CTGAN:** Due to their powerful performance and adaptability in data representation, generative models beginning with Variational Autoencoders (VAEs) and then Generative Adversarial Networks (GANs) through their numerous modifications [23]–[26] have attracted a lot of interest.

  CTGAN, or Conditional Tabular Generative Adversarial Network, is a deep learning-based model developed specifically for synthetic tabular data generation. It builds on the idea of generative adversarial networks, where two neural networks are trained together: a generator that creates synthetic samples and a discriminator that attempts to distinguish synthetic samples from real samples [27]. Through this adversarial process, the generator learns to produce data that becomes increasingly similar to the real data distribution. GANs are also routinely used to generate tabular data, especially in medical records.

  For instance, [28] suggested a GAN-based technique for producing discrete tabular data, whereas [29] employed GANs to produce continuous time-series medical records. Similarly, medGAN [13] creates heterogeneous, non-time-series data with continuous and/or binary variables by combining an autoencoder with a GAN. The goal of ehrGAN [30] is to produce enhanced electronic health records. TableGAN [22] uses a convolutional neural network to produce synthetic tabular data and enhances the label column's quality so that the generated data can be used to train classifiers. Furthermore, synthetic data in differential privacy guarantees is produced by PATE-GAN [31].

  Tabular data creates specific challenges for standard generative adversarial networks. Xu et al. [12] explain that tabular datasets often contain a mixture of discrete and continuous variables, multimodal continuous variables, and

imbalanced categorical variables. These characteristics make tabular data harder to model than image-like data. CTGAN addresses these issues using techniques such as mode-specific normalisation for continuous variables and conditional training-by-sampling for discrete variables. Mode-specific normalisation helps the model handle continuous variables that do not follow a single normal distribution, while conditional sampling helps the model learn from minority categories more effectively.

The strength of CTGAN is its flexibility. Since it is based on neural networks, it can potentially learn complex relationships between variables and handle different types of tabular data. This makes it useful when the dataset contains non-linear patterns or mixed variable types. However, adversarial training can also be unstable. The generator and discriminator may converge differently across repeated runs, especially when the dataset is small, imbalanced, or contains rare categories. Because of this, CTGAN may show stronger variation in fidelity scores across repeated runs compared with simpler statistical models.

- **TVAE:** It is another deep learning-based method for synthetic tabular data generation. It is based on the general variational autoencoder framework proposed by Kingma and Welling [32]. A variational autoencoder learns a compressed latent representation of the data and then reconstructs or generates new samples from this latent space. The model consists of an encoder, which maps the input data into a latent distribution, and a decoder, which generates data from the latent representation.

  In the context of tabular data, TVAE adapts the variational autoencoder idea to handle mixed data types. Xu et al. [12] introduced TVAE as part of their tabular data generation study, alongside CTGAN. While CTGAN uses adversarial training, TVAE uses reconstruction-based learning through a latent-variable model. This gives TVAE a different training behaviour. It does

not rely on a discriminator, which may make it less affected by the instability of adversarial training. At the same time, because it learns a latent representation, it may smooth the data distribution and may not always preserve rare categories or sharp categorical boundaries well.

- **RTVAE:** It is a robust version of the variational autoencoder approach for tabular data. It is based on the idea that standard variational autoencoders may be sensitive to outliers, noise, or irregular patterns in the training data. Akrami et al. [33] proposed a robust variational autoencoder with beta divergence for tabular data containing mixed categorical and continuous features. The aim of this approach is to reduce the effect of outlying or corrupted observations during representation learning. In practical synthetic data libraries such as SynthCity, RTVAE is provided as a robust VAE-based method for tabular data generation [34].

  The main difference between TVAE and RTVAE is that RTVAE introduces robustness into the learning objective. Standard VAE-based methods usually rely on likelihood-based reconstruction and a regularisation term for the latent distribution. RTVAE modifies this idea by using a robust divergence formulation so that the model is less influenced by unusual data points. This can be useful in real tabular datasets where noise, extreme values, or irregular records are common.

  Although robustness can be useful, it does not automatically guarantee better synthetic data fidelity. If the model treats some rare but valid patterns as outlying behaviour, it may under-represent them in the generated data . Similarly, if the model smooths the distribution too strongly, it may preserve the broad structure of the data but fail to reproduce detailed category-level or variable-level distributions. Therefore, the behaviour of RTVAE may depend on the dataset structure and on the type of variables being generated.

2.1.2 DP Synthetic Data Generators

DP data generators are designed to produce artificial datasets while providing a formal privacy guarantee for individuals in the original data [16]. Unlike Non-DP generators, which mainly aim to learn and reproduce the statistical structure of the real data, these generators must also limit the amount of information that can be revealed about any single record. This makes them useful in settings where data contains sensitive personal, medical, financial, or demographic information. However, the privacy guarantee usually comes with a cost: noise must be introduced during the generation process, and this can reduce the fidelity of the synthetic data.

Differential privacy provides a mathematical definition of privacy-preserving data analysis. The main idea is that the output of an algorithm should not change too much when one individual record is added to or removed from the dataset. In other words, the presence or absence of one person should have only a limited effect on the final output. This reduces the risk that an attacker can infer whether a specific individual was included in the original dataset. Dwork introduced the idea of calibrating noise to the sensitivity of a query, where sensitivity refers to how much the output of a function can change because of one record [14]. Dwork and Roth later provided a broader foundation for differential privacy and described it as a rigorous framework for privacy-preserving data analysis [35].

Definition 1 (Differential Privacy)**.** *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ)-differentially private if, for all subsets $S \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $|x - y|_1 \leq 1$, the following condition holds:*

$$\Pr[\mathcal{M}(x) \in S] \leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in S] + \delta. \tag{2.1}$$

In Equation 2.1, $\mathcal{M}$ denotes the randomized privacy mechanism or algorithm. The set $\mathcal{X}$ represents the universe of possible data records, and $\mathbb{N}^{|\mathcal{X}|}$ represents the

database domain in histogram form. The terms x and y denote two neighbouring databases, and the condition $|x - y|_1 \leq 1$ means that the two databases differ in at most one individual record. The set S represents any possible subset of outputs produced by the mechanism, while Range($\mathcal{M}$) denotes the set of all possible outputs of $\mathcal{M}$. The probability is taken over the random choices made by the mechanism $\mathcal{M}$.

The parameter ϵ controls the privacy loss. A smaller value of ϵ gives stronger privacy protection, but it may also reduce the accuracy of the output.

- **AIM:** This method was proposed by McKenna et al.[16]. It follows a select-measure-generate approach. First, the method selects useful queries or measurements from the data. Second, it privately measures these queries by adding noise according to the privacy budget. Third, it generates synthetic data that is consistent with the noisy measurements. The method is adaptive because it does not select all measurements at once. Instead, it iteratively chooses measurements that are expected to be useful for improving the quality of the synthetic data.

  The main strength of AIM is that it is workload-adaptive. A workload refers to the set of queries or statistics that the synthetic data is expected to preserve. In many tabular data settings, these queries may include low-dimensional marginal distributions. AIM uses the available privacy budget to focus on measurements that are relevant to the workload and useful for approximating the input data. This is important because the privacy budget is limited. If the budget is spent on too many unnecessary measurements, the amount of noise added to each measurement may increase, reducing the quality of the generated data. By selecting measurements adaptively, AIM attempts to use the privacy budget more effectively.

- **PrivBayes:** PrivBayes was proposed by Zhang et al. [19] for private release of high-dimensional data. The method builds a Bayesian network to represent

dependencies between attributes in the dataset. Instead of modelling the full high-dimensional joint distribution directly, it approximates the data distribution using a set of lower-dimensional conditional distributions. Noise is then added to these distributions to satisfy differential privacy, and synthetic records are sampled from the resulting noisy Bayesian network.

The motivation behind PrivBayes was that directly releasing or modelling high-dimensional data under differential privacy can require a large amount of noise. When a dataset has many variables, the full joint distribution becomes very large, and estimating it privately can become inaccurate. A Bayesian network provides a more compact representation of dependencies between variables. By decomposing the joint distribution into smaller conditional distributions, PrivBayes attempts to preserve important relationships while reducing the dimensionality of the private estimation problem.

PrivBayes has two important stages that can affect its performance. The first stage is the private construction of the Bayesian network structure. This stage determines which variables are treated as dependent on which other variables. The second stage is the private estimation of the conditional distributions used for sampling synthetic data. Both stages can be affected by the privacy budget, sample size, and variable cardinality. When the sample size is small or when variables have many categories, the noisy conditional distributions may become less reliable. This can lead to higher distance values or greater run-to-run variability in the generated data.

2.1.3 Evaluation of Synthetic Data Quality

A synthetic dataset is not useful simply because it has the same columns as the original dataset or because it contains artificial records. Its quality depends on whether it preserves the important statistical properties of the real data, whether it

remains useful for analysis, and whether it provides acceptable privacy protection. For this reason, synthetic data evaluation is very important.

A synthetic dataset with high fidelity may preserve the real data very well, but this may also increase privacy risk if individual-level patterns are reproduced too closely. A dataset with strong privacy protection may reduce disclosure risk, but it may also have lower fidelity or weaker utility due to added noise. Similarly, a dataset may perform well for a specific prediction task but still fail to preserve some marginal distributions or minority categories. Therefore, synthetic data quality should not be judged using only one score. The evaluation should be connected to the purpose of the synthetic data and the risks involved in its use.

Achieving a perfect match is challenging, though, particularly when privacy regulations are in place [36], and it may even be undesirable when biases are present [37]. Examining low-dimensional marginals [13], the syntactical quality of the synthetic data [38], or the distribution of the other attributes conditional on a feature which is known to be biased in the original data are some alternatives to looking for a complete statistical match.

Repeated-run evaluation is therefore useful because it shows whether the quality of a synthesizer is reproducible across runs. For example, a method may produce a low average fidelity distance, but if the distance values vary widely across runs, the method may be less reliable in practice. In contrast, another method may have slightly higher average distance but much lower variation, meaning that it produces more predictable results. This distinction matters in practical applications because researchers and organisations need synthetic data generation methods that are not only accurate in one run, but also stable across repeated use.

2.1.4 Fidelity Metrics for Synthetic Data Evaluation

In synthetic tabular data evaluation, fidelity can be assessed at different levels. Univariate fidelity focuses on individual variables and examines whether each column in the synthetic data has a distribution similar to the corresponding column in the real data. Multivariate fidelity focuses on relationships between variables, such as correlations, conditional distributions, and joint distributions. Since this thesis focuses on univariate distributional evaluation, the selected fidelity metrics compare real and synthetic empirical probability distributions at the variable level.

Distance-based fidelity metrics are commonly used because they provide a direct way to quantify the difference between real and synthetic distributions. In this setting, a lower distance value indicates that the synthetic distribution is closer to the real distribution, while a higher value indicates weaker fidelity. For categorical variables, the empirical distribution can be obtained by calculating the relative frequency of each category. For binned numerical variables, the same approach can be used after converting continuous values into discrete intervals. For numerical variables evaluated through value frequencies or bins, the real and synthetic values must be aligned before the metric is calculated. This alignment is important because a category or bin may appear in one dataset but not in the other.

Hellinger Distance

It is a probability distance used to quantify the divergence between two probability distributions. In this study, it is going to measure how much difference is there between the empirical distribution of a variable in the synthetic data from the corresponding distribution in the real data. A lower Hellinger distance therefore indicates stronger distributional fidelity [39]. Since Hellinger distance is symmetric and bounded between 0 and 1, and this bound is independent of the distribution support, it is suitable for comparing fidelity across variables with different numbers

of categories, bins, or observed values.

For a categorical or discrete variable, let $p = (p_1, p_2, \ldots, p_k)$ denote the empirical probability distribution of a variable in the real data, and let $q = (q_1, q_2, \ldots, q_k)$ denote the corresponding empirical probability distribution in the synthetic data. Here, k is the number of aligned categories, bins, or observed values. The discrete Hellinger distance is defined as:

$$H(p, q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} \left(\sqrt{p_i} - \sqrt{q_i}\right)^2} \tag{2.2}$$

In Equation (2.2), p_i and q_i represent the probabilities of the i^{th} category, bin, or observed value in the real and synthetic distributions, respectively.

The value of Hellinger distance lies within the bounded interval shown by the Equation (2.3).

$$0 \leq H(p, q) \leq 1 \tag{2.3}$$

In this study, the empirical discrete form in Equation (2.2) is used for categorical variables and for numerical variables after alignment through observed values or bins. Therefore, the calculation is based on aligned empirical probability distributions rather than continuous probability density estimation.

Jensen Shannon Distance

This is another distributional distance used to compare probability distributions. It is based on Jensen Shannon divergence, which is a symmetric and smoothed form of Kullback Leibler divergence. The Jensen Shannon divergence was introduced as an information-theoretic divergence measure based on Shannon entropy [40]. Unlike direct Kullback Leibler divergence, Jensen Shannon divergence is symmetric and remains well defined in many cases where the two distributions do not have exactly

the same support.

For two probability distributions p and q, the midpoint distribution m is defined as:

$$m = \frac{p+q}{2} \tag{2.4}$$

Using the midpoint distribution from the Equation (2.4), the Jensen Shannon divergence between p and q is calculated as follows:

$$\mathrm{JSD}_{\mathrm{div}(p,q)} = \frac{1}{2} D_{\mathrm{KL}}(p \parallel m) + \frac{1}{2} D_{\mathrm{KL}}(q \parallel m) \tag{2.5}$$

In Equation (2.5), D_{KL} denotes the Kullback Leibler divergence. Based on the midpoint distribution from Equation (2.4), Equation (2.5) can be rewritten as:

$$\mathrm{JSD}_{\mathrm{div}(p,q)} = \frac{1}{2} D_{\mathrm{KL}}\left(p \parallel \frac{p+q}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(q \parallel \frac{p+q}{2}\right) \tag{2.6}$$

The Jensen Shannon distance is obtained by taking the square root of the Jensen Shannon divergence, as shown in Equation (2.7).

$$\mathrm{JSD}(p,q) = \sqrt{\mathrm{JSD}_{\mathrm{div}}(p,q)} \tag{2.7}$$

Endres and Schindelin [41] showed that the square root form in Equation 2.7 defines a metric for probability distributions. This makes Jensen–Shannon distance suitable for comparing real and synthetic probability distributions in a stable and interpretable way. When logarithms with base 2 are used, Jensen Shannon divergence is bounded between 0 and 1. Therefore, Jensen Shannon distance also lies between 0 and 1. A value of 0 indicates identical distributions, while values closer to 1 indicate larger differences between the real and synthetic distributions.

3 Methodology

In this chapter we will present the experimental procedure used in this study. Here we will presents the datasets, preprocessing steps, synthetic data generation methods, experimental settings, and evaluation procedure.

The experiments were carried out using three public tabular datasets: Adult [42], Bank Marketing [43], and Cardiovascular Disease [44] which will be discussed in detail later in the chapter.

3.1 Experimental Design Overview

The experimental design followed a fixed pipeline consisting of four main stages: dataset preprocessing, synthetic data generation, fidelity metric calculation, and result aggregation. First, each original dataset was cleaned and preprocessed as a complete dataset. After preprocessing, samples of size $n = 500$, $n = 5000$, and $n = 10000$ were drawn from the preprocessed data and used to train the synthesizers.

The same overall procedure was applied to all datasets and synthesizers. For the Non-DP synthesizers, the cleaned preprocessed datasets were used. These synthesizers included Gaussian Copula, CTGAN, TVAE, and RTVAE. For the DP synthesizers, the preprocessed dataset was discretised and then it was used. These synthesizers included AIM and PrivBayes. The DP synthesizers were evaluated using three privacy budget values: $\epsilon = 1$, $\epsilon = 5$, and $\epsilon = 10$.

For each valid configuration, 100 synthetic datasets were generated using separate

random seeds to make the repeated experiments reproducible. The generated synthetic datasets were evaluated using Hellinger distance and Jensen Shannon distance as fidelity metrics. The repeated runs were then used to examine the run-to-run variation of these fidelity scores.

Some AIM configurations were excluded from the final analysis because they could not be completed within the available computational resources. These excluded configurations were AIM with $\epsilon = 5$ for the Bank dataset at $n = 10000$, AIM with $\epsilon = 10$ for the Adult dataset at $n = 10000$, and AIM with $\epsilon = 10$ for the Bank dataset at both $n = 5000$ and $n = 10000$. For all completed configurations, repeated runs were used to examine the run-to-run variation of the fidelity metrics including Hellinger distance and Jensen Shannon distance.

3.2 Datasets

The following subsections describe each of the three dataset and its final representation in the experiments.

Adult Dataset

The Adult dataset was obtained from the UCI Machine Learning Repository [42]. It contained the demographic and employment-related records and it is commonly used for income classification for target variable `income`. The Adult dataset contained 32,561 records and 15 variables. The dataset consisted of 9 categorical variables and 6 numerical variables.

Bank Dataset

The Bank Marketing dataset was obtained from the UCI Machine Learning Repository [43]. The dataset contained records from direct marketing campaigns of a Portuguese

banking institution, where the target variable indicates whether a client subscribed to a term deposit. The target variable was `y`, with two classes: `yes` and `no`. The raw version of the dataset contained 45,211 records and 17 variables. It consists of 10 categorical and 7 numerical variables.

Cardio Dataset

The Cardiovascular Disease dataset was obtained from Kaggle [44]. The dataset contained patient-level health and lifestyle information. The target variable is `cardio`, which indicated whether cardiovascular disease is present or not. The raw dataset contained 69,988 records. It had 13 variables out of which 7 were categorical variables and 5 were numerical variables.

3.3 Data Preprocessing

In the preprocessing, each dataset was converted into a consistent tabular format, each was checked for missing values, and saved as a preprocessed file.

Data Cleaning

Initial preprocessing was carried out separately for each full dataset before synthetic data generation. The main steps what were included were: assigning suitable column names, removing unnecessary identifier columns, checking missing values, and preparing the datasets in a consistent tabular format.

For the Adult dataset, the column names were assigned manually because the original data file did not include a header row. After assigning the column names, the dataset was checked for missing values, and no missing values were found in any variables except for three variables `workclass`, `occupation`, and `native_country`. Each had entries represented by unknown marker `?`, and they were handled by

replacing them with the category "Unknown". This avoided removing records to keep the originality of the dataset.

For the Bank dataset, it was checked for missing values and no missing values were found except for four variables `job`, `education`, `contact` and `poutcome`. The value `unknown` was retained as a valid category for these variables.

For the Cardiovascular Disease dataset, the identifier column was removed because it did not provide useful information for synthetic data generation. The `age` variable was converted from days to years. The dataset was checked for missing values, and no missing values were found, and after removing the identifier column, only 12 variables were used for synthetic data generation.

Dataset Versions Used for Synthetic Data Generation:

After data cleaning, two versions of each dataset were created for each dataset. The first version preserved the cleaned numerical and categorical variables. This version was used for the Non-DP synthesizers.

The second version was used for DP synthesizers, In this version, numerical variables were replaced with binned variables as discussed in next section.

Binning

Numerical variables were transformed into binned variables. Discretization is commonly used to convert continuous attributes into categorical intervals and is often required in DP synthetic data generation workflows [45]. Since discretisation was applied only to the DP dataset versions, the DP and Non-DP synthesizers were trained on different representations of the same original datasets. Therefore, the comparison between DP and Non-DP methods should be interpreted with this preprocessing difference in mind.

Two binning strategies were used. Manual binning was applied when meaningful

interval ranges could be defined directly. Quantile-based binning was applied when variables had skewed distributions and required more balanced bin sizes [45], [46].

Several numerical variables in each dataset were discretised using either interval-based ranges or quantile-based binning. In the Adult dataset, five numerical variables were binned: `age`, `fnlwgt`, `capital_gain`, `capital_loss`, and `hours_per_week`. In the Bank dataset, seven numerical variables were binned: `age`, `balance`, `day`, `duration`, `campaign`, `pdays`, and `previous`. In the Cardio dataset, five numerical variables were binned: `age`, `height`, `weight`, `ap_hi`, and `ap_lo`. All of these variable were converted into corresponding binned variables. The resulting discretised datasets were used as the DP dataset versions for AIM and PrivBayes.

3.4 Experimentation

Data generation was carried out separately for each dataset, synthesizer, sample size, and privacy setting. For each run, a real data sample of size n was first drawn from the corresponding preprocessed dataset. The synthesizer was then trained on this sample and used to generate a synthetic dataset of the same size.

The generated datasets were saved directly after generation, without applying additional post-processing. Some of the python libraries and their versions are discussed in the Table 3.1.

Gaussian Copula was implemented using the python SDV library and its version can be seen in the Table 3.1. The model first detected the metadata of the input table, and categorical variables were specified using the dataset configuration. CTGAN and TVAE were implemented using the `ctgan` library. Both models were trained using the list of discrete columns defined for each dataset. RTVAE was implemented using the `synthcity` library, where each real data sample was passed through a generic tabular data loader before model fitting.

AIM was implemented using the `snsynth` library. PrivBayes was implemented

using the `synthcity` library.

For each valid configuration, 100 synthetic datasets were generated. For CTGAN, TVAE, and RTVAE, the default number of training epochs or iterations was set to 300, and the batch size was set to 250. Gaussian Copula does not use epochs or batch size in the same way as neural network-based models.

Library	**Version**	**Purpose**	**Reference**
SciPy	1.13.1	Calculation of distance metrics (Jensen Shannon)	[47]
CTGAN	0.12.1	Implementation of CTGAN and TVAE.	[48]
SDV	1.36.1	Implementation of the Gaussian Copula.	[49]
SynthCity	0.2.4	Implementation of RTVAE and PrivBayes.	[50]
SmartNoise (`snsynth`)	1.0.8	Implementation of the AIM.	[51]
MBI	1.1.0	Supporting package used by AIM.	[52]

Table 3.1: Version of libraries used in the experiments

Three sample sizes were used in the experiments: $n = 500$, $n = 5000$, and $n = 10000$. These sample sizes were applied to each dataset and synthesizer where computationally feasible. The same sample size was used for both training and generation, meaning that a model trained on n real records generated n synthetic records.

Each valid experimental configuration was repeated 100 times. The generated

synthetic datasets were saved as separate CSV files for each run. In each run, a different random seed was used. The seed was assigned according to the run index: seed = run_index -1

For DP synthesizers, three privacy budget values were used and they were passes as model parameter during the training: $\epsilon \in \{1, 5, 10\}$

Computational Environment

The experiments were executed on the Puhti supercomputer using SLURM job scripts. The jobs were submitted separately for different combinations of dataset, synthesizer, sample size, and privacy budget.

3.5 Evaluation Procedure

The evaluation was carried out using univariate distributional comparisons between real and synthetic data. For each generated synthetic dataset, the corresponding real sample was recreated using the same sample size and random seed used during synthetic data generation. This ensured that each synthetic dataset was compared with the real sample used to train its synthesizer.

Since analysing every variable in detail would make the results section unnecessarily long, five representative variables were selected from each dataset for the main analysis. For the Adult dataset, the selected variables were `workclass`, `education`, `relationship`, `occupation`, and `income`. For the Bank dataset, the selected variables were `job`, `education`, `contact`, `poutcome`, and `y`. For the Cardio dataset, the selected variables were `age`, `cholesterol`, `gluc`, `ap_hi`, and `cardio`. These variables were chosen to provide examples of categorical, binary, ordinal, and numerical variables across all three datasets.

For each variable, the real and synthetic columns were converted into empirical

probability distributions. The categories or bins present in the real and synthetic data were aligned before calculating the distance metrics. If a category appeared in one distribution but not in the other, its probability was set to zero for the missing side.

Aggregation Across Runs

For each synthetic dataset, distance scores were calculated separately for each variable and the results were stored in a separate file.

For each valid experimental configuration, the distance scores were then aggregated across repeated runs. The mean distance was used to summarise the average fidelity of a synthesizer for a given variable, while the standard deviation (SD) was used to describe how much the fidelity scores varied across repeated runs. In addition to the mean and standard deviation, the median and interquartile range were also computed. The median provided a summary of the typical fidelity score, especially when some runs produce unusually high or low distance values. The interquartile range measures the spread of the middle 50% of the repeated-run scores and is therefore it is useful for describing variability in a way that is less affected by extreme values.

The aggregated statistics were also supported by boxplots. Boxplots were used to visualise the distribution of fidelity scores across repeated runs for each dataset, synthesizer, sample size, privacy budget where applicable, and selected variable. In these plots, the central line represents the median, the box represents the interquartile range, and the whiskers and outlying points show the wider spread of the repeated-run results. Therefore, the boxplots provide a visual way to compare both fidelity and run-to-run stability across synthesizers and experimental settings.

For a distance score d_r obtained in run r, the mean distance across R runs is defined as: $\bar{d} = \frac{1}{R} \sum_{r=1}^{R} d_r$

The standard deviation across runs is defined as: $s = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}(d_r - \bar{d})^2}$ where R is the number of completed runs for the configuration. Lower mean distance indicates better distributional similarity, while a lower standard deviation indicates lower run-to-run variability across repeated runs.

Median can be calculated as follows: the distance scores were first arranged from the smallest to the largest value. If the number of runs was odd, the middle value was taken as the median. If the number of runs was even, the median was calculated as the average of the two middle values.

The interquartile range is defined as the difference between the third quartile and the first quartile: $\text{IQR} = Q_3 - Q_1$

where Q_1 is the first quartile and Q_3 is the third quartile of the ordered distance scores.

4 Results and Discussion

In this chapter, we will present the experimental results of the univariate evaluation. The analysis is organised by synthesizer type, and it begins with the Non-DP synthesizers and then proceeds to the DP synthesizers.

The tables report summarises the statistics such as mean, median, standard deviation, and interquartile range, while the boxplots show the distribution of fidelity scores across repeated runs.

The analysis in this chapter focuses on a selected set of variables from each dataset. For the Adult dataset, the selected variables are `workclass`, `education`, `relationship`, `occupation`, and `income`. For the Bank dataset, the selected variables are `job`, `education`, `contact`, `poutcome`, and `y`. For the Cardio dataset, the selected variables are `age`, `cholesterol`, `gluc`, `ap_hi`, and `cardio`. In the DP analysis, numerical variables are evaluated in their discretised form where applicable, for example `age_bin` and `ap_hi_bin` for the Cardio dataset.

The complete set of variable-level boxplots for all three datasets is provided in Appendix A.

4.1 Non-DP Synthesizers

This section presents the univariate fidelity results for the Non-DP synthesizers: Gaussian Copula, CTGAN, TVAE, and RTVAE.

4.1.1 Adult Dataset

The tables 4.1 and 4.2 show that both of the distances for Gaussian Copula are the lowest across the selected variables. Its average Hellinger distance decreases from about 0.044 at n = 500 to 0.014 at n = 5000 and 0.010 at n = 10000. A similar trend appears for Jensen Shannon Distance, where the average distance decreases from about 0.052 to 0.017 and then 0.012. For example, for `workclass`, the mean distance decreases from 0.042 at $n = 500$ to 0.015 at $n = 5000$ and 0.011 at $n = 10000$. The box plots 4.1 and 4.2 are also narrow, showing low run-to-run variation.

CTGAN

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.055	0.021	0.025	0.054	0.079	0.028	0.033	0.071	0.083	0.027	0.028	0.077
education	0.094	0.018	0.023	0.093	0.081	0.010	0.010	0.080	0.078	0.008	0.011	0.078
relationship	0.074	0.024	0.033	0.071	0.078	0.022	0.030	0.078	0.076	0.019	0.025	0.074
occupation	0.082	0.014	0.021	0.083	0.105	0.020	0.028	0.104	0.111	0.020	0.024	0.112
income	0.049	0.033	0.046	0.047	0.033	0.023	0.037	0.031	0.029	0.018	0.025	0.029

Gaussian Copula

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.042	0.012	0.017	0.043	0.015	0.005	0.006	0.015	0.011	0.003	0.004	0.011
education	0.062	0.011	0.013	0.063	0.020	0.004	0.005	0.020	0.014	0.002	0.003	0.014
relationship	0.037	0.012	0.020	0.036	0.011	0.003	0.004	0.011	0.008	0.003	0.003	0.008
occupation	0.059	0.012	0.021	0.057	0.019	0.003	0.004	0.020	0.014	0.003	0.004	0.014
income	0.021	0.012	0.018	0.018	0.006	0.004	0.006	0.005	0.003	0.003	0.004	0.003

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.1: Hellinger distance summary for selected Adult variables using CTGAN and Gaussian Copula

CTGAN produces higher distances than Gaussian Copula, but its results remain relatively controlled. Its average Hellinger distance stays around 0.071–0.075, while its Jensen Shannon Distance stays around 0.085–0.090 across the three sample sizes. The CTGAN box plots Figure 4.1, 4.2 show moderate spread, especially for `occupation`, but the median and mean values remain fairly close, suggesting no major skew in most variables.

CTGAN

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.066	0.025	0.030	0.064	0.094	0.034	0.040	0.084	0.099	0.032	0.034	0.091
education	0.112	0.021	0.027	0.112	0.097	0.012	0.012	0.096	0.093	0.010	0.013	0.093
relationship	0.088	0.029	0.039	0.085	0.094	0.027	0.036	0.093	0.091	0.022	0.030	0.089
occupation	0.099	0.017	0.026	0.099	0.126	0.024	0.033	0.124	0.133	0.024	0.029	0.134
income	0.059	0.039	0.056	0.056	0.040	0.027	0.044	0.037	0.034	0.022	0.030	0.035

Gaussian Copula

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.049	0.015	0.020	0.050	0.018	0.005	0.007	0.017	0.013	0.003	0.004	0.013
education	0.073	0.013	0.018	0.075	0.024	0.005	0.005	0.024	0.017	0.003	0.003	0.017
relationship	0.045	0.015	0.024	0.043	0.013	0.004	0.005	0.013	0.010	0.003	0.003	0.009
occupation	0.069	0.014	0.024	0.067	0.023	0.004	0.005	0.023	0.016	0.003	0.004	0.016
income	0.025	0.015	0.022	0.022	0.007	0.005	0.007	0.006	0.004	0.003	0.005	0.003

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.2: Jensen–Shannon distance summary for selected Adult variables using CTGAN and Gaussian Copula

For the autoencoder-based models in tables 4.3 and 4.4, the Adult dataset shows larger distances and greater variation. TVAE has relatively high distances at n = 500, with an average Hellinger distance of about 0.286 and Jensen Shannon Distance of about 0.308. However, its results improve clearly at when the size of is increased, with the average Hellinger distance falling to 0.107 at n = 5000 and 0.086 at n = 10000. The same pattern appears for Jensen Shannon Distance.

RTVAE has the weakest overall performance in this dataset, with the highest average distances and the widest box plots. Its average Hellinger distance decreases from 0.326 to 0.248 and then 0.229, while Jensen Shannon Distance decreases from 0.349 to 0.269 and then 0.248. The largest spreads appear for `education` and `occupation`, showing that these variables are harder to reproduce across runs. In contrast, `income` shows lower distances and smaller variation across most synthesizers.

For a more detailed variable-level view of the boxplot distributions for the Adult dataset, including all variables can be referred to Appendix A, Figures A.1 and A.2.

In addition to the distance values, the stability pattern also differs across the

TVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.393	0.018	0.026	0.397	0.169	0.048	0.076	0.172	0.114	0.030	0.049	0.112
education	0.317	0.033	0.044	0.317	0.120	0.027	0.030	0.118	0.109	0.026	0.029	0.107
relationship	0.256	0.036	0.058	0.256	0.087	0.021	0.027	0.089	0.081	0.017	0.023	0.080
occupation	0.331	0.035	0.048	0.332	0.150	0.026	0.039	0.147	0.115	0.020	0.027	0.116
income	0.132	0.038	0.049	0.137	0.011	0.009	0.011	0.009	0.013	0.008	0.011	0.011

RTVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.131	0.044	0.067	0.127	0.079	0.033	0.030	0.074	0.055	0.026	0.020	0.050
education	0.651	0.048	0.049	0.640	0.560	0.192	0.306	0.671	0.536	0.176	0.299	0.628
relationship	0.079	0.030	0.035	0.077	0.044	0.020	0.018	0.041	0.038	0.018	0.019	0.037
occupation	0.729	0.055	0.044	0.739	0.531	0.119	0.093	0.584	0.501	0.104	0.109	0.531
income	0.041	0.030	0.045	0.040	0.024	0.017	0.022	0.021	0.014	0.010	0.011	0.012

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.3: Hellinger distance summary for selected Adult variables using TVAE and RTVAE

TVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.404	0.016	0.022	0.407	0.191	0.050	0.081	0.195	0.133	0.033	0.052	0.131
education	0.338	0.032	0.042	0.342	0.141	0.032	0.033	0.140	0.128	0.031	0.035	0.126
relationship	0.282	0.036	0.054	0.283	0.102	0.023	0.031	0.105	0.096	0.019	0.025	0.096
occupation	0.357	0.037	0.047	0.358	0.174	0.029	0.043	0.171	0.136	0.023	0.032	0.138
income	0.157	0.045	0.058	0.163	0.013	0.010	0.013	0.011	0.015	0.010	0.013	0.013

RTVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
workclass	0.150	0.045	0.062	0.146	0.094	0.038	0.036	0.089	0.065	0.031	0.024	0.059
education	0.688	0.054	0.051	0.678	0.607	0.206	0.338	0.723	0.582	0.190	0.339	0.682
relationship	0.094	0.035	0.041	0.092	0.053	0.024	0.022	0.049	0.046	0.021	0.023	0.044
occupation	0.765	0.057	0.048	0.778	0.560	0.118	0.099	0.609	0.529	0.105	0.118	0.557
income	0.050	0.035	0.054	0.047	0.029	0.020	0.026	0.026	0.017	0.012	0.013	0.014

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.4: Jensen–Shannon distance summary for selected Adult variables using TVAE and RTVAE

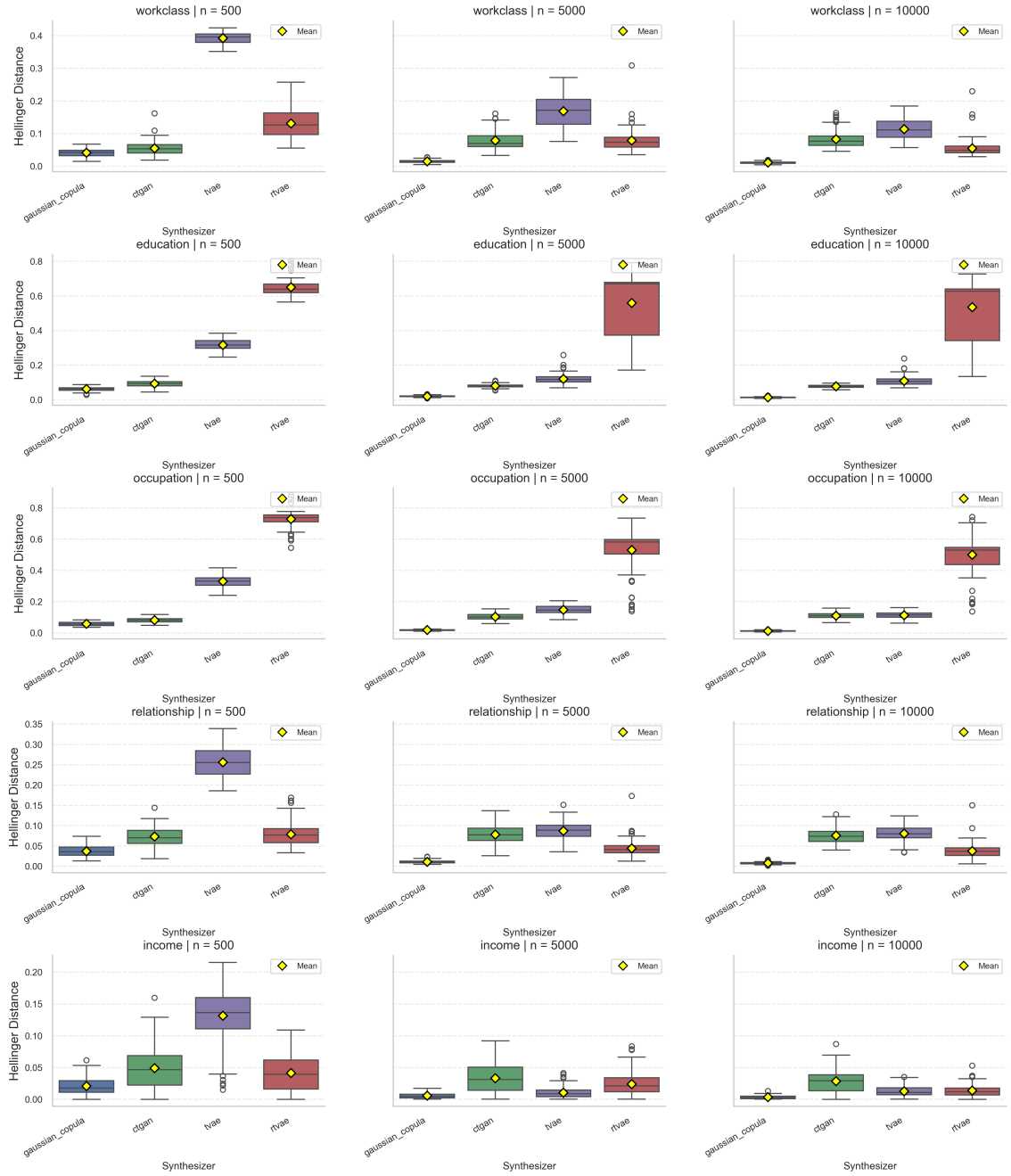

Figure 4.1: Hellinger distance results for selected variables in the Adult dataset.

synthesizers. Gaussian Copula shows the lowest run-to-run dispersion across repeated runs, with its average Hellinger standard deviation decreasing from about 0.012 at n = 500 to around 0.003 at n = 10000. This suggests that increasing the sample size makes its results not only more accurate but also reduces run-to-run dispersion. From Figure 4.1 and 4.2, the CTGAN shows a different pattern, as its variability changes only slightly with larger samples, with the average Hellinger standard deviation moving from about 0.022 at n = 500 to about 0.018 at n = 10000. This indicates that CTGAN's variation is not mainly caused by sample size, but may also come from the random nature of adversarial training. RTVAE shows a clear high-dispersion issue for `education`, where the Hellinger standard deviation remains high at larger sample sizes, around 0.192 at n = 5000 and 0.176 at n = 10000. This means that repeated RTVAE runs are not able to reproduce the variable across runs. The results also suggest that `income` is easier to model because it has only two categories, while `education` and `occupation` are more difficult because they contain more categories and uneven category frequencies, making rare groups harder to preserve.

4.1.2 Bank Dataset

From the table 4.5, the Gaussian Copula average Hellinger distance decreases from about 0.032 at n = 500 to 0.009 at n = 5000 and 0.007 at n = 10000. Jensen Shannon Distance follows the same trend, decreasing from about 0.038 to 0.011 and then 0.008. The box plots 4.3 are very compact across all variables, showing that Gaussian Copula remains stable across repeated runs.

CTGAN has moderate distances, but unlike Gaussian Copula, its distances do not decrease with sample size. Its average Hellinger distance increases from about 0.058 at n = 500 to 0.073 at n = 5000 and remains around 0.071 at n = 10000. For `job`, the mean distance increases from 0.073 at $n = 500$ to 0.110 at $n = 5000$ and remains high at 0.107 for $n = 10000$. A similar pattern is observed for `poutcome`. This suggests

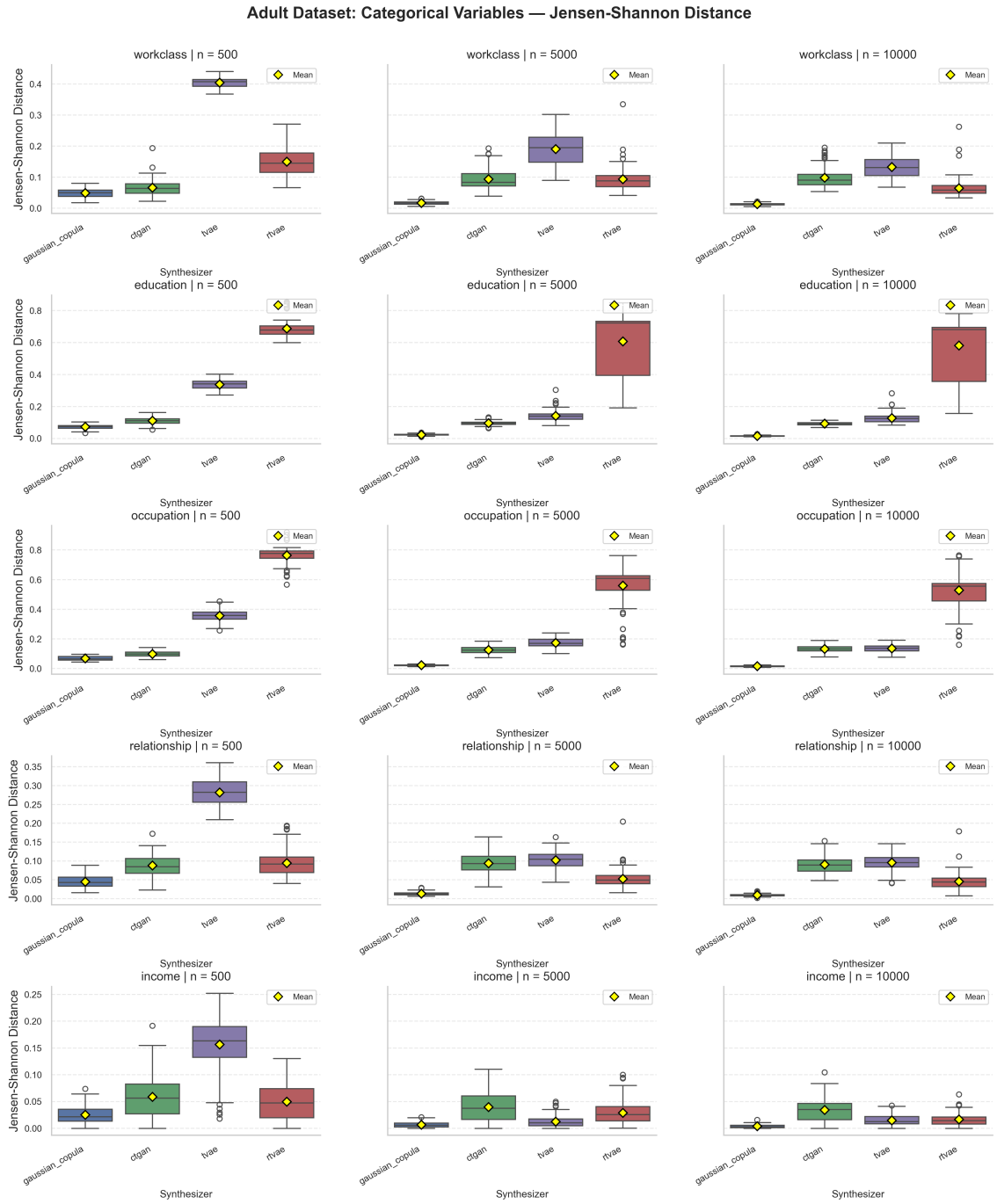

Figure 4.2: Jensen–Shannon distance results for selected variables in the Adult dataset.

CTGAN

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.061	0.029	0.041	0.061	0.055	0.025	0.036	0.053	0.058	0.027	0.039	0.053
education	0.058	0.025	0.036	0.052	0.080	0.035	0.050	0.077	0.066	0.025	0.035	0.063
job	0.073	0.016	0.024	0.072	0.110	0.024	0.031	0.111	0.107	0.029	0.043	0.105
poutcome	0.048	0.020	0.024	0.045	0.066	0.013	0.015	0.063	0.064	0.016	0.020	0.061
y	0.047	0.033	0.055	0.047	0.054	0.045	0.066	0.042	0.057	0.045	0.052	0.043

Gaussian Copula

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.024	0.013	0.015	0.023	0.007	0.004	0.004	0.006	0.005	0.003	0.004	0.005
education	0.029	0.013	0.018	0.027	0.009	0.004	0.005	0.008	0.007	0.003	0.004	0.006
job	0.055	0.013	0.017	0.054	0.016	0.003	0.004	0.016	0.011	0.002	0.003	0.011
poutcome	0.029	0.013	0.015	0.028	0.008	0.004	0.005	0.008	0.007	0.002	0.003	0.006
y	0.021	0.015	0.021	0.018	0.005	0.004	0.005	0.004	0.004	0.003	0.004	0.004

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.5: Hellinger distance summary for selected Bank variables using CTGAN and Gaussian Copula

CTGAN

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.073	0.035	0.049	0.073	0.066	0.030	0.044	0.063	0.070	0.033	0.047	0.063
education	0.069	0.030	0.043	0.062	0.096	0.042	0.060	0.092	0.079	0.029	0.042	0.076
job	0.088	0.020	0.029	0.087	0.132	0.028	0.037	0.132	0.128	0.034	0.051	0.126
poutcome	0.058	0.024	0.029	0.054	0.079	0.015	0.018	0.076	0.077	0.019	0.024	0.073
y	0.057	0.040	0.066	0.056	0.065	0.053	0.079	0.051	0.068	0.054	0.062	0.051

Gaussian Copula

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.028	0.016	0.018	0.027	0.008	0.004	0.005	0.008	0.006	0.003	0.004	0.005
education	0.035	0.015	0.022	0.033	0.011	0.005	0.007	0.010	0.008	0.003	0.004	0.007
job	0.066	0.014	0.021	0.064	0.020	0.004	0.005	0.019	0.014	0.003	0.004	0.014
poutcome	0.035	0.015	0.018	0.034	0.010	0.005	0.006	0.009	0.008	0.003	0.004	0.008
y	0.025	0.017	0.026	0.021	0.006	0.005	0.006	0.005	0.005	0.004	0.005	0.004

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.6: Jensen–Shannon distance summary for selected Bank variables using CTGAN and Gaussian Copula

that larger training samples do not necessarily improve CTGAN's univariate fidelity for these categorical variables. The variability of CTGAN's fidelity scores is generally moderate, although the target variable `y` shows wider dispersion, which may be related to its imbalanced binary distribution. Jensen Shannon Distance follows the same pattern, increasing from 0.069 to around 0.088 and 0.084. The boxplots show more spread for `job` and `y`, while contact and poutcome remain comparatively more stable.

TVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.200	0.022	0.033	0.199	0.123	0.019	0.023	0.122	0.127	0.017	0.017	0.127
education	0.268	0.042	0.062	0.264	0.132	0.014	0.018	0.133	0.135	0.014	0.019	0.135
job	0.385	0.037	0.050	0.386	0.215	0.021	0.028	0.212	0.203	0.017	0.021	0.201
poutcome	0.199	0.015	0.020	0.198	0.095	0.023	0.028	0.097	0.099	0.019	0.028	0.099
y	0.194	0.032	0.045	0.195	0.039	0.030	0.044	0.033	0.040	0.028	0.046	0.035

RTVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.042	0.020	0.025	0.040	0.039	0.014	0.017	0.041	0.037	0.015	0.022	0.033
education	0.062	0.025	0.035	0.060	0.036	0.015	0.019	0.036	0.032	0.012	0.017	0.032
job	0.666	0.051	0.045	0.680	0.379	0.123	0.113	0.368	0.358	0.144	0.160	0.379
poutcome	0.079	0.026	0.039	0.079	0.065	0.027	0.027	0.056	0.052	0.014	0.020	0.052
y	0.053	0.026	0.039	0.051	0.054	0.020	0.030	0.054	0.038	0.016	0.018	0.039

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.7: Hellinger distance summary for selected Bank variables using TVAE and RTVAE

From the table 4.7 and 4.8, for TVAE, it shows a clear improvement from small to larger sample sizes. At n = 500, TVAE has an average Hellinger distance of about 0.249 and Jensen Shannon Distance of about 0.271. These values drop to about 0.121 and 0.136, respectively, at both n = 5000 and n = 10000. The largest TVAE distances occur for job and education, especially at the smallest sample size. RTVAE performs better than TVAE on average in the Bank dataset, but it still shows noticeable variability. Its average Hellinger distance decreases from 0.180 to 0.115 and then 0.103, while Jensen Shannon Distance decreases from 0.196 to 0.130

and then 0.117.

TVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.213	0.023	0.030	0.209	0.143	0.019	0.025	0.142	0.146	0.018	0.020	0.147
education	0.302	0.048	0.072	0.296	0.142	0.011	0.011	0.144	0.147	0.011	0.015	0.148
job	0.414	0.038	0.055	0.417	0.240	0.023	0.028	0.238	0.226	0.018	0.018	0.223
poutcome	0.214	0.014	0.020	0.214	0.111	0.025	0.032	0.113	0.115	0.021	0.030	0.116
y	0.213	0.025	0.037	0.214	0.046	0.036	0.053	0.039	0.048	0.033	0.055	0.042

RTVAE

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
contact	0.050	0.024	0.030	0.048	0.047	0.016	0.020	0.049	0.045	0.018	0.026	0.039
education	0.074	0.028	0.042	0.072	0.043	0.017	0.022	0.043	0.038	0.015	0.020	0.038
job	0.697	0.057	0.050	0.711	0.416	0.137	0.108	0.398	0.393	0.160	0.209	0.401
poutcome	0.094	0.031	0.046	0.094	0.078	0.032	0.032	0.067	0.062	0.016	0.023	0.062
y	0.064	0.031	0.047	0.061	0.065	0.024	0.036	0.065	0.045	0.019	0.021	0.046

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.8: Jensen–Shannon distance summary for selected Bank variables using TVAE and RTVAE

However, RTVAE performs poorly for `job`, where the mean distance remains high across all sample sizes, decreasing from 0.666 at $n = 500$ to 0.358 at $n = 10000$. The box plots 4.3 and 4.4 for `job` also show substantial spread, indicating high run-to-run variability in the fidelity scores. The Bank results show that job is the most challenging selected variable, while contact, `poutcome`, and `y` are relatively more stable.

In terms of run-to-run stability, the Bank dataset further shows that Gaussian Copula benefits the most from larger sample sizes, as its average Hellinger standard deviation decreases from about 0.013 at $n = 500$ to 0.003 at $n = 10000$. CTGAN does not show the same stability gain, since its average Hellinger standard deviation remains almost unchanged, around 0.025–0.028 across the three sample sizes. This indicates that its variation may be linked more to the stochastic adversarial training process than to the amount of training data alone. For RTVAE, the main instability is observed for `job`, where the Hellinger and Jensen Shannon standard deviations

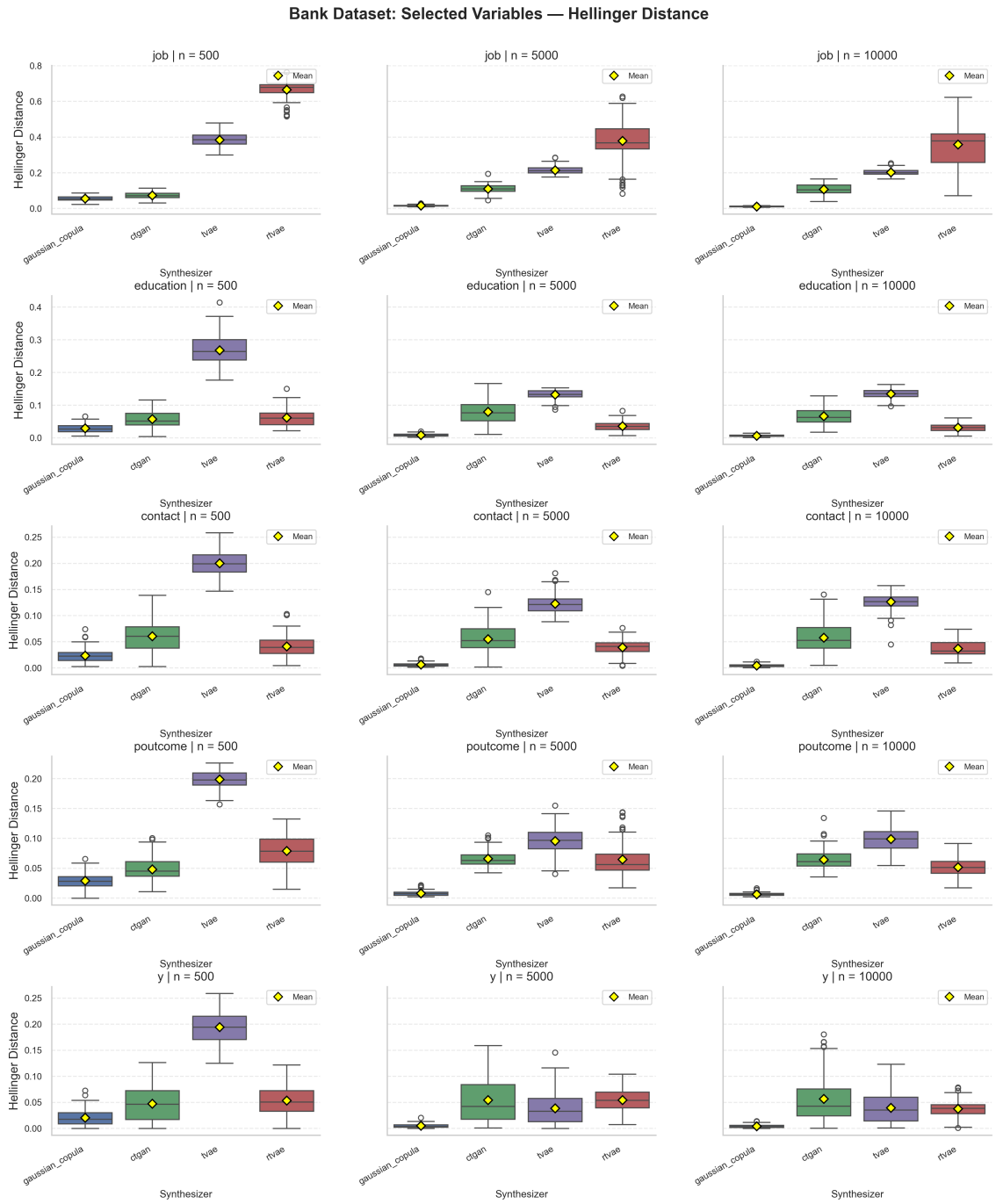

Figure 4.3: Hellinger distance results for selected variables in the Bank dataset.

remain high at $n = 10000$, around 0.144 and 0.160, respectively. This implies that that `job` is difficult to reproduce with low run-to-run variability, likely because it contains several categories with uneven frequencies. The target variable `y` also shows noticeable variation for CTGAN, which may be due to the imbalanced class distribution, where small changes in the minority class proportion can have a visible effect on the distance scores.

For a more detailed variable-level view of the boxplot distributions for the Bank dataset, including all variables, can be referred to Appendix A, Figures A.3 and A.4.

4.1.3 Cardio Dataset

From Table 4.9 and Table 4.10, the overall distance values for Cardio dataset are higher than in Adult and Bank. Gaussian Copula is still the most stable model in terms of boxplot spread, but its mean distances are higher because variables such as `ap_hi` are difficult to reproduce. Its average Hellinger distance is about 0.204 at n = 500, 0.184 at n = 5000, and 0.183 at n = 10000. A similar pattern is observed for `age`. These two variables are ordinal variables, which makes their marginal distributions easier to approximate.

For Jensen Shannon Distance, the average values are about 0.221, 0.201, and 0.200. The boxplots remain narrow, but the distance for `ap_hi` stays high across all sample sizes. CTGAN shows some improvement from n = 500 to larger sample sizes. Its average Hellinger distance decreases from 0.259 to 0.203 and then stays around 0.206, while Jensen Shannon Distance decreases from 0.277 to 0.225 and then 0.230. However, the CTGAN box plots from the Figures 4.5 and 4.6, it can be observed that they become wider at larger sample sizes for some variables, especially cardio, gluc, and cholesterol.

From the Table 4.11 and 4.12, the Cardio dataset shows a decrease in average distance after increasing the sample size from 500 to 5000. Its average Hellinger

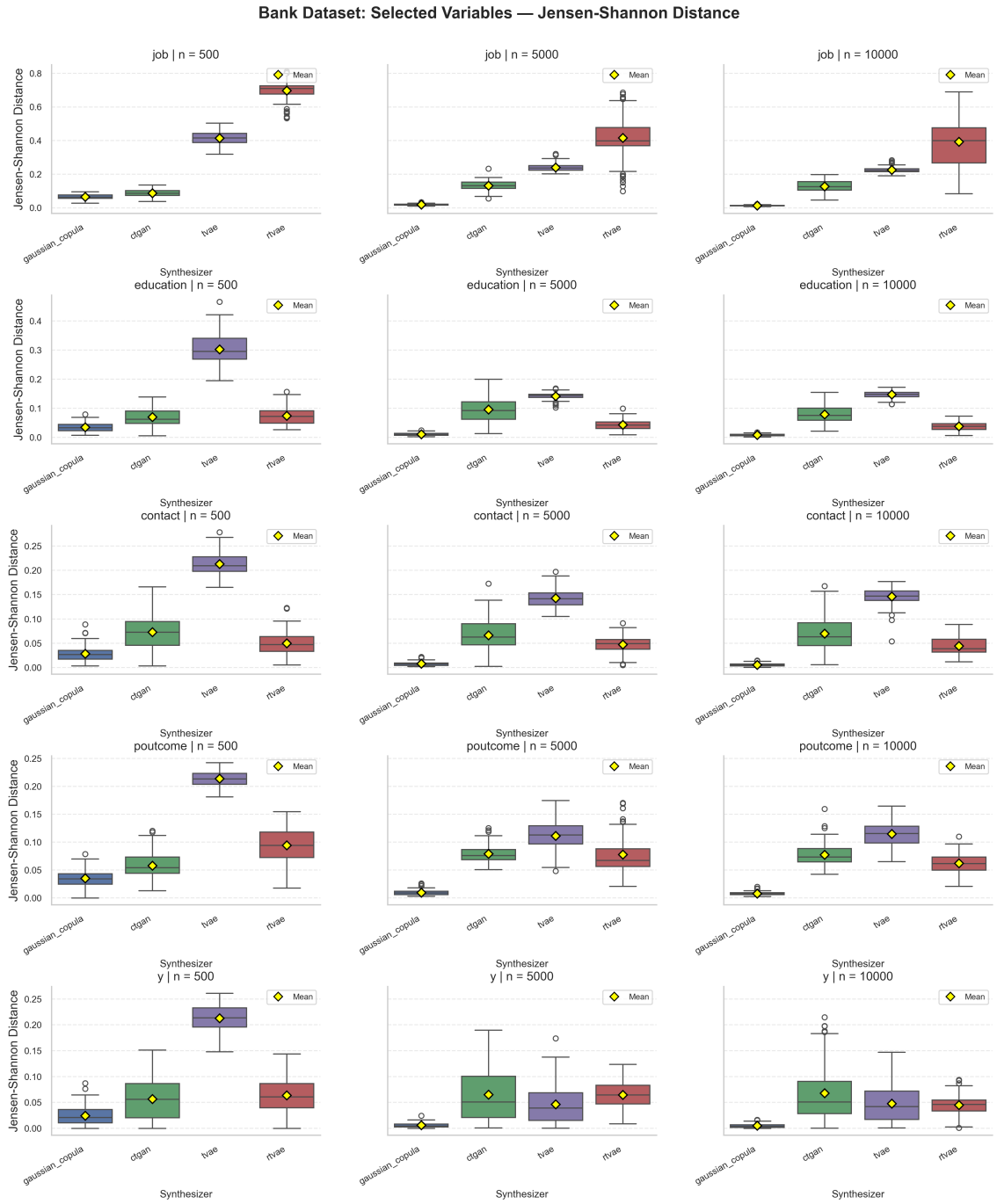

Figure 4.4: Jensen–Shannon distance results for selected variables in the Bank dataset.

CTGAN

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.331	0.066	0.097	0.312	0.152	0.028	0.035	0.149	0.142	0.023	0.032	0.141
ap_hi	0.816	0.023	0.028	0.814	0.660	0.121	0.219	0.653	0.683	0.129	0.043	0.748
cardio	0.031	0.023	0.028	0.028	0.046	0.030	0.044	0.038	0.054	0.034	0.057	0.053
cholesterol	0.058	0.021	0.025	0.057	0.077	0.038	0.057	0.073	0.072	0.041	0.063	0.070
gluc	0.058	0.017	0.025	0.058	0.079	0.038	0.060	0.076	0.078	0.042	0.065	0.076

Gaussian Copula

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.150	0.023	0.028	0.147	0.109	0.015	0.014	0.107	0.108	0.018	0.013	0.105
ap_hi	0.812	0.032	0.021	0.806	0.795	0.029	0.039	0.789	0.792	0.019	0.027	0.792
cardio	0.015	0.011	0.015	0.013	0.003	0.002	0.004	0.003	0.003	0.002	0.003	0.003
cholesterol	0.021	0.011	0.018	0.019	0.007	0.004	0.005	0.006	0.005	0.003	0.004	0.005
gluc	0.021	0.012	0.017	0.020	0.007	0.004	0.004	0.007	0.005	0.003	0.003	0.005

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.9: Hellinger distance summary for selected Cardio variables using CTGAN and Gaussian Copula

CTGAN

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.351	0.070	0.101	0.332	0.174	0.033	0.040	0.170	0.161	0.027	0.037	0.161
ap_hi	0.857	0.021	0.024	0.858	0.710	0.135	0.249	0.699	0.746	0.144	0.048	0.818
cardio	0.037	0.027	0.034	0.034	0.055	0.036	0.052	0.046	0.065	0.041	0.068	0.063
cholesterol	0.070	0.025	0.030	0.069	0.092	0.046	0.069	0.087	0.087	0.050	0.076	0.084
gluc	0.070	0.020	0.030	0.069	0.095	0.046	0.071	0.091	0.094	0.050	0.078	0.091

Gaussian Copula

Variable	500				5K				10K			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.179	0.028	0.033	0.176	0.130	0.017	0.017	0.128	0.129	0.020	0.016	0.125
ap_hi	0.857	0.027	0.019	0.853	0.855	0.022	0.030	0.851	0.857	0.015	0.021	0.857
cardio	0.018	0.013	0.018	0.015	0.004	0.003	0.004	0.004	0.004	0.002	0.003	0.004
cholesterol	0.026	0.013	0.022	0.023	0.008	0.004	0.006	0.007	0.006	0.003	0.004	0.006
gluc	0.025	0.014	0.020	0.024	0.009	0.004	0.004	0.008	0.007	0.003	0.003	0.006

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.10: Jensen–Shannon distance summary for selected Cardio variables using CTGAN and Gaussian Copula

distance falls from 0.302 to 0.218, and then remains almost unchanged at 0.217 for n = 10000. Jensen Shannon Distance follows the same pattern, moving from 0.324 to 0.243 and then staying at 0.243. It improves clearly for `cholesterol`, where the mean distance decreases from 0.276 at $n = 500$ to 0.047 at $n = 10000$. The plots Figure 4.5 and 4.6 show that TVAE still has wide variation for `ap_hi`, especially at $n = 5000$ and $n = 10000$.

TVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.241	0.048	0.065	0.236	0.211	0.048	0.061	0.208	0.223	0.060	0.078	0.213
ap_hi	0.715	0.057	0.054	0.708	0.588	0.183	0.340	0.593	0.608	0.157	0.185	0.653
cardio	0.019	0.014	0.019	0.016	0.020	0.014	0.019	0.017	0.035	0.019	0.029	0.036
cholesterol	0.276	0.038	0.057	0.279	0.103	0.055	0.080	0.096	0.047	0.039	0.021	0.034
gluc	0.258	0.022	0.028	0.257	0.166	0.025	0.026	0.167	0.174	0.028	0.035	0.182

RTVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.793	0.020	0.026	0.794	0.692	0.100	0.019	0.718	0.664	0.102	0.023	0.686
ap_hi	0.743	0.102	0.091	0.738	0.842	0.142	0.292	0.895	0.874	0.124	0.225	0.950
cardio	0.029	0.020	0.026	0.025	0.036	0.021	0.033	0.038	0.022	0.013	0.020	0.022
cholesterol	0.073	0.028	0.041	0.067	0.039	0.023	0.036	0.035	0.038	0.021	0.023	0.037
gluc	0.089	0.029	0.043	0.087	0.045	0.026	0.038	0.041	0.031	0.019	0.020	0.026

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.11: Hellinger distance summary for selected Cardio variables using TVAE and RTVAE

RTVAE shows the weakest performance for `age` and `ap_hi`. RTVAE has the highest overall distances in Cardio. Its average Hellinger distance remains high, moving only slightly from 0.345 to 0.331 and 0.326. Jensen Shannon Distance also remains high, decreasing only from 0.362 to 0.346 and 0.339. For `ap_hi`, the mean distance increases from 0.743 to 0.874, which shows that RTVAE fails to preserve this numerical distribution. The boxplots 4.5 and 4.6, show that `ap_hi` and age create the largest distance values, while cardio, cholesterol, and gluc are comparatively easier.

Cardio is the most difficult dataset among the three, mainly because some selected

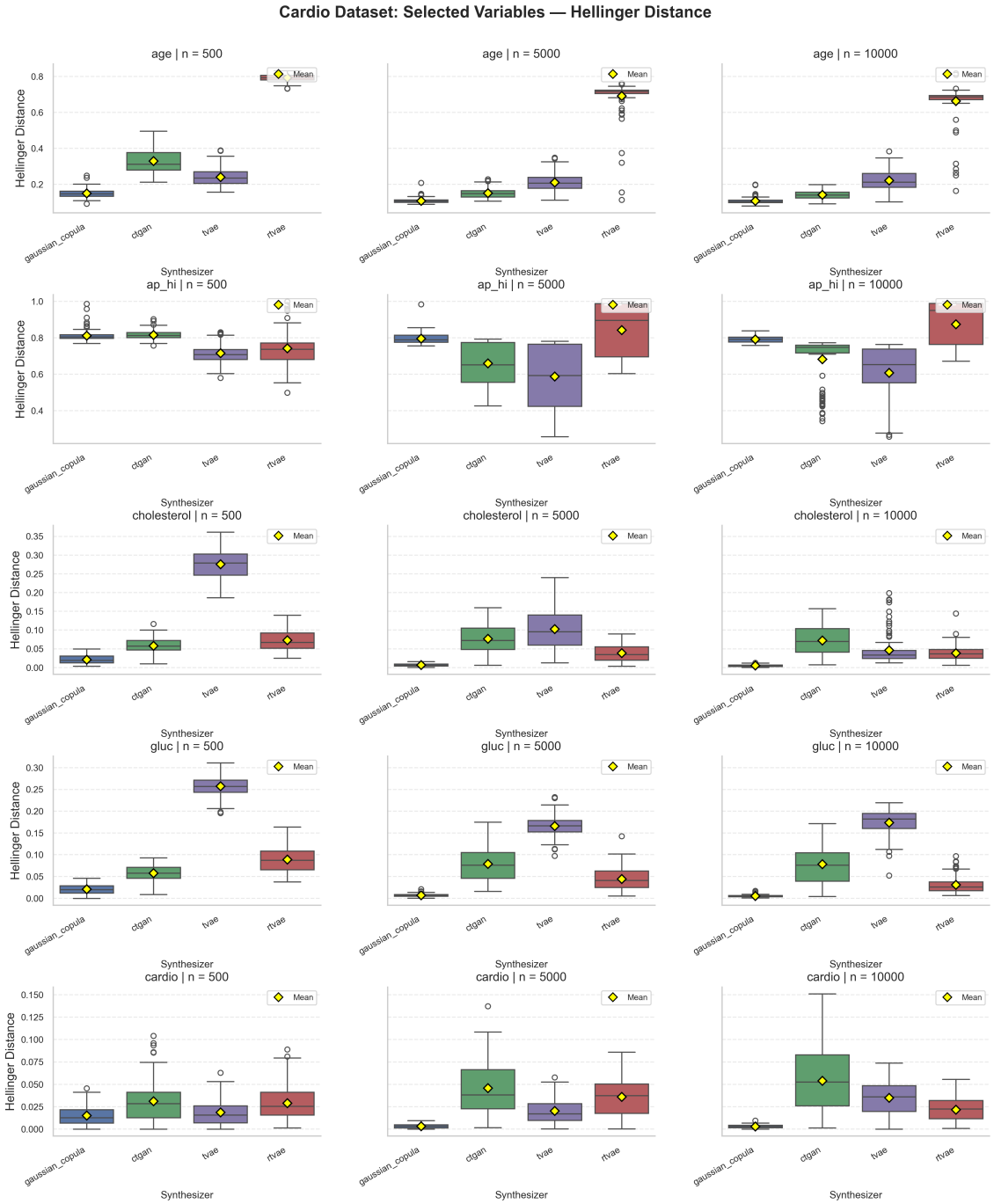

Figure 4.5: Hellinger distance results for selected variables in the Cardio dataset.

variables have more complex numerical or clinical distributions.

An additional it can be observed that run-to-run variability and fidelity do not always move together. Gaussian Copula gives narrow boxplots, but for `ap_hi` the distance remains high, which means that the model has low run-to-run dispersion across runs but still does not approximate this variable well. The variability pattern is also different from Adult and Bank, as the neural models do not become clearly more stable with larger sample sizes. For example, the average Hellinger standard deviation increases from about 0.030 to 0.054 for CTGAN, from 0.035 to 0.061 for TVAE, and from 0.040 to 0.056 for RTVAE between $n = 500$ and $n = 10000$. This suggests that the difficulty in Cardio is not only due to limited sample size, but also due to the structure of variables such as `ap_hi` and `age`. On the contrary, `cardio`, `cholesterol`, and `gluc` are comparatively more stable because they are binary or have fewer discrete levels.

TVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.275	0.055	0.077	0.271	0.247	0.057	0.070	0.242	0.262	0.071	0.090	0.251
ap_hi	0.746	0.061	0.053	0.736	0.635	0.201	0.375	0.630	0.665	0.173	0.203	0.716
cardio	0.023	0.016	0.023	0.019	0.024	0.017	0.022	0.020	0.042	0.023	0.035	0.043
cholesterol	0.306	0.032	0.046	0.310	0.121	0.062	0.093	0.114	0.056	0.045	0.026	0.040
gluc	0.268	0.018	0.025	0.267	0.186	0.024	0.024	0.185	0.189	0.024	0.022	0.196

RTVAE

Variable	**500**				**5K**				**10K**			
	Mean	SD	IQR	Median	Mean	SD	IQR	Median	Mean	SD	IQR	Median
age	0.837	0.021	0.027	0.837	0.730	0.100	0.019	0.754	0.697	0.102	0.025	0.718
ap_hi	0.746	0.102	0.088	0.741	0.855	0.143	0.295	0.927	0.887	0.123	0.223	0.962
cardio	0.035	0.024	0.031	0.031	0.043	0.025	0.039	0.045	0.026	0.016	0.024	0.027
cholesterol	0.087	0.033	0.049	0.080	0.046	0.027	0.043	0.042	0.046	0.024	0.028	0.044
gluc	0.105	0.033	0.051	0.104	0.053	0.030	0.045	0.049	0.037	0.023	0.024	0.031

Note. Values are reported across repeated 100 runs. SD denotes standard deviation, and IQR denotes interquartile range.

Table 4.12: Jensen–Shannon distance summary for selected Cardio variables using TVAE and RTVAE

For a more detailed variable-level view of the boxplot distributions for the Cardio dataset, including all variables can be referred to Appendix A, Figures A.5 and A.6.

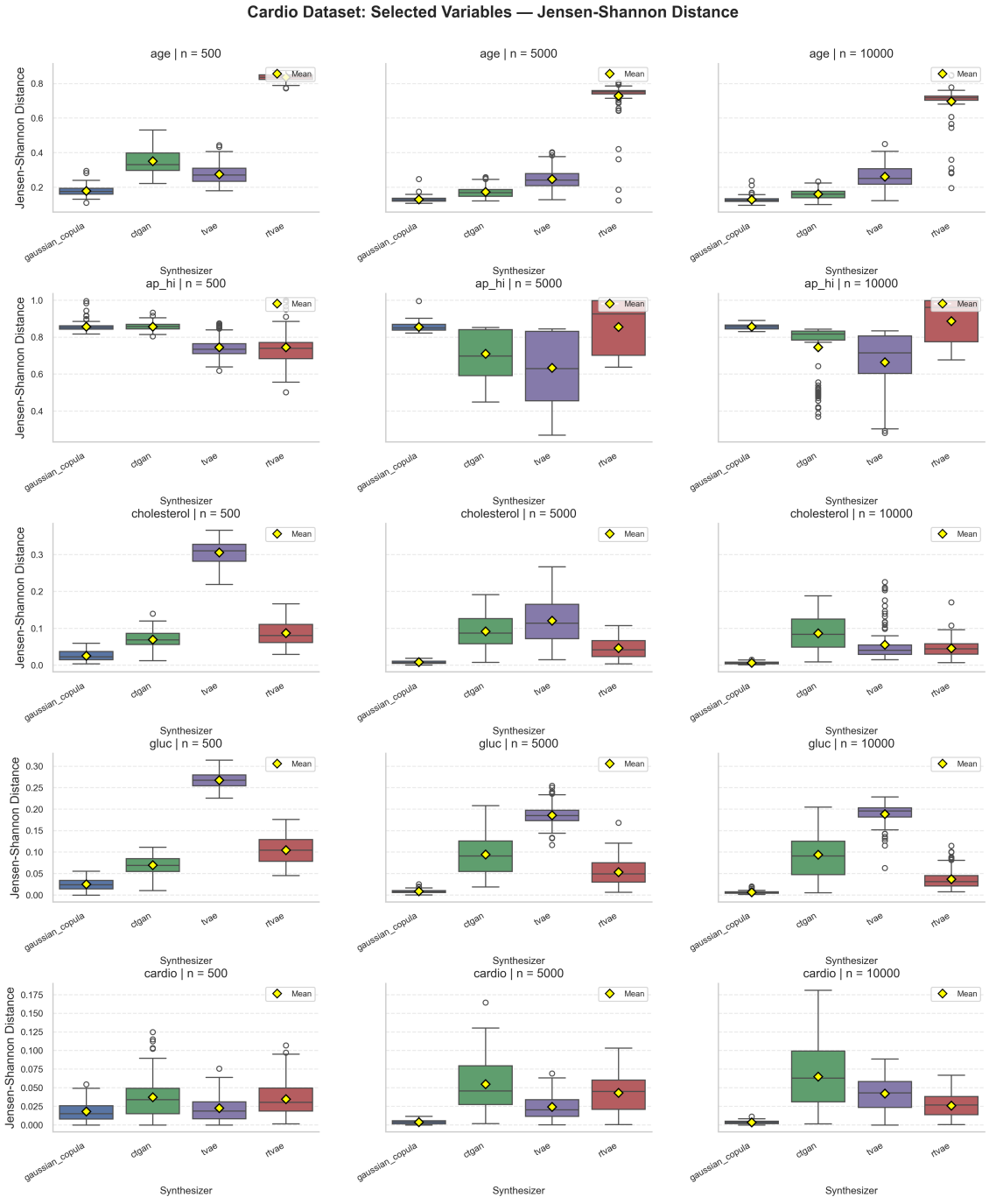

Figure 4.6: Jensen–Shannon distance results for selected variables in the Cardio dataset.

4.2 Discussion on Non-DP Synthesizers

Across the three datasets, Gaussian Copula showed the lowest run-to-run dispersion overall. It produced the lowest average standard deviation across almost every dataset and metric, indicating that its fidelity scores were less variable across repeated runs than those of the neural-network-based models. This was also visible in the boxplots, where Gaussian Copula usually had narrower boxes and smaller interquartile ranges. For the Adult dataset, its average Hellinger and Jensen Shannon standard deviations were about 0.006 and 0.007, compared with 0.020 and 0.024 for CTGAN, 0.026 and 0.028 for TVAE, and 0.062 and 0.066 for RTVAE. A similar pattern appeared in the Bank dataset, where Gaussian Copula had standard deviations of about 0.007 for Hellinger Distance and 0.008 for Jensen Shannon Distance. In Cardio, the variability was higher for all models, but Gaussian Copula still remained the most stable, with standard deviations of about 0.012 and 0.013 for the two metrics.

The broad run-to-run dispersion ranking was Gaussian Copula first, followed by CTGAN and TVAE, while RTVAE was usually the least stable, with some dataset-specific differences. In terms of average distance values, Gaussian Copula also gave the strongest overall fidelity for the Adult and Bank datasets. For example, in Adult, its average Hellinger Distance was about 0.023, compared with 0.074 for CTGAN, 0.160 for TVAE, and 0.268 for RTVAE. In Bank, Gaussian Copula again had the lowest average Hellinger distance, at about 0.016. The Cardio dataset behaved differently because some variables, especially `ap_hi`, were difficult for all synthesizers. As a result, Gaussian Copula was still stable in Cardio, but its average distance values were higher than in Adult and Bank.

When comparing across the sample size, it generally had a positive effect on the results, but this effect was not equal across all synthesizers. For Gaussian Copula, the effect was the clearest. In the Adult and Bank datasets, both the mean distance values and the variability decreased as the sample size increased. This suggested that

Gaussian Copula estimated the marginal distributions and dependence structure more reliably when more data was available. TVAE also benefited from larger sample sizes, especially in the Adult and Bank datasets, where the distance values decreased as the training data increased. This indicated that TVAE learned more stable representations with more data, although the improvement was weaker in the Cardio dataset, particularly for `ap_hi`.

For CTGAN, the sample-size effect was less uniform. The model did not always become more stable when the sample size increased, which may be linked to the instability of adversarial training. This training process is called adversarial training because the generator and discriminator are trained with opposing objectives. The generator attempts to create synthetic records that are difficult to distinguish from real records, while the discriminator attempts to correctly identify whether a record is real or synthetic. During training, both networks are updated iteratively, and the generator improves by learning from the discriminator's feedback. Even with more data, the generator and discriminator could converge differently across repeated runs. RTVAE also showed a mixed response to larger sample sizes. Larger sample sizes reduced the average distance in some cases, but they did not always reduce variability. In the Adult and Bank datasets, RTVAE remained unstable for variables with many categories, while in Cardio it remained difficult for `ap_hi`. This suggests that sample size influenced the results, but the model architecture had a stronger role in determining both fidelity and run-to-run dispersion.

When comparing both the Hellinger and Jensen Shannon metrics, both led to nearly same overall conclusions. The ranking of synthesizers remained mostly unchanged across Hellinger distance and Jensen Shannon distance. Gaussian Copula showed the lowest run-to-run dispersion across most settings, while the neural-network-based models showed higher variability. This indicated that the main findings were not dependent on only one metric, since both measures captured similar

differences between the synthesizers across datasets, variables, and sample sizes.

However, Jensen Shannon distance was slightly more variable than Hellinger distance in most cases. Across all datasets, models, variables, and sample sizes, Jensen Shannon distance had a higher mean value in all 180 comparisons. It also had a higher standard deviation in 163 out of 180 comparisons and a higher interquartile range in 156 out of 180 comparisons. This suggested that Jensen Shannon distance was slightly more sensitive to run-to-run changes in the generated distributions. The difference was small for stable models such as Gaussian Copula, but it became clearer in unstable settings, such as RTVAE on Adult `education`, RTVAE on Bank `job`, and TVAE or RTVAE on Cardio `ap_hi`.

The architecture of each synthesizer helps explain these differences in fidelity and stability. Gaussian Copula is a statistical model that directly estimated marginal distributions and the dependency structure between variables [9]. Since it did not rely on adversarial training or deep latent optimisation, it was less sensitive to random initialisation. This explained why it produced narrow boxplots and low standard deviations across most datasets. However, its limitation became visible when the variable distribution was more complex, irregular, or clinical in nature, for example `ap_hi` in the Cardio dataset. CTGAN, in contrast, was based on adversarial training. This made it more flexible for complex tabular data, but it also introduced more instability across repeated runs. The generator and discriminator could learn differently each time, which could explain why CTGAN often showed moderate variability and why larger sample size did not always reduce the spread.

TVAE used a variational autoencoder structure, where the model learned a latent representation and reconstructed samples from that latent space. This helped when more data was available, which explained why TVAE improved in the Adult and Bank datasets as the sample size increased. However, the latent representation could smooth or blur category boundaries, especially for categorical variables with

many levels. This may have contributed to higher distance values for variables like for example `education`, `occupation`, and `job`. RTVAE also followed a VAE-based structure, but in these results it was the least stable overall. It showed high variability for complex variables, suggesting that its training and sampling process was more sensitive to variable structure. This was visible in Adult for `education` and `occupation`, in Bank for `job`, and in Cardio for `ap_hi`. One possible reason is that RTVAE's objective may reduce the influence of unusual observations. This can be useful when those observations are noise, but it can be harmful when rare categories or irregular values are valid parts of the real data. Overall, the results showed that model architecture affected both optimisation stability and the ability to handle rare categories or irregular numerical distributions.

When we consider patterns across datasets, three main pattersn can be drawn from the Non-DP results. First, Gaussian Copula had the lowest run-to-run dispersion across most of the settings. Even when its fidelity was weaker for difficult variables such as `ap_hi` in the Cardio dataset, its run-to-run variability remained very low. This showed that a model could be stable across repeated runs while still having higher distance values for variables with complex distributions. Second, variables with simpler distributions were easier to model. Binary or low-cardinality variables such as Adult `income`, Bank `y`, and Cardio `cardio` generally showed lower variability.

In contrast, variables with more categories, like for example Adult `occupation` and Bank `job`, produced higher variability because their category distributions were harder to reproduce with low variability. The Cardio dataset also showed that clinical numerical variables were more difficult than standard categorical variables. The clearest example was `ap_hi`, which produced high distance values and high variability across several models. Finally, Jensen Shannon Distance was slightly more sensitive than Hellinger Distance, but both metrics supported the same main conclusion. Therefore, the observed patterns were not caused by one metric only,

but were consistent across both evaluation measures.

4.3 DP Synthesizers

This section presents the selected-variable results for the DP synthesizers, AIM and PrivBayes. The results are grouped by privacy budget so that the effect of ϵ can be compared more clearly across datasets and distance metrics.

4.3.1 Differential Privacy Budget $\epsilon = 1$

Adult Dataset

From Figure 4.7 and Table 4.13, AIM provides stronger univariate fidelity than PrivBayes for most selected variables, especially at larger sample sizes. At $n = 500$, both synthesizers show relatively high distance values for variables with larger category spaces, such as `workclass`, `education`, and `occupation`. However, AIM improves sharply when the sample size increases to $n = 5000$. For example, the mean Hellinger distance for AIM decreases from 0.3486 to 0.0568 for `education`, and from 0.2942 to 0.0463 for `occupation`. This indicates that AIM becomes substantially more accurate and less variable across runs when more training records are available, even under a strict privacy budget.

PrivBayes also improves with increasing sample size, but its fidelity scores remain higher and more variable than AIM for most variables. This is particularly visible for `workclass`, where the mean Hellinger distance for PrivBayes remains 0.2939 at $n = 5000$ and decreases to 0.1153 at $n = 10000$, while AIM reaches 0.0500 and 0.0482 for the same sample sizes. The box plots also show wider dispersion for PrivBayes, especially at $n = 500$ and $n = 5000$, indicating less stable fidelity measurements across repeated runs. The binary variable `income` is reproduced most accurately by AIM, with the mean distance decreasing to 0.0043 at $n = 5000$ and 0.0034 at

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.268	0.275	0.127	0.094	0.050	0.049	0.019	0.015	0.048	0.049	0.018	0.012
education	0.349	0.321	0.195	0.124	0.057	0.054	0.017	0.014	0.037	0.037	0.015	0.010
relationship	0.131	0.113	0.085	0.081	0.027	0.021	0.016	0.020	0.048	0.053	0.026	0.018
occupation	0.294	0.258	0.187	0.137	0.046	0.044	0.016	0.014	0.038	0.038	0.010	0.009
income	0.029	0.024	0.032	0.021	0.004	0.003	0.004	0.003	0.003	0.003	0.004	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.326	0.399	0.264	0.151	0.294	0.320	0.284	0.156	0.115	0.112	0.089	0.064
education	0.331	0.394	0.179	0.117	0.193	0.182	0.286	0.142	0.074	0.068	0.057	0.048
relationship	0.205	0.254	0.166	0.096	0.132	0.142	0.218	0.104	0.036	0.027	0.037	0.029
occupation	0.206	0.231	0.089	0.067	0.144	0.150	0.176	0.088	0.068	0.063	0.052	0.038
income	0.131	0.168	0.132	0.072	0.081	0.055	0.146	0.072	0.016	0.009	0.018	0.017

Table 4.13: Hellinger distance results for selected Adult with Epsilon 1

$n = 10000$. This reflects the simpler two-category structure of the variable. In contrast, variables such as `education` and `occupation` remain more challenging, although AIM still provides comparatively low and stable distance values at larger sample sizes.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.290	0.301	0.132	0.098	0.057	0.055	0.020	0.017	0.055	0.056	0.019	0.013
education	0.377	0.347	0.224	0.131	0.067	0.065	0.018	0.015	0.044	0.044	0.017	0.012
relationship	0.149	0.133	0.097	0.085	0.032	0.025	0.019	0.024	0.057	0.064	0.031	0.021
occupation	0.323	0.284	0.175	0.142	0.054	0.052	0.019	0.016	0.044	0.044	0.012	0.010
income	0.034	0.029	0.039	0.025	0.005	0.004	0.005	0.004	0.004	0.003	0.005	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.381	0.468	0.304	0.174	0.333	0.366	0.320	0.177	0.129	0.126	0.097	0.071
education	0.388	0.463	0.207	0.136	0.226	0.215	0.333	0.166	0.087	0.080	0.067	0.057
relationship	0.243	0.302	0.195	0.113	0.156	0.169	0.258	0.123	0.044	0.032	0.045	0.035
occupation	0.241	0.271	0.106	0.077	0.165	0.173	0.199	0.100	0.078	0.072	0.059	0.042
income	0.156	0.201	0.158	0.086	0.097	0.066	0.175	0.086	0.020	0.011	0.022	0.021

Table 4.14: Jensen–Shannon distance results for selected Adult with Epsilon 1

For Jensen Shannon results from Figure 4.8 and Table 4.14, AIM again achieves

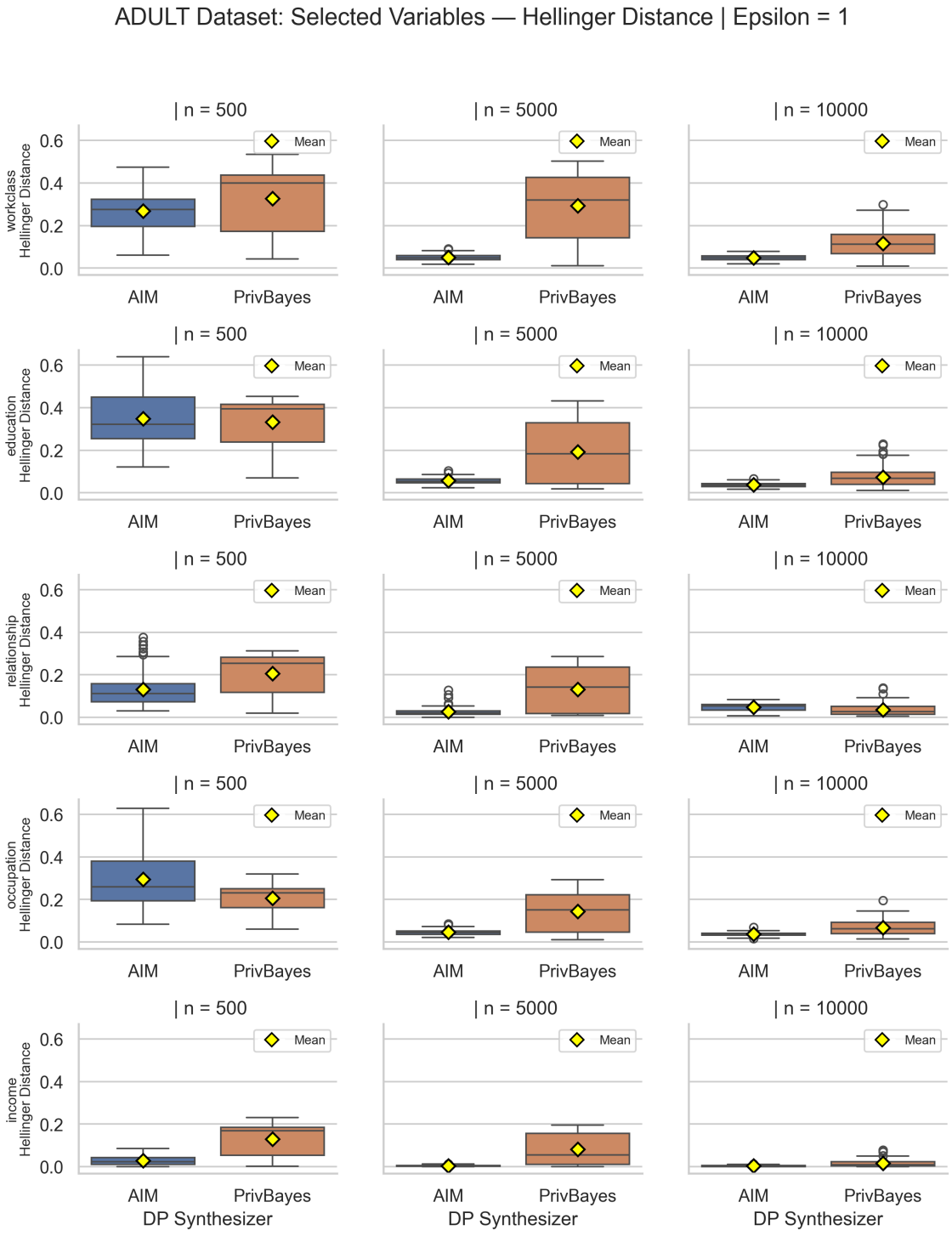

Figure 4.7: Hellinger distance results for selected variables in the Adult dataset with $\epsilon = 1$.

lower average distance values and narrower boxplots than PrivBayes for most variables at $n = 5000$ and $n = 10000$. For example, the mean Jensen Shannon distance for AIM on `education` decreases from 0.3774 at $n = 500$ to 0.0670 at $n = 5000$ and 0.0444 at $n = 10000$. For `occupation`, the corresponding values decrease from 0.3227 to 0.0539 and 0.0435. These reductions show that AIM produces more stable fidelity scores as the sample size increases.

PrivBayes shows the same general direction of improvement, but with higher distance values and greater run-to-run variability. At $n = 5000$, its Jensen Shannon distances remain high for several variables, including 0.3331 for `workclass`, 0.2262 for `education`, and 0.1649 for `occupation`. The large interquartile ranges for these variables indicate that repeated runs do not produce equally reproducible fidelity scores. Overall, both metrics shows that AIM provides more reliable preservation of the Adult marginal distributions, while PrivBayes is more sensitive to sample size and shows higher variability.

Bank Dataset

For Hellinger Distance from Figure 4.9 and Table 4.15, both AIM and PrivBayes show higher fidelity distances at $n = 500$, particularly for variables with more complex or imbalanced distributions. However, AIM improves substantially when the sample size increases. For example, the mean Hellinger distance for AIM decreases from 0.2651 to 0.0367 for `job`, from 0.1360 to 0.0126 for `education`, and from 0.1429 to 0.0193 for `poutcome` when the sample size increases from $n = 500$ to $n = 5000$.

PrivBayes follows the same general direction but requires a larger sample size to achieve comparable fidelity. At $n = 5000$, PrivBayes still shows relatively high mean Hellinger distances for several variables, including 0.1576 for `job`, 0.1364 for `contact`, and 0.1459 for `poutcome`. The wide interquartile ranges at this sample size indicate that repeated runs produce more variable fidelity scores. By $n = 10000$, the

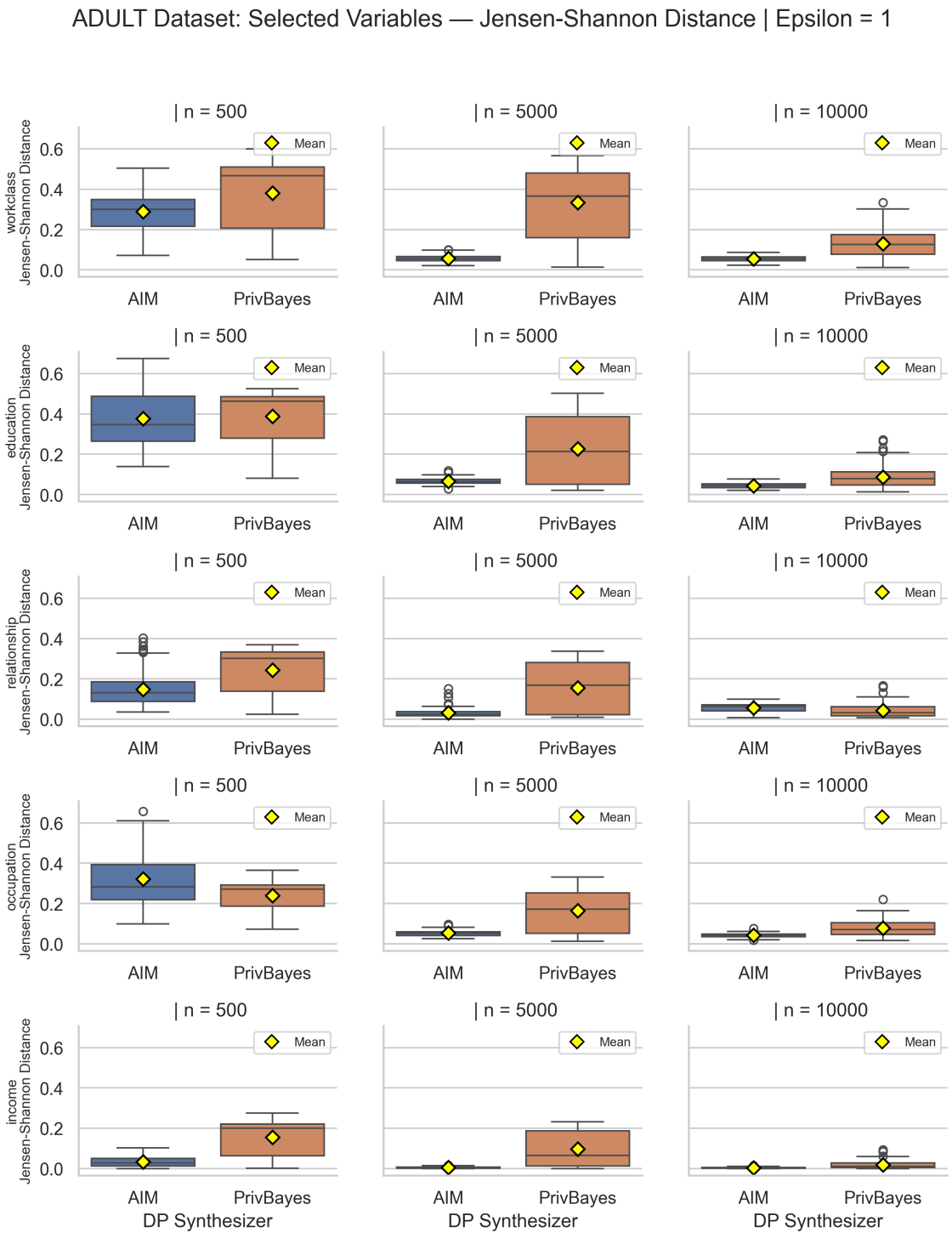

Figure 4.8: Jensen–Shannon distance results for selected variables in the Adult dataset with $\epsilon = 1$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
job	0.265	0.226	0.211	0.125	0.037	0.036	0.015	0.012	0.029	0.026	0.017	0.011
education	0.136	0.119	0.093	0.080	0.013	0.012	0.007	0.006	0.008	0.007	0.005	0.004
contact	0.106	0.085	0.108	0.064	0.009	0.008	0.007	0.006	0.007	0.006	0.005	0.005
poutcome	0.143	0.135	0.105	0.073	0.019	0.016	0.010	0.014	0.018	0.014	0.013	0.017
y	0.048	0.039	0.049	0.042	0.007	0.005	0.006	0.005	0.005	0.004	0.005	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
job	0.243	0.293	0.124	0.091	0.158	0.181	0.203	0.100	0.028	0.024	0.014	0.015
education	0.210	0.248	0.109	0.084	0.092	0.053	0.149	0.092	0.015	0.010	0.009	0.017
contact	0.200	0.248	0.166	0.097	0.136	0.159	0.212	0.101	0.013	0.010	0.011	0.010
poutcome	0.289	0.379	0.285	0.160	0.146	0.080	0.282	0.151	0.017	0.010	0.013	0.017
y	0.183	0.257	0.269	0.131	0.067	0.010	0.142	0.093	0.009	0.006	0.007	0.014

Table 4.15: Hellinger distance results for selected Bank with Epsilon 1

differences between AIM and PrivBayes become much smaller, with both synthesizers producing low average distances for most variables. This improvement is especially visible for `y`, where the binary structure of the target variable makes the marginal distribution easier to reproduce.

From Figure 4.10 and Table 4.16, the results are consistent with the Hellinger distance analysis. For instance, the mean Jensen Shannon distance for AIM decreases from 0.2916 to 0.0439 for `job`, from 0.1535 to 0.0152 for `education`, and from 0.1199 to 0.0109 for `contact` when moving from $n = 500$ to $n = 5000$. At $n = 10000$, AIM reaches very low values across all selected variables, with particularly small distances for `education`, `contact`, and `y`.

PrivBayes shows higher Jensen Shannon distances and larger run-to-run variation at $n = 500$ and $n = 5000$. This is most evident for `poutcome` and `y` variables, where the interquartile ranges remain large at $n = 5000$, suggesting highly variable fidelity measurements across repeated runs. These variables are affected by distributional imbalance: `poutcome` is dominated by the `unknown` category, while `y` is an imbalanced binary target variable. At $n = 10000$, PrivBayes improves strongly and produces

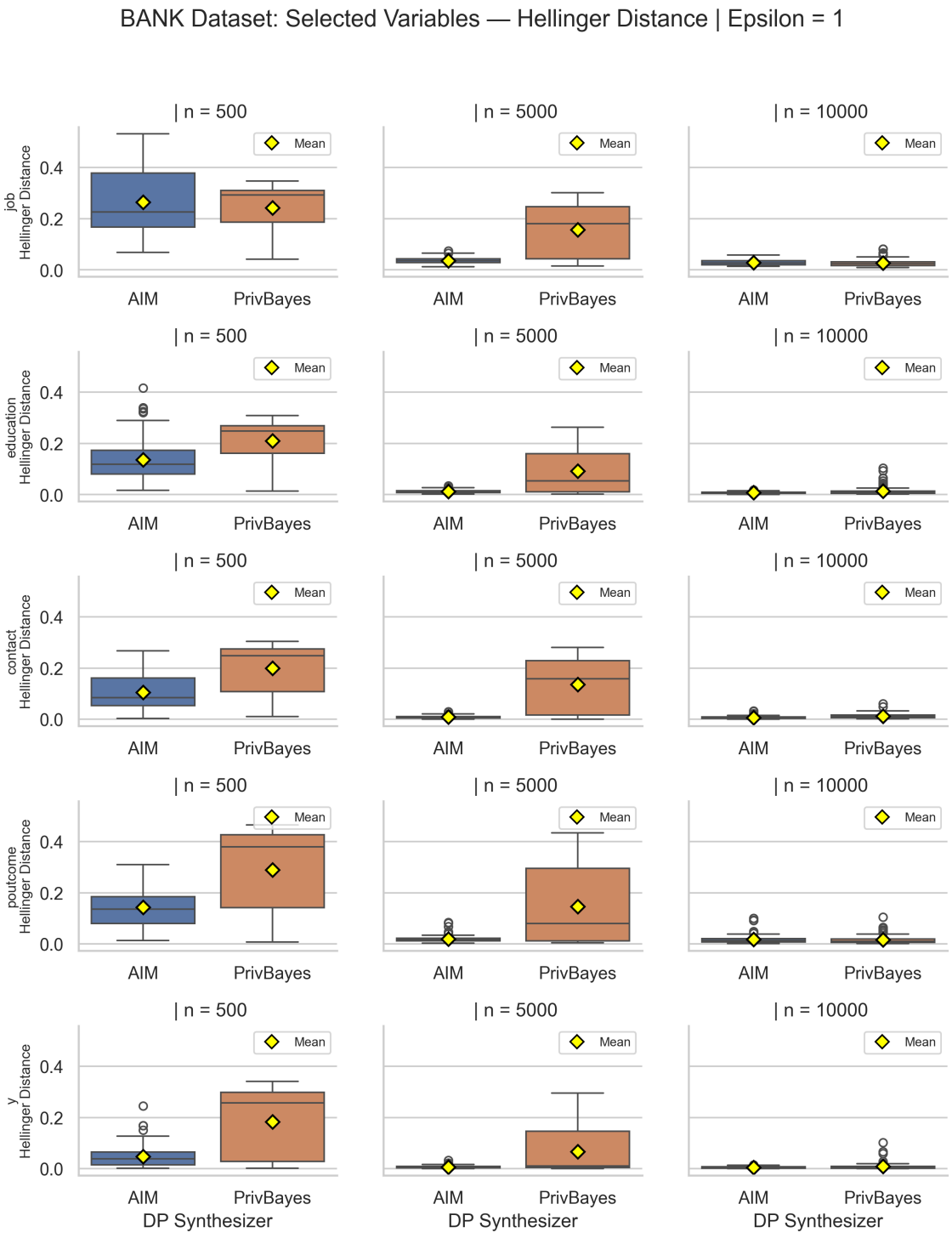

Figure 4.9: Hellinger distance results for selected variables in the Bank dataset with $\epsilon = 1$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
job	0.292	0.252	0.224	0.130	0.044	0.043	0.018	0.014	0.035	0.032	0.020	0.013
education	0.154	0.139	0.101	0.084	0.015	0.014	0.009	0.007	0.009	0.008	0.006	0.005
contact	0.120	0.102	0.118	0.068	0.011	0.010	0.008	0.007	0.009	0.007	0.006	0.006
poutcome	0.158	0.152	0.101	0.074	0.023	0.020	0.012	0.016	0.022	0.017	0.016	0.021
y	0.057	0.046	0.059	0.048	0.008	0.006	0.007	0.006	0.006	0.005	0.006	0.004

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
job	0.286	0.344	0.145	0.106	0.187	0.215	0.239	0.118	0.033	0.029	0.017	0.018
education	0.248	0.292	0.127	0.098	0.109	0.063	0.177	0.108	0.018	0.012	0.011	0.020
contact	0.237	0.293	0.195	0.115	0.162	0.189	0.251	0.119	0.016	0.012	0.013	0.012
poutcome	0.341	0.446	0.332	0.187	0.173	0.096	0.334	0.177	0.020	0.012	0.016	0.020
y	0.217	0.305	0.318	0.154	0.080	0.012	0.170	0.110	0.010	0.007	0.008	0.016

Table 4.16: Jensen–Shannon distance results for selected Bank with Epsilon 1

values close to AIM for several variables, especially `job` and `poutcome`. AIM reaches stable fidelity earlier, while PrivBayes shows greater sensitivity to sample size and produces more variable fidelity scores at smaller training sizes.

Cardio Dataset

From the Figure 4.11 and Table 4.17, AIM achieves lower distance values and more stable fidelity scores than PrivBayes as the sample size increases. At $n = 500$, both synthesizers show noticeable variability, particularly for `age_bin`, `cholesterol`, and `gluc`. However, AIM improves substantially at $n = 5000$ and $n = 10000$. For example, the mean Hellinger distance for AIM decreases from 0.1076 to 0.0205 for `age_bin`, from 0.0820 to 0.0081 for `cholesterol`, and from 0.0967 to 0.0126 for `gluc` when moving from $n = 500$ to $n = 5000$.

PrivBayes also improves from $n = 500$ to $n = 5000$, but its performance is less uniform at $n = 10000$. For instance, the mean Hellinger distance for `cholesterol` decreases from 0.1128 at $n = 500$ to 0.0546 at $n = 5000$, but increases again to 0.0786 at $n = 10000$. This suggests that increasing the sample size does not always

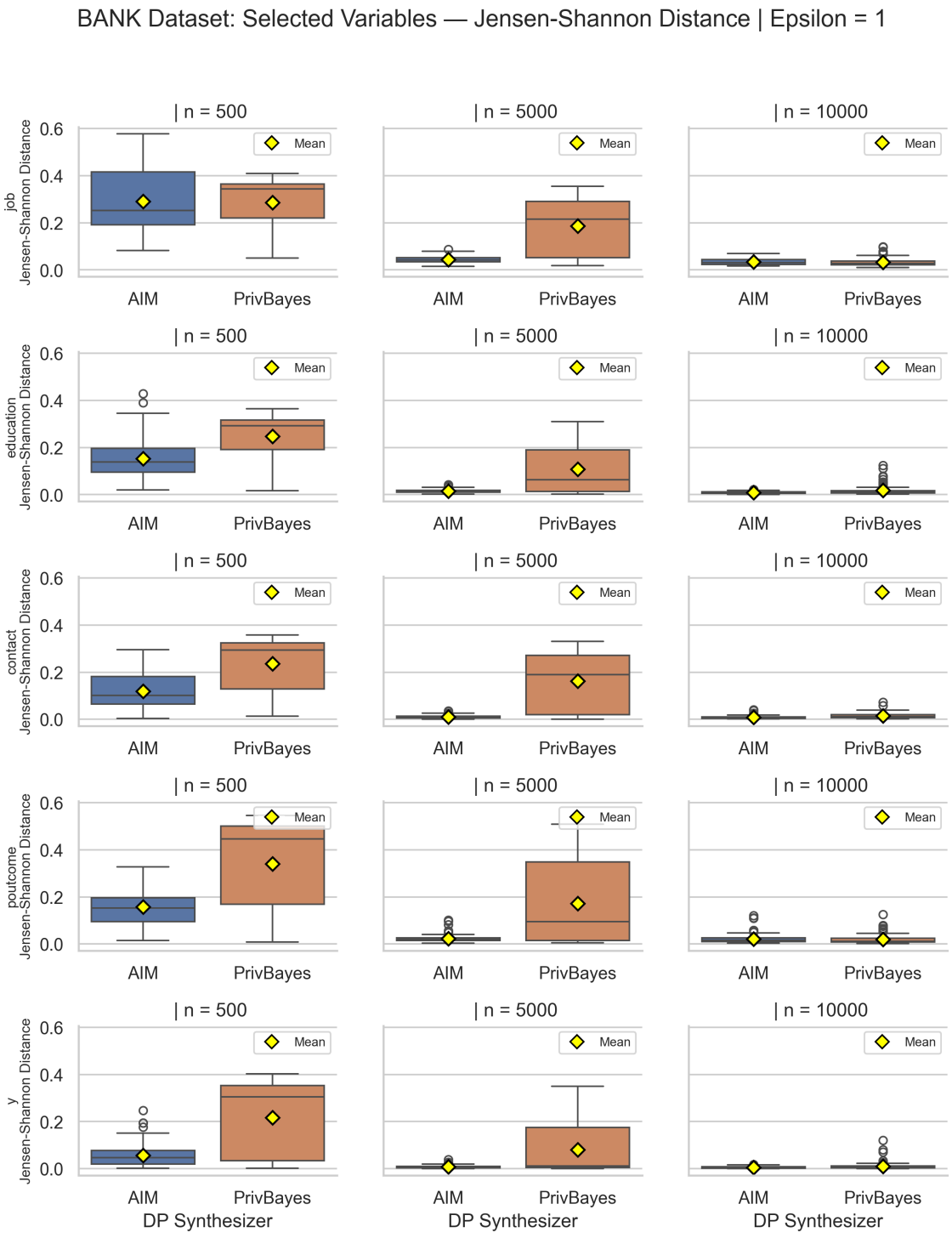

Figure 4.10: Jensen–Shannon distance results for selected variables in the Bank dataset with $\epsilon = 1$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.108	0.083	0.105	0.071	0.021	0.015	0.012	0.015	0.018	0.010	0.021	0.014
cholesterol	0.082	0.065	0.056	0.060	0.008	0.007	0.007	0.005	0.006	0.005	0.005	0.003
gluc	0.097	0.084	0.068	0.062	0.013	0.011	0.011	0.008	0.009	0.008	0.007	0.005
ap_hi_bin	0.064	0.055	0.037	0.043	0.012	0.010	0.007	0.010	0.007	0.005	0.003	0.005
cardio	0.025	0.021	0.019	0.019	0.005	0.004	0.005	0.004	0.003	0.002	0.004	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.121	0.135	0.106	0.064	0.043	0.041	0.016	0.012	0.050	0.050	0.007	0.008
cholesterol	0.113	0.098	0.121	0.072	0.055	0.054	0.014	0.009	0.079	0.078	0.014	0.011
gluc	0.152	0.154	0.180	0.098	0.043	0.043	0.012	0.010	0.065	0.065	0.012	0.009
ap_hi_bin	0.081	0.082	0.077	0.046	0.059	0.058	0.018	0.011	0.081	0.081	0.014	0.011
cardio	0.016	0.013	0.015	0.012	0.028	0.027	0.021	0.013	0.042	0.043	0.020	0.014

Table 4.17: Hellinger distance results for selected Cardio with Epsilon 1

lead to more accurate marginal distribution preservation for PrivBayes under the strict privacy budget. The binary variable `cardio` is reproduced accurately by both synthesizers at $n = 500$, but AIM becomes clearly stronger at larger sample sizes, reaching a mean distance of 0.0029 at $n = 10000$, compared with 0.0419 for PrivBayes.

From the Figure 4.12 and Table 4.18, AIM again shows a clear reduction in average distance values when the sample size increases. For example, the mean Jensen Shannon distance for AIM decreases from 0.1218 to 0.0233 for `age_bin`, and from 0.1096 to 0.0151 for `gluc` between $n = 500$ and $n = 5000$. At $n = 10000$, AIM reaches very low distances for most variables. The small interquartile ranges and standard deviations at larger sample sizes indicate that the fidelity scores remain stable across repeated runs.

PrivBayes shows higher Jensen Shannon distances than AIM for most variables at $n = 5000$ and $n = 10000$. Its results improve from $n = 500$ to $n = 5000$, but several variables show increased distances again at $n = 10000$. For example, the mean distance for `ap_hi_bin` decreases from 0.0971 at $n = 500$ to 0.0712 at $n = 5000$, but

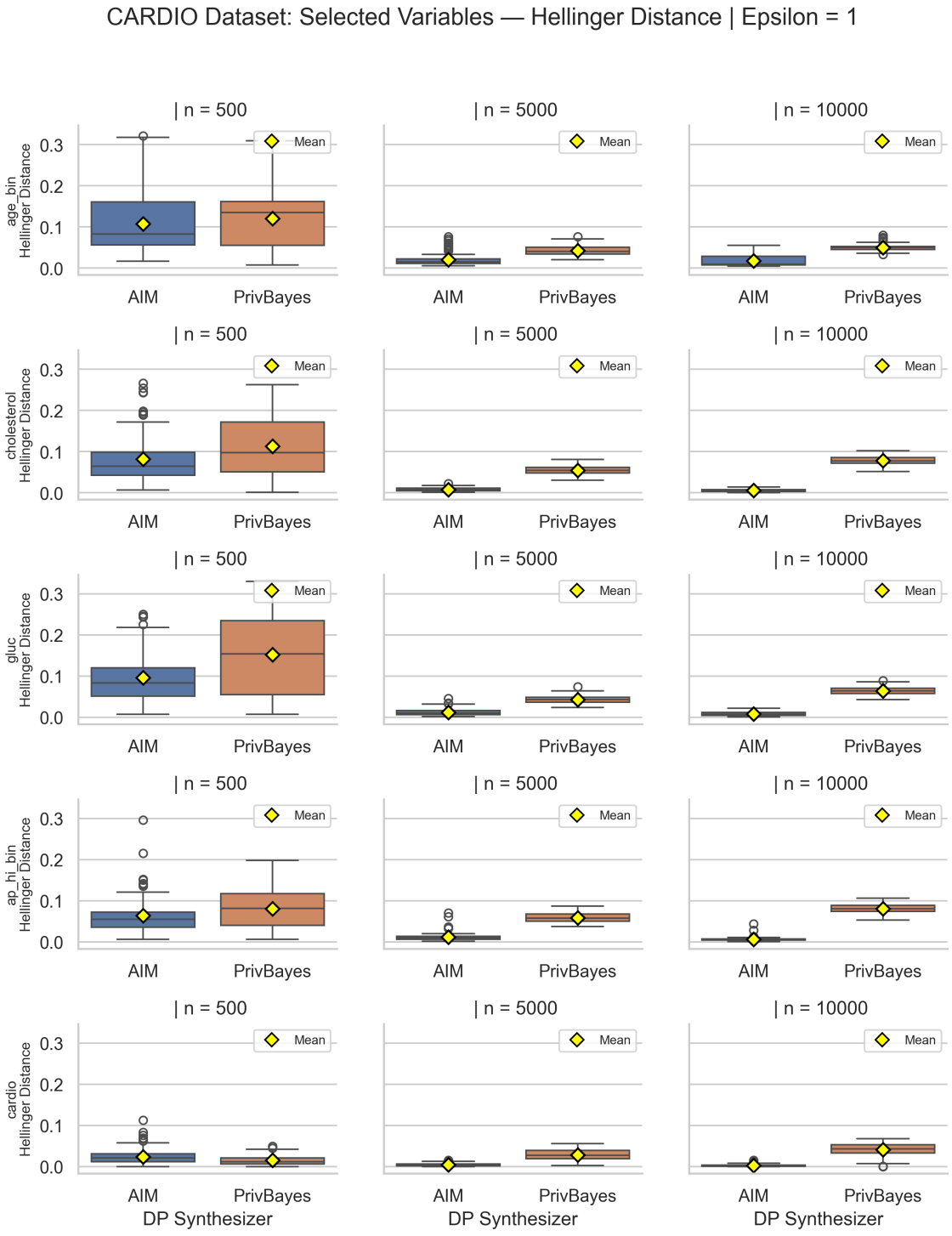

Figure 4.11: Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 1$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.122	0.100	0.096	0.075	0.023	0.018	0.014	0.016	0.020	0.011	0.024	0.015
cholesterol	0.095	0.078	0.067	0.064	0.010	0.009	0.008	0.006	0.007	0.006	0.006	0.004
gluc	0.110	0.100	0.079	0.063	0.015	0.013	0.013	0.010	0.010	0.009	0.008	0.006
ap_hi_bin	0.076	0.067	0.044	0.048	0.014	0.012	0.009	0.012	0.008	0.007	0.003	0.006
cardio	0.029	0.025	0.023	0.023	0.006	0.005	0.006	0.005	0.004	0.003	0.004	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.142	0.161	0.126	0.074	0.051	0.049	0.018	0.013	0.060	0.059	0.009	0.009
cholesterol	0.135	0.117	0.144	0.086	0.066	0.065	0.017	0.011	0.094	0.093	0.017	0.013
gluc	0.181	0.184	0.214	0.116	0.052	0.052	0.015	0.011	0.078	0.078	0.015	0.011
ap_hi_bin	0.097	0.098	0.093	0.056	0.071	0.070	0.022	0.014	0.097	0.097	0.017	0.013
cardio	0.019	0.015	0.018	0.015	0.034	0.033	0.025	0.016	0.050	0.052	0.024	0.017

Table 4.18: Jensen–Shannon distance results for selected Cardio with Epsilon 1

increases to 0.0974 at $n = 10000$. Similarly, `cholesterol`, `gluc`, and `cardio` show higher mean distances at $n = 10000$ than at $n = 5000$. This indicates that PrivBayes does not show the same clear sample-size improvement as AIM under $\epsilon = 1$. Taken together, both metrics show that AIM provides stronger and more stable univariate fidelity for the selected Cardio variables, while PrivBayes is more sensitive to sample size and shows more variable fidelity measurements across repeated runs.

4.3.2 Differential Privacy Budget $\epsilon = 5$

Adult Dataset

For hellinger distance from Figure 4.13 and Table 4.19, AIM achieves lower mean and median distances than PrivBayes for almost all selected variables and sample sizes. This difference is already visible at $n = 500$, where AIM records moderate distances for `workclass`, `education`, and `occupation`, while PrivBayes produces substantially higher values for the same variables. For example, the mean Hellinger distance for `education` is 0.1049 for AIM compared with 0.3035 for PrivBayes, and for `workclass` it is 0.0808 compared with 0.3325. This indicates that AIM preserves

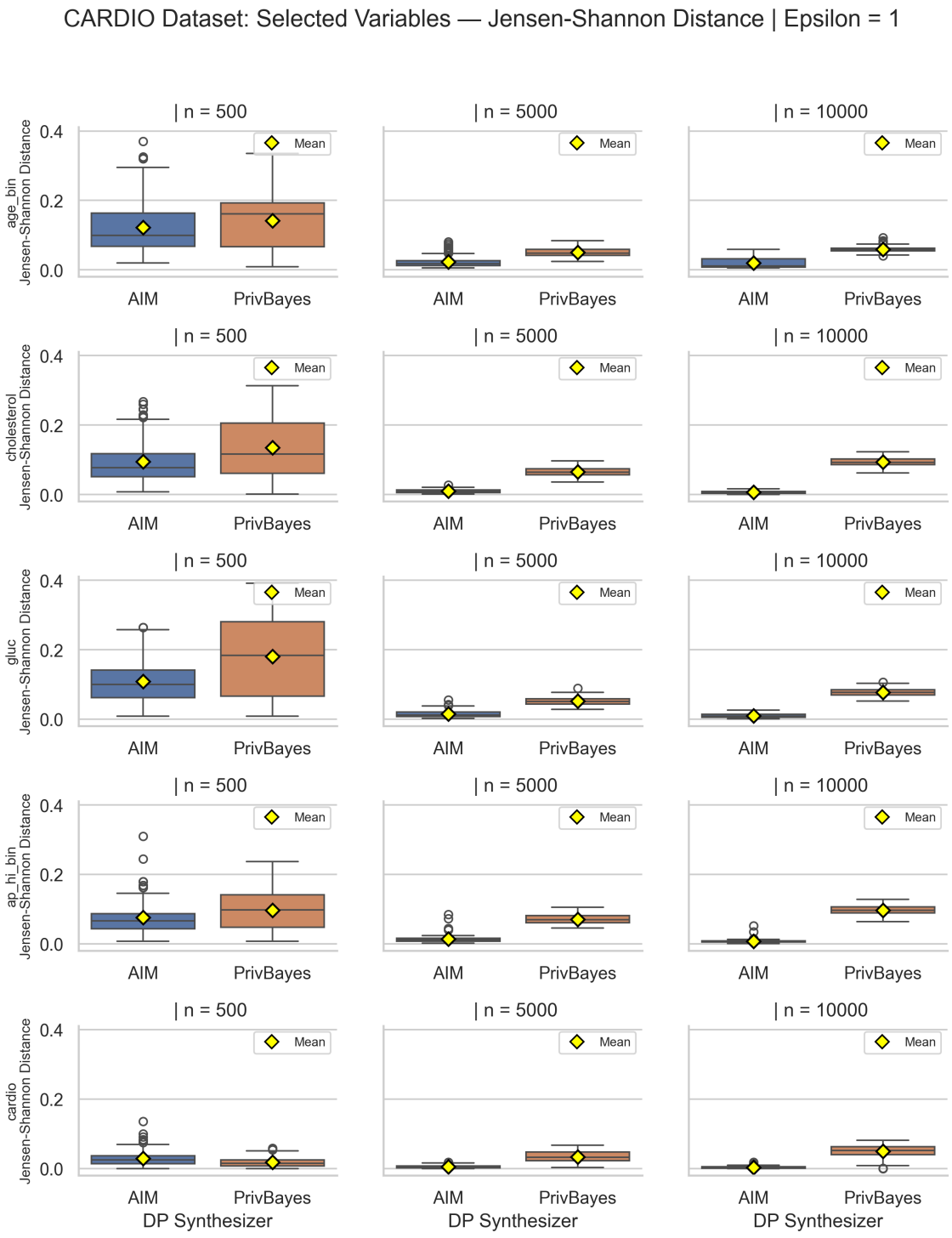

Figure 4.12: Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 1$.

the Adult marginal distributions more effectively under this privacy budget.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.081	0.070	0.036	0.034	0.033	0.033	0.012	0.008	0.023	0.023	0.007	0.005
education	0.105	0.102	0.032	0.028	0.025	0.024	0.006	0.006	0.016	0.016	0.005	0.004
relationship	0.057	0.050	0.029	0.027	0.031	0.028	0.041	0.021	0.028	0.013	0.044	0.023
occupation	0.089	0.086	0.028	0.020	0.028	0.028	0.009	0.007	0.021	0.021	0.009	0.006
income	0.014	0.012	0.012	0.011	0.004	0.003	0.004	0.004	0.003	0.002	0.003	0.004

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.333	0.399	0.241	0.146	0.131	0.128	0.094	0.069	0.110	0.117	0.070	0.052
education	0.304	0.363	0.215	0.127	0.097	0.088	0.078	0.055	0.077	0.069	0.055	0.042
relationship	0.182	0.240	0.211	0.106	0.040	0.027	0.036	0.035	0.027	0.019	0.018	0.026
occupation	0.198	0.212	0.090	0.063	0.067	0.058	0.047	0.034	0.047	0.040	0.027	0.024
income	0.129	0.156	0.112	0.072	0.018	0.013	0.018	0.018	0.011	0.006	0.009	0.014

Table 4.19: Hellinger distance results for selected Adult with Epsilon 5

The effect of sample size is also clear. For AIM, the mean Hellinger distance decreases from 0.0808 to 0.0226 for `workclass`, from 0.1049 to 0.0164 for `education`, and from 0.0893 to 0.0208 for `occupation` when the sample size increases from $n = 500$ to $n = 10000$. The boxplots become narrower at larger sample sizes, showing that AIM produces more stable fidelity scores across repeated runs. PrivBayes also improves as the sample size increases, but its distances remain higher for most variables. At $n = 10000$, the mean Hellinger distance for PrivBayes is still 0.1098 for `workclass`, 0.0769 for `education`, and 0.0473 for `occupation`. The binary variable `income` is reproduced most accurately by both synthesizers, especially AIM, which reaches a mean distance of 0.0032 at $n = 10000$. In contrast, `education`, `occupation`, and `workclass` remain more challenging because they contain larger category spaces and more uneven distributions.

For Jensen Shannon from Figure 4.14 and Table 4.20, the trends are closely aligned with the Hellinger distance results. AIM again produces lower average distances and more compact boxplots than PrivBayes for most selected variables.

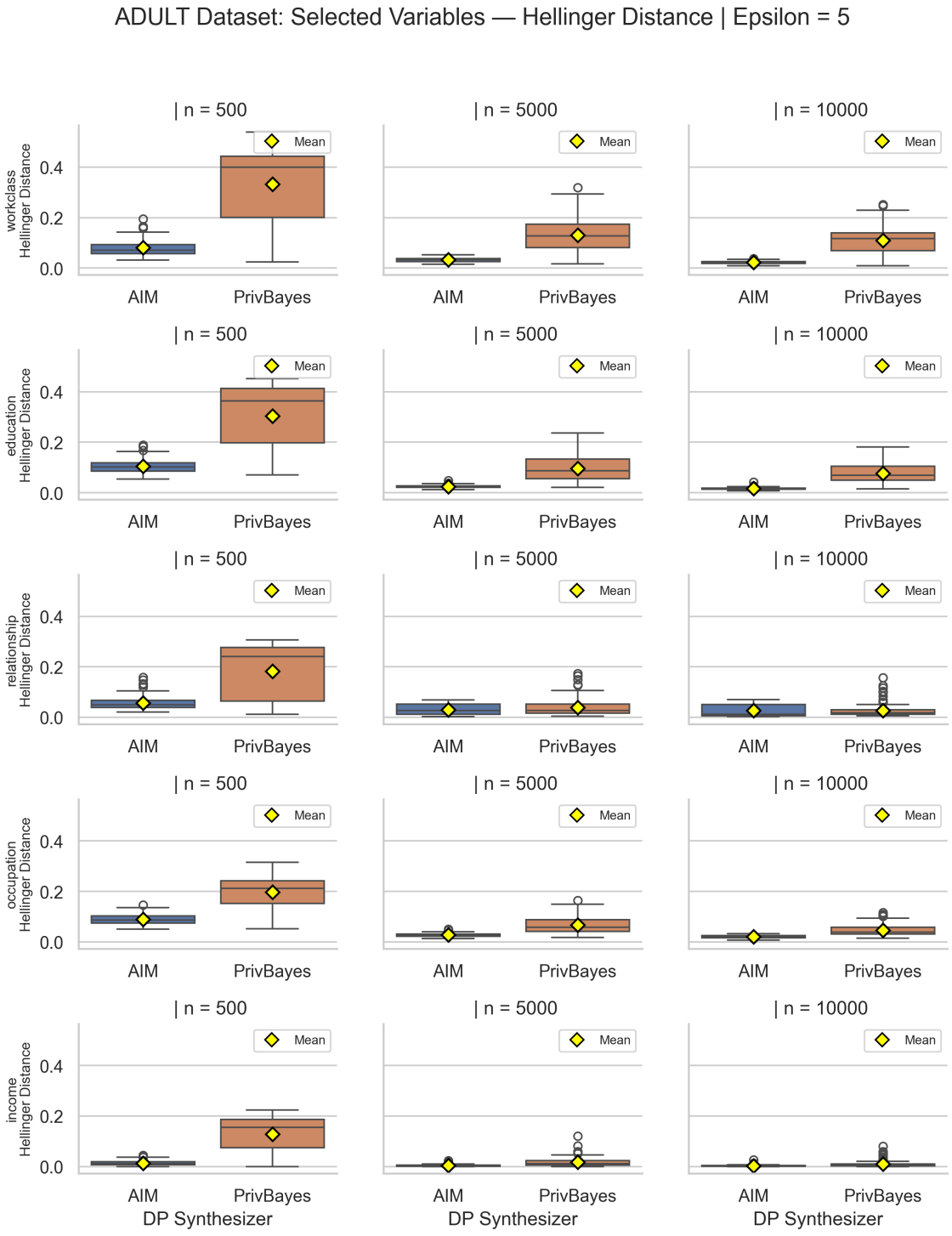

Figure 4.13: Hellinger distance results for selected variables in the Adult dataset with $\epsilon = 5$.

For example, at $n = 500$, the mean Jensen Shannon distance for `education` is 0.1204 for AIM and 0.3553 for PrivBayes, while for `workclass` it is 0.0942 for AIM and 0.3878 for PrivBayes. These differences indicate that PrivBayes has greater difficulty preserving the selected Adult categorical distributions at smaller sample sizes.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.094	0.083	0.043	0.038	0.038	0.038	0.014	0.009	0.027	0.027	0.008	0.006
education	0.120	0.116	0.038	0.031	0.030	0.029	0.008	0.007	0.020	0.019	0.006	0.005
relationship	0.068	0.060	0.034	0.032	0.037	0.033	0.049	0.025	0.034	0.015	0.053	0.028
occupation	0.105	0.101	0.034	0.023	0.033	0.033	0.010	0.008	0.025	0.024	0.011	0.007
income	0.017	0.015	0.015	0.013	0.005	0.004	0.005	0.005	0.004	0.003	0.004	0.004

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
workclass	0.388	0.471	0.278	0.168	0.147	0.144	0.104	0.077	0.123	0.131	0.077	0.057
education	0.355	0.427	0.250	0.148	0.115	0.104	0.091	0.064	0.091	0.082	0.065	0.049
relationship	0.216	0.285	0.249	0.125	0.047	0.033	0.043	0.042	0.033	0.022	0.022	0.031
occupation	0.232	0.249	0.104	0.072	0.077	0.067	0.052	0.038	0.055	0.046	0.031	0.027
income	0.155	0.186	0.133	0.086	0.022	0.016	0.021	0.022	0.013	0.007	0.010	0.017

Table 4.20: Jensen–Shannon distance results for selected Adult with Epsilon 5

As the sample size increases, AIM shows a clear reduction in both average distance and run-to-run dispersion. At $n = 10000$, the mean Jensen Shannon distances for AIM are 0.0265 for `workclass`, 0.0197 for `education`, 0.0248 for `occupation`, and 0.0038 for `income`. PrivBayes improves as well, but remains less competitive for variables with larger category spaces, with mean distances of 0.1231 for `workclass`, 0.0913 for `education`, and 0.0547 for `occupation` at $n = 10000$. The only variable where the difference becomes small is `relationship`; however, the median values and boxplot structure suggest that AIM still provides more reproducible low-distance outcomes, while some outliers affect the mean. Overall, both distances show that, at $\epsilon = 5$, AIM provides stronger and more stable univariate fidelity for the selected Adult variables, whereas PrivBayes remains more variable and less effective for categorical variables with more complex distributions.

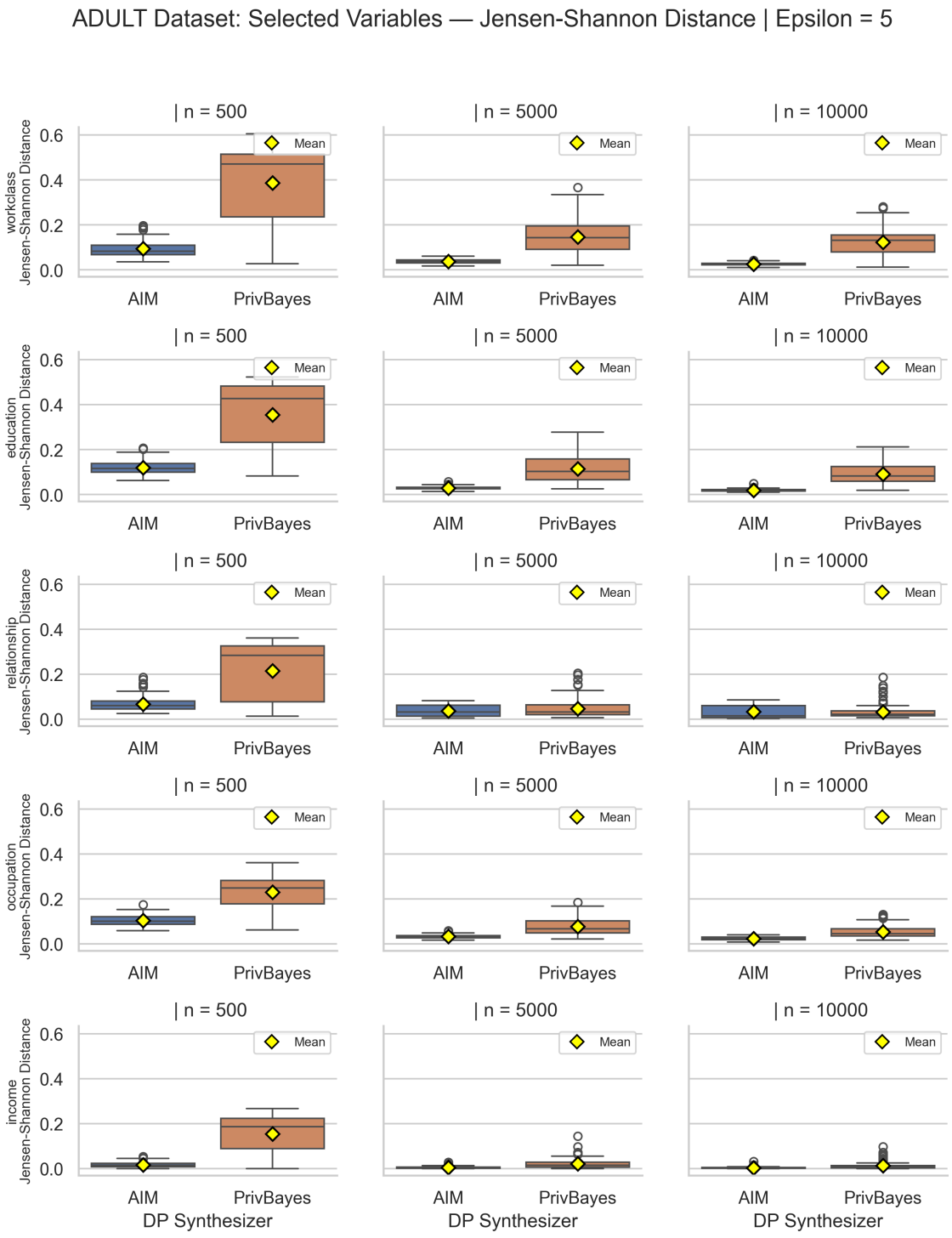

Figure 4.14: Jensen–Shannon distance results for selected variables in the Adult dataset with $\epsilon = 5$.

Cardio Dataset

For the Hellinger distance from Figure 4.15 and Table 4.21, AIM produces lower distances than PrivBayes across all selected variables and sample sizes. This indicates stronger univariate fidelity, while the narrower boxplots show that these fidelity scores are also more stable across repeated runs. The improvement with sample size is clear for AIM. For example, the mean Hellinger distance decreases from 0.0373 to 0.0105 for `age_bin`, from 0.0207 to 0.0050 for `cholesterol`, from 0.0266 to 0.0051 for `gluc`, and from 0.0307 to 0.0049 for `ap_hi_bin` when the sample size increases from $n = 500$ to $n = 10000$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.037	0.034	0.024	0.019	0.014	0.011	0.008	0.007	0.011	0.009	0.007	0.005
cholesterol	0.021	0.020	0.016	0.011	0.007	0.007	0.006	0.004	0.005	0.005	0.003	0.003
gluc	0.027	0.024	0.026	0.016	0.007	0.007	0.006	0.004	0.005	0.005	0.003	0.003
ap_hi_bin	0.031	0.030	0.016	0.013	0.008	0.008	0.004	0.003	0.005	0.005	0.003	0.002
cardio	0.013	0.010	0.017	0.010	0.004	0.004	0.004	0.003	0.003	0.002	0.003	0.002

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.135	0.143	0.115	0.070	0.045	0.044	0.011	0.013	0.055	0.054	0.011	0.009
cholesterol	0.114	0.104	0.145	0.076	0.063	0.064	0.011	0.010	0.094	0.095	0.011	0.008
gluc	0.164	0.179	0.193	0.099	0.051	0.050	0.014	0.009	0.079	0.079	0.009	0.007
ap_hi_bin	0.085	0.075	0.068	0.049	0.067	0.068	0.015	0.011	0.093	0.096	0.012	0.009
cardio	0.016	0.015	0.014	0.012	0.038	0.037	0.013	0.011	0.054	0.056	0.014	0.011

Table 4.21: Hellinger distance results for selected Cardio with Epsilon 5

PrivBayes shows weaker and less consistent behaviour. Its distances decrease from $n = 500$ to $n = 5000$, but they increase again at $n = 10000$ for several variables. For instance, the mean Hellinger distance for `cholesterol` decreases from 0.1140 at $n = 500$ to 0.0634 at $n = 5000$, but increases to 0.0944 at $n = 10000$. A similar pattern is observed for `gluc` and `ap_hi_bin`. The binary variable `cardio` is reproduced accurately by AIM, reaching a mean distance of 0.0029 at $n = 10000$, whereas PrivBayes increases from 0.0158 at $n = 500$ to 0.0540 at $n = 10000$. This

suggests that, under $\epsilon = 5$, AIM benefits more clearly from larger sample sizes, while PrivBayes remains more sensitive to the interaction between sample size and variable structure.

For Jensen Shannon distance from Figure 4.16 and Table 4.22, AIM achieves lower mean and median values across the examined settings than PrivBayes, with compact boxplots at larger sample sizes. For AIM, the mean Jensen Shannon distance decreases from 0.0442 to 0.0118 for `age_bin`, from 0.0248 to 0.0061 for `cholesterol`, from 0.0319 to 0.0061 for `gluc`, and from 0.0369 to 0.0059 for `ap_hi_bin` between $n = 500$ and $n = 10000$. These results show that AIM produces both low average fidelity distances and low-dispersion fidelity measurements across repeated runs.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.044	0.041	0.029	0.022	0.016	0.013	0.010	0.008	0.012	0.010	0.009	0.006
cholesterol	0.025	0.024	0.020	0.013	0.009	0.009	0.008	0.005	0.006	0.006	0.004	0.003
gluc	0.032	0.029	0.032	0.019	0.009	0.008	0.007	0.005	0.006	0.006	0.004	0.003
ap_hi_bin	0.037	0.036	0.019	0.015	0.009	0.009	0.005	0.004	0.006	0.006	0.004	0.003
cardio	0.016	0.012	0.020	0.012	0.005	0.004	0.005	0.004	0.004	0.003	0.004	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.159	0.168	0.135	0.080	0.053	0.053	0.013	0.014	0.066	0.064	0.012	0.010
cholesterol	0.136	0.125	0.174	0.091	0.076	0.077	0.013	0.012	0.113	0.114	0.013	0.009
gluc	0.195	0.213	0.229	0.118	0.062	0.061	0.017	0.011	0.094	0.095	0.011	0.008
ap_hi_bin	0.102	0.091	0.082	0.059	0.080	0.081	0.018	0.013	0.112	0.115	0.014	0.011
cardio	0.019	0.018	0.017	0.014	0.046	0.045	0.016	0.013	0.065	0.067	0.017	0.013

Table 4.22: Jensen–Shannon distance results for selected Cardio with Epsilon 5

PrivBayes again shows higher distances and a less monotonic sample-size pattern. Its Jensen Shannon distance values improve from $n = 500$ to $n = 5000$, but increase again at $n = 10000$ for all selected variables. For example, the mean distance for `gluc` decreases from 0.1950 to 0.0616 and then rises to 0.0943. For `ap_hi_bin`, the corresponding values are 0.1015, 0.0801, and 0.1119. The same trend is visible for `cardio`, where the mean distance increases to 0.0649 at $n = 10000$. Overall, both

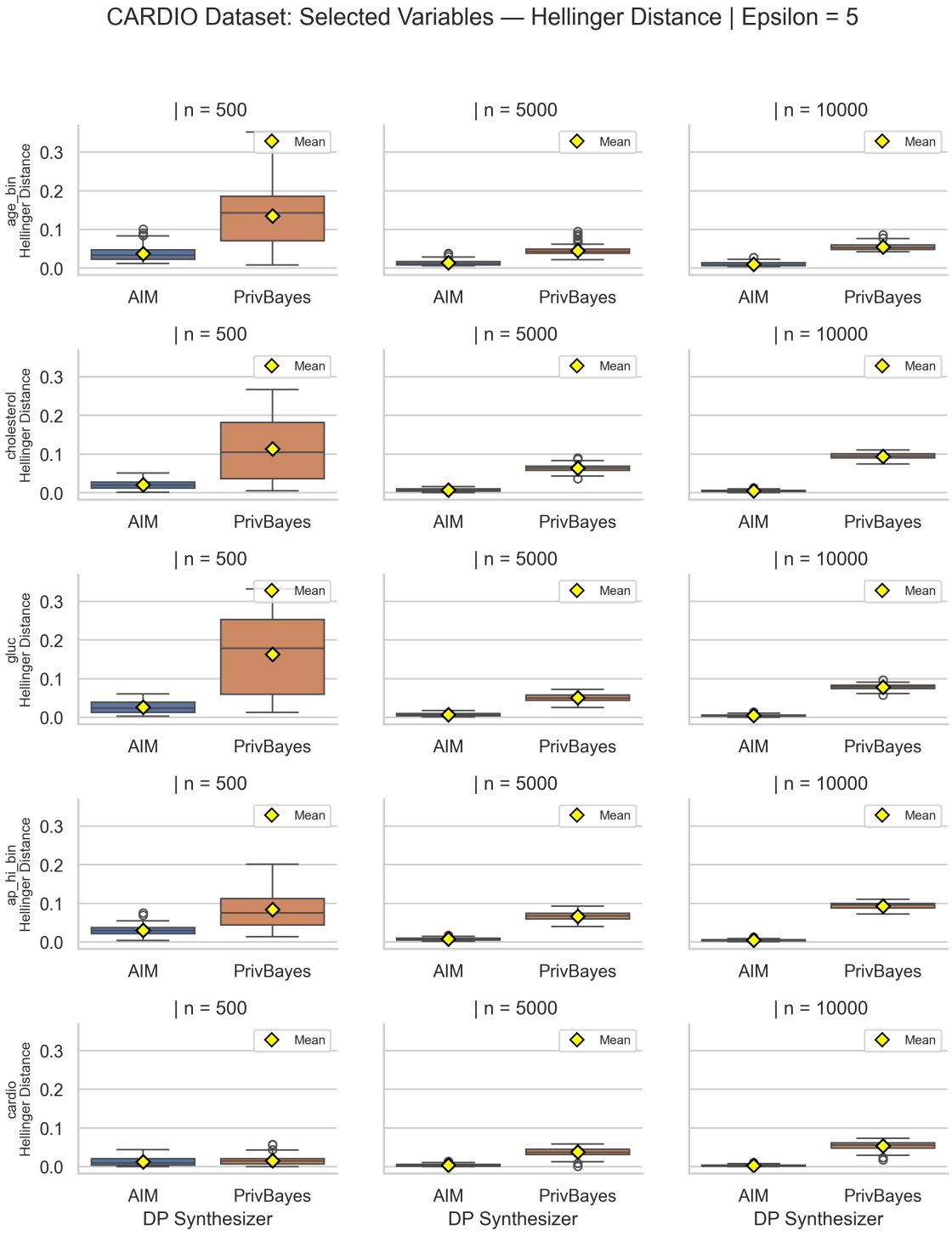

Figure 4.15: Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 5$.

fidelity metrics indicate that AIM provides stronger and more stable preservation of the selected Cardio marginal distributions at $\epsilon = 5$. PrivBayes improves at the medium sample size, but its higher distances and renewed increase at $n = 10000$ show less stable fidelity behaviour across repeated runs.

4.3.3 Differential Privacy Budget $\epsilon = 10$

Cardio Dataset

Figure 4.17 and Table 4.23 shows that AIM achieves lower Hellinger distances than PrivBayes across all selected variables and sample sizes. The difference is already visible at $n = 500$, where AIM reports lower mean distances for `age_bin`, `cholesterol`, `gluc`, `ap_hi_bin`, and `cardio`. The boxplots for AIM are also more compact, indicating more stable fidelity scores across repeated runs.

AIM shows a consistent improvement as the sample size increases. For example, the mean Hellinger distance decreases from 0.0310 to 0.0095 for `age_bin`, from 0.0213 to 0.0050 for `cholesterol`, from 0.0260 to 0.0056 for `gluc`, and from 0.0283 to 0.0049 for `ap_hi_bin` when moving from $n = 500$ to $n = 10000$. The binary target variable `cardio` also reaches a very low mean distance of 0.0028 at $n = 10000$. These results indicate that AIM benefits clearly from larger sample sizes and produces both low average fidelity distances and low-variability repeated-runbehaviour.

PrivBayes shows a less uniform sample-size pattern. Although it performs reasonably at $n = 500$, its mean Hellinger distances increase at larger sample sizes for several variables. For instance, the mean distance for `cholesterol` increases from 0.0483 at $n = 500$ to 0.0976 at $n = 10000$, while `ap_hi_bin` increases from 0.0481 to 0.0954. A similar increase is observed for `gluc` and `cardio`. This suggests that, for the Cardio dataset, increasing the sample size does not necessarily improve the marginal fidelity of PrivBayes under $\epsilon = 10$.

For Jensen Shannon metric, from Figure 4.18 and Table 4.24, AIM again produces

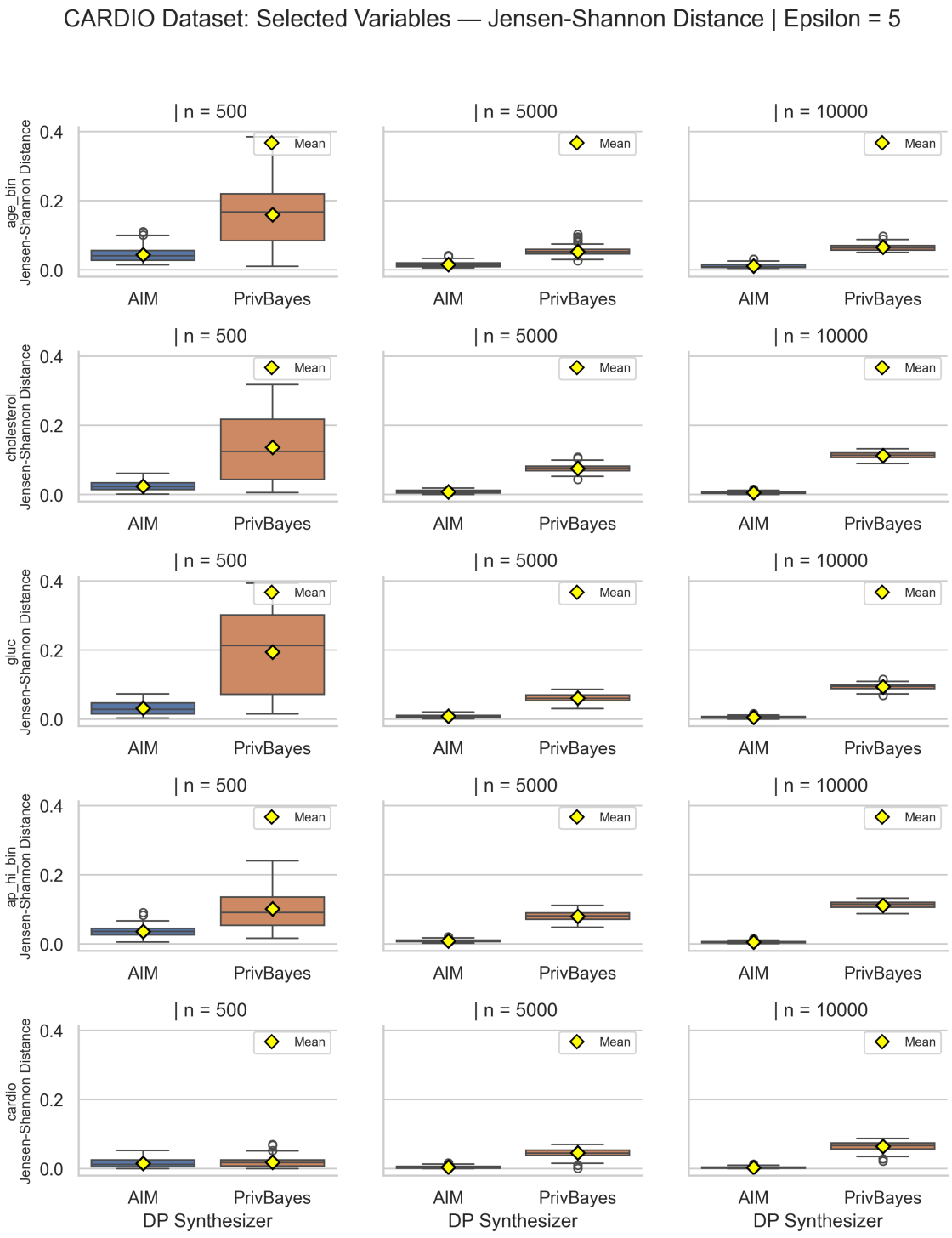

Figure 4.16: Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 5$.

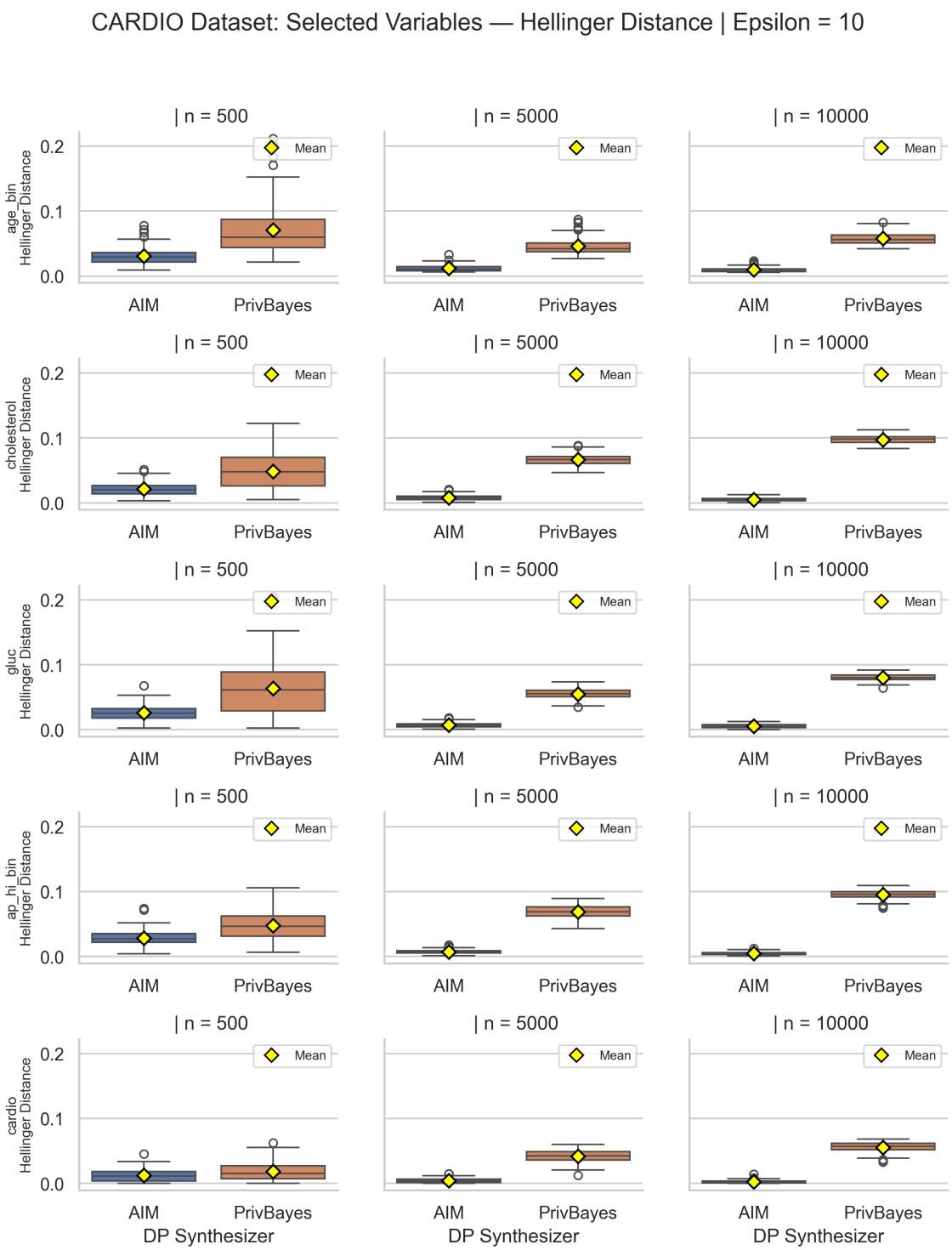

Figure 4.17: Hellinger distance results for selected variables in the Cardio dataset with $\epsilon = 10$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.031	0.029	0.015	0.014	0.012	0.010	0.006	0.005	0.010	0.008	0.004	0.004
cholesterol	0.021	0.020	0.013	0.010	0.008	0.008	0.005	0.004	0.005	0.005	0.004	0.003
gluc	0.026	0.026	0.015	0.012	0.007	0.007	0.005	0.004	0.006	0.005	0.005	0.003
ap_hi_bin	0.028	0.028	0.014	0.012	0.008	0.007	0.003	0.003	0.005	0.005	0.003	0.002
cardio	0.012	0.012	0.014	0.010	0.004	0.004	0.005	0.003	0.003	0.002	0.003	0.002

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.071	0.060	0.044	0.038	0.046	0.042	0.013	0.013	0.058	0.056	0.012	0.010
cholesterol	0.048	0.048	0.044	0.026	0.067	0.067	0.011	0.009	0.098	0.098	0.009	0.006
gluc	0.064	0.061	0.060	0.040	0.055	0.056	0.010	0.009	0.080	0.080	0.007	0.005
ap_hi_bin	0.048	0.046	0.031	0.022	0.069	0.069	0.014	0.009	0.095	0.096	0.009	0.007
cardio	0.019	0.015	0.020	0.013	0.042	0.042	0.013	0.010	0.056	0.057	0.010	0.008

Table 4.23: Hellinger distance results for selected Cardio with Epsilon 10

lower mean and median distances than PrivBayes across the selected variables. For AIM, the mean Jensen Shannon distance decreases from 0.0366 to 0.0108 for `age_bin`, from 0.0256 to 0.0060 for `cholesterol`, from 0.0312 to 0.0067 for `gluc`, and from 0.0340 to 0.0059 for `ap_hi_bin` between $n = 500$ and $n = 10000$. The small standard deviations and interquartile ranges at larger sample sizes show that these fidelity measurements are show low dispersion across repeated runs.

PrivBayes again shows higher Jensen Shannon distances than AIM, with a non-monotonic sample-size pattern. For `cholesterol`, the mean distance increases from 0.0580 at $n = 500$ to 0.1170 at $n = 10000$. For `ap_hi_bin`, it increases from 0.0577 to 0.1146, and for `cardio` from 0.0222 to 0.0669. This pattern indicates that larger sample sizes do not always improve PrivBayes fidelity for the selected Cardio variables in this configuration. Overall, both fidelity metrics show that AIM provides stronger and more stable univariate fidelity at $\epsilon = 10$, while PrivBayes produces more stable boxplots at larger sample sizes but with higher average distances, indicating weaker preservation of the real marginal distributions.

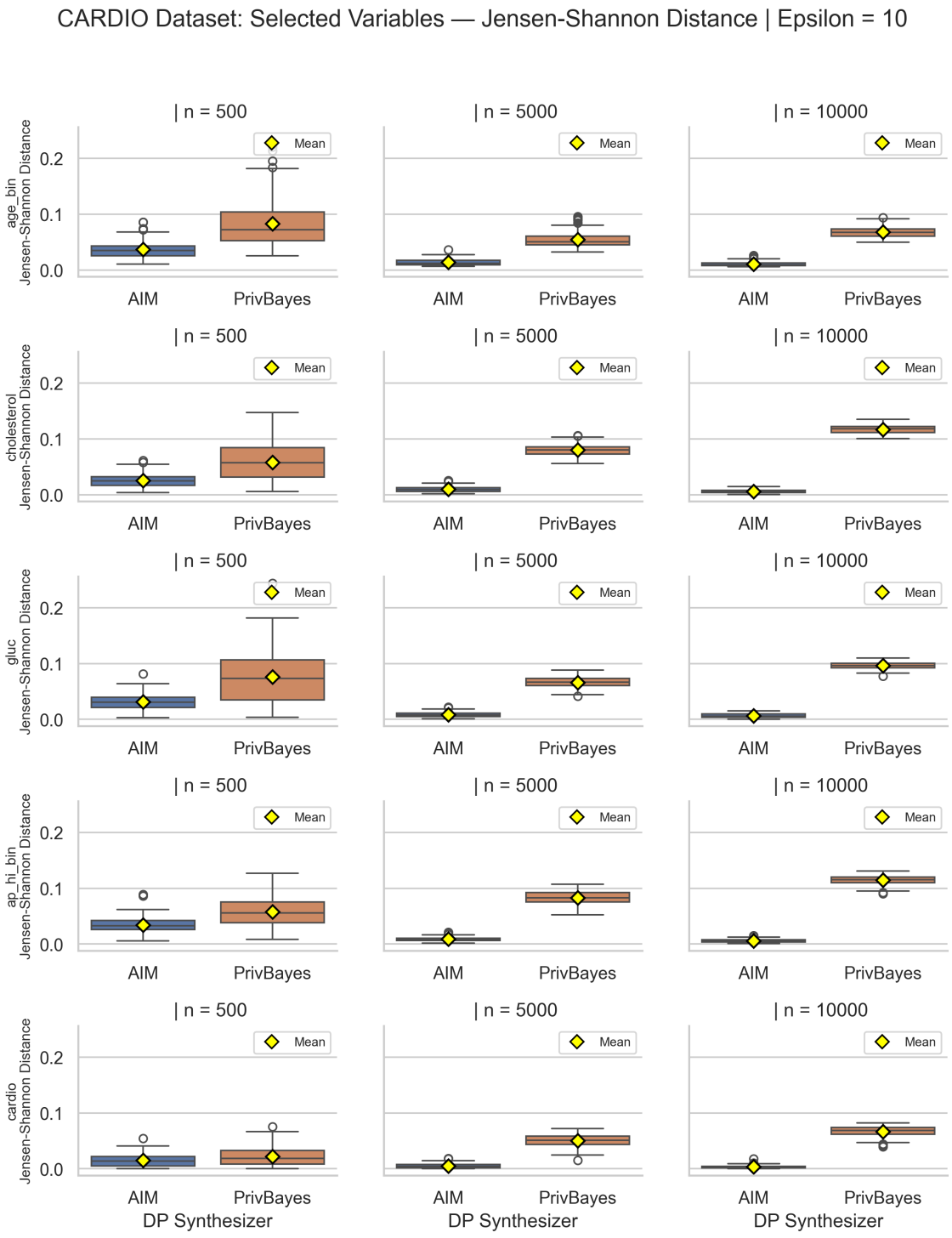

Figure 4.18: Jensen–Shannon distance results for selected variables in the Cardio dataset with $\epsilon = 10$.

AIM

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.037	0.035	0.017	0.016	0.014	0.012	0.008	0.006	0.011	0.009	0.005	0.005
cholesterol	0.026	0.024	0.015	0.012	0.010	0.009	0.007	0.005	0.006	0.006	0.005	0.003
gluc	0.031	0.031	0.018	0.014	0.009	0.008	0.006	0.005	0.007	0.006	0.006	0.004
ap_hi_bin	0.034	0.033	0.016	0.015	0.009	0.009	0.004	0.004	0.006	0.006	0.004	0.003
cardio	0.015	0.014	0.017	0.012	0.005	0.005	0.006	0.004	0.003	0.003	0.003	0.003

PrivBayes

Variable	500				5000				10000			
	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD	Mean	Med.	IQR	SD
age_bin	0.083	0.072	0.052	0.042	0.054	0.051	0.016	0.014	0.068	0.068	0.013	0.010
cholesterol	0.058	0.057	0.053	0.031	0.080	0.080	0.013	0.010	0.117	0.118	0.011	0.007
gluc	0.076	0.074	0.072	0.048	0.066	0.067	0.012	0.010	0.096	0.096	0.008	0.006
ap_hi_bin	0.058	0.056	0.038	0.026	0.083	0.083	0.017	0.011	0.115	0.115	0.010	0.008
cardio	0.022	0.018	0.024	0.016	0.050	0.051	0.015	0.012	0.067	0.069	0.012	0.010

Table 4.24: Jensen–Shannon distance results for selected Cardio with Epsilon 10

4.4 Discussion on DP Synthesizers

For comparing the variability of the synthesizers, the main difference between AIM and PrivBayes was observed in their run-to-run variability. AIM shows much lower variability than PrivBayes across almost all datasets, privacy budgets, sample sizes, and selected variables. In the Adult dataset, the average Hellinger standard deviation for AIM was about 0.023, while PrivBayes reaches about 0.066. A similar gap was observed for Jensen Shannon Distance, where AIM has an average standard deviation of about 0.025 compared with about 0.076 for PrivBayes. From that we can conclude that PrivBayes varies roughly two to three times more than AIM in the Adult dataset. The same pattern also appears in the Bank dataset, where AIM has an average Hellinger standard deviation of about 0.022 compared with 0.056 for PrivBayes. For Jensen Shannon Distance, the values are about 0.024 for AIM and 0.066 for PrivBayes. The wider boxplots for PrivBayes, especially for variables such as `job`, `education`, `contact`, `poutcome`, and `y`, show that its repeated runs have higher dispersion, particularly at smaller sample sizes.

In the Cardio dataset, AIM still remains more stable than PrivBayes, although

the difference between the two models was smaller than in Adult and Bank. AIM has an average Hellinger standard deviation of about 0.012, while PrivBayes has about 0.023. For Jensen Shannon distance, AIM has an average standard deviation of about 0.013 compared with 0.027 for PrivBayes. The Cardio boxplots show that both models become show lower variability at larger sample sizes, but AIM continues to produce narrower boxes. The results show a clear stability ranking, where AIM is more stable than PrivBayes. This suggests that AIM is more reliable when the same DP synthesis process is repeated multiple times.

While comparing the the influence of data size, it was found that larger sample size generally reduces run-to-run variability, especially for AIM. This effect was most visible when the sample size increases from $n = 500$ to $n = 5000$, while the improvement from $n = 5000$ to $n = 10000$ was usually smaller. In the Adult dataset, AIM's average Hellinger standard deviation decreases from about 0.044 at $n = 500$ to 0.010 at $n = 5000$ and 0.009 at $n = 10000$. Jensen–Shannon Distance followed the same trend, decreasing from about 0.048 to 0.012 and then 0.011. PrivBayes also becomes more stable with larger sample sizes, but it remains more variable than AIM. Its Hellinger standard deviation decreases from about 0.101 at $n = 500$ to 0.065 at $n = 5000$ and 0.032 at $n = 10000$.

The same sample-size effect was also observed in the Bank and Cardio datasets. In Bank, AIM's average Hellinger standard deviation decreases from about 0.036 at $n = 500$ to 0.008 at both $n = 5000$ and $n = 10000$. PrivBayes shows a larger reduction, from about 0.112 at $n = 500$ to 0.046 at $n = 5000$ and 0.010 at $n = 10000$, which shows that it benefits strongly from larger samples, although it was still less stable at smaller sample sizes. In Cardio, both models become more stable as the sample size increases. AIM's Hellinger standard deviation decreases from about 0.025 to 0.006 and then 0.004, while PrivBayes decreases from about 0.049 to 0.011 and then 0.009. However, lower variability does not always mean better fidelity. In the

Cardio dataset, PrivBayes becomes more stable at larger sample sizes, but its mean distance for some variables still remains higher than AIM.

Another observation about the variability, it differs clearly across variables, which shows that the synthesizers do not reproduce all variables with the same level of reproducibility. In the Adult dataset, `workclass` and `education` shows the highest variability overall. For Hellinger distance, their average standard deviations are about 0.063 and 0.056, respectively, while for Jensen Shannon distance they increased to about 0.071 and 0.064. These variables contain several categories and may include smaller groups, which makes them harder to reproduce under differential privacy. In contrast, `income` was the variable with the lowest run-to-run dispersion, with an average Hellinger standard deviation of about 0.025 and Jensen Shannon standard deviation of about 0.030. This was expected because `income` has fewer categories and a simpler distribution.

A similar pattern was observed in the Bank and Cardio datasets. In Bank, `poutcome` has the highest variability, followed by `job`. The average Hellinger standard deviation was about 0.058 for `poutcome` and 0.043 for `job`, while the Jensen Shannon standard deviation was about 0.068 and 0.049, respectively. This suggests that variables with several categories or uneven category frequencies are more difficult for DP generation. The target variable `y` was simpler, but its imbalance can still create some variability, especially for PrivBayes at smaller sample sizes. In Cardio, the overall variability is lower because the selected variables are binned or low-cardinality variables. However, `gluc`, `age_bin`, and `cholesterol` still vary more than `cardio`. For example, `gluc` has an average Hellinger standard deviation of about 0.022 and Jensen–Shannon standard deviation of about 0.026, while `cardio` has lower values of about 0.009 and 0.011. Variables with more categories or uneven distributions show higher variability, while binary or simpler variables are reproduced with lower variability.

When comparing both the metrics, the results show that Jensen Shannon distance is slightly more variable than Hellinger distance under almost all observed conditions. This pattern can be observed in both the numerical tables and the boxplots, where the Jensen Shannon boxplots usually have wider spreads than the Hellinger boxplots. Across all DP comparisons, the Jensen Shannon mean was higher than the Hellinger mean in all 250 cases. Its standard deviation was also higher in all 250 cases, while its interquartile range was higher in 247 out of 250 cases. This indicates that Jensen Shannon distance reacts more strongly to changes in the generated distributions across repeated runs.

However, this does not reduce the usefulness of Hellinger Distance. Both metrics give the same overall model ranking and show the same main trends across Adult, Bank, and Cardio. The main difference was that Jensen Shannon distance gives slightly larger changes when the synthetic distribution varies from one run to another. This difference was small for AIM, especially at larger sample sizes, but it becomes more visible for PrivBayes, particularly at $n = 500$ and $\epsilon = 1$, where the boxplots are comparatively wider. Therefore, in these results, we can say that Jensen Shannon distance is slightly more sensitive and more variable for almost all settings.

Apart from that, there were several recurring patterns observed across the Adult, Bank, and Cardio datasets. Firstly, the AIM generally produced lower distance values and lower run-to-run variability than PrivBayes. This was the strongest overall pattern in the results, as AIM usually produced lower and narrower boxplots, which indicates that it preserved the selected univariate distributions more accurately and with lower run-to-run dispersion. Sample size also had a clear effect. Increasing the sample size usually reduced variability, especially when moving from $n = 500$ to $n = 5000$. The improvement from $n = 5000$ to $n = 10000$ was usually smaller, which suggests that most of the stability gain occurred after increasing the sample size beyond the smallest setting. This effect was stronger for AIM, while PrivBayes

also improved but remained more variable in many cases.

We can also notice that the privacy budget also affected the results. Moving from $\epsilon = 1$ to $\epsilon = 5$ usually reduced both the mean distance and the spread of the boxplots, while the improvement from $\epsilon = 5$ to $\epsilon = 10$ was smaller. This suggested that the main benefit came from moving from a strict privacy setting to a moderate privacy setting. PrivBayes was especially unstable at smaller sample sizes, particularly in Adult and Bank at $n = 500$, where variables such as `workclass`, `education`, `job`, `contact`, and `poutcome` showed wider boxplots. At larger sample sizes, PrivBayes became more stable, but it usually still had higher distance values than AIM. The Cardio dataset behaved slightly differently, as its selected variables were more stable overall, especially for AIM. This was likely because the selected Cardio variables were binned or low-cardinality variables, such as `age_bin`, `ap_hi_bin`, `cholesterol`, `gluc`, and `cardio`, which were easier to reproduce as marginal distributions than high-cardinality variables such as `workclass` or `job`.

The observed behaviour can also be connected to the architecture of the two DP synthesizers. AIM was designed as a workload-adaptive mechanism, where useful queries were selected, privately measured, and then used to generate synthetic data from noisy measurements [16]. This design was suitable for preserving marginal distributions, especially because the evaluation focused on univariate fidelity scores such as Hellinger and Jensen Shannon distances. We can relate this and understand why AIM showed stronger and more reproducible performance in the plots.

PrivBayes, in contrast, was based on DP Bayesian networks. It learned a Bayesian network structure and used noisy conditional distributions to generate synthetic data. This approach was useful for modelling dependencies between variables, but it also introduced more variation across repeated runs, especially when the sample size was small, the variable had many categories, or the privacy budget was strict [19]. This architecture-level difference matched the observed results: AIM was more direct

for preserving selected marginal distributions, while PrivBayes depended on private structure learning and conditional distribution estimation. As a result, PrivBayes varied more across repeated runs, particularly in Adult and Bank, where variables such as `workclass`, `education`, `job`, and `poutcome` had more complex category structures.

5 Conclusion

This thesis aimed to study the fidelity and run-to-run stability of synthetic tabular data generation methods. The main objective was to compare how closely DP and Non-DP synthesizers reproduce the univariate distributions of real data, and how variable these fidelity results are across repeated runs.

The experiments used three tabular datasets: Adult, Bank Marketing, and Cardiovascular Disease. The Non-DP synthesizers were Gaussian Copula, CTGAN, TVAE, and RTVAE, while the DP synthesizers were AIM and PrivBayes. The models were evaluated across different sample sizes and, for DP models, across privacy budgets of $\epsilon = 1$, $\epsilon = 5$, and $\epsilon = 10$.

5.1 Answering the Research Questions

The first research question examined how much statistical dispersion is observed in the fidelity scores across repeated runs, and variable types affect the observed fidelity scores. The results showed that run-to-run dispersion differed clearly across synthesizers, datasets, and variables. In the Non-DP setting, Gaussian Copula produced the lowest dispersion overall, especially for the Adult and Bank datasets, where it showed low distance values and narrow boxplots. In contrast, CTGAN, TVAE, and RTVAE showed larger variation across repeated runs, particularly for variables with larger category spaces or more complex distributions. The Cardio dataset was comparatively more difficult, mainly because variables such as `age`,

`ap_hi`, and `ap_hi_bin` produced higher distance values. Binary or low-cardinality variables such as `income`, `y`, and `cardio` were generally easier to reproduce.

The second research question examined how DP and Non-DP synthesizers differ in terms of fidelity and run-to-run dispersion. In the Non-DP setting, Gaussian Copula achieved the strongest overall combination of low distance values and low dispersion, especially for Adult and Bank. The neural network based models showed more variable behaviour, with RTVAE producing the weakest results in several settings. In the DP setting, AIM generally outperformed PrivBayes by producing lower distance values and smaller run-to-run dispersion across most datasets, sample sizes, and privacy budgets. PrivBayes improved in some larger sample settings, but it remained more variable, especially for variables with complex categorical structures.

The third research question examined how the privacy budget affects the fidelity and run-to-run dispersion of DP synthesizers. Increasing the privacy budget from $\epsilon = 1$ to $\epsilon = 5$ generally reduced both the mean distance values and the spread of the repeated-run scores. This improvement was especially clear for AIM. However, the change from $\epsilon = 5$ to $\epsilon = 10$ did not produce a uniform additional improvement across all variables, datasets, and synthesizers. In some cases, the distance values decreased further, while in others the improvement was small or the pattern was non-monotonic, particularly for PrivBayes on the Cardio dataset. Therefore, the main improvement came from moving from a strict privacy setting to a moderate privacy setting.

The fourth research question examined how increasing sample size influences the fidelity scores and their dispersion across repeated runs. The results showed that larger sample sizes in some cases reduced both the average distance values and the run-to-run dispersion, especially when moving from $n = 500$ to $n = 5000$. This pattern was strongest for Gaussian Copula in the Non-DP setting and AIM in the DP setting. However, the improvement from $n = 5000$ to $n = 10000$ was smaller, and

some synthesizers did not benefit uniformly from larger samples. CTGAN, RTVAE, and PrivBayes still showed variability in several settings, indicating that sample size helps but does not fully remove the effects of model architecture, stochastic training, privacy noise, or difficult variable distributions.

5.2 Conclusions, Limitations and Future Work

The main conclusion of this thesis is that a reliable synthesizer should produce both low distance values and low variation across repeated runs as observed in Gaussian Copulas in several of the evaluated datasets. A model with low average distance but high variability may not be dependable in practice, because its output quality can change from one run to another. Similarly, a model with low variability but high distance may have low run-to-run variability, but it may still fail to reproduce the real data well. This was visible in the Cardio dataset, where some variables remained difficult even when the model behaviour was stable across runs.

The findings also highlights that binary and low-cardinality variables were generally easier to reproduce, while variables with many categories, rare groups, imbalance, or irregular numerical distributions were more difficult. Therefore, synthetic data evaluation should be carried out at the variable level, not only at the dataset level.

This study has some limitations. First, the evaluation mainly focused on univariate fidelity. The metrics measured how well individual variables were reproduced, but they did not fully examine relationships between variables or the overall joint distribution. Due to time constraint this as not explored, and as a result, the findings provide only a partial view of synthetic data quality. Future work should therefore incorporate multivariate fidelity measures, such as correlation analysis and conditional distribution analysis, to provide a more comprehensive assessment of how well synthetic data preserves complex relationships within the original dataset.

Another limitation is that the study did not examine machine learning utility

in detail. While this thesis findings provide important insight into distributional fidelity, they do not indicate how useful the synthetic data would be for downstream predictive tasks. Future research should therefore include utility-based evaluations in which models are trained on synthetic data and tested on real data. This would be helpful in determining whether the low-dispersion fidelity scores also translate into practical usefulness for classification and regression tasks or not.

Third, the study did not include a full privacy risk analysis for the Non-DP models. Differential privacy provides formal protection for AIM and PrivBayes, but methods such as Gaussian Copula, CTGAN, TVAE, and RTVAE may still pose privacy risks that were not evaluated. Future work should include privacy risk analyses, such as membership inference and attribute inference attacks, to better understand the trade-off between data utility and privacy protection across different synthesis approaches.

6 Disclosure of AI Assistance in the Thesis

During the preparation of this thesis, publicly available AI tools such as ChatGPT (https://www.chatgpt.com) and Perplexity (https://www.perplexity.ai/) were used as supportive aids for tasks such as organising the thesis, assisting with code development, preparing computational commands, understanding literature, and refining language. These tools were used only to assist the process, while all key research decisions, experimental design, analysis, and final interpretations were carried out independently by the author.

AI tools supported the development and debugging of Python code used in the experimental workflow, including dataset handling, experiment setup, and working with various synthesizers. They were also used to help prepare and refine Puhti supercomputer commands and SLURM job scripts. All generated code and commands were carefully reviewed, modified, and validated by the author before use.

Additionally, tools like QuillBot (https://quillbot.com) and ChatPDF (https://www.chatpdf.com) were used in understanding research literature and technical concepts, as well as for language editing and improving clarity. While they helped in summarising complex topics and refine my writing, the selection of references, interpretation of findings, and final presentation were completed by the author.

References

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets", in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, IEEE, 2008, pp. 111–125.

[2] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility", *Scientific reports*, vol. 3, no. 1, p. 1376, 2013.

[3] L. Sweeney, "K-anonymity: A model for protecting privacy", *International journal of uncertainty, fuzziness and knowledge-based systems*, vol. 10, no. 05, pp. 557–570, 2002.

[4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity", *Acm transactions on knowledge discovery from data (tkdd)*, vol. 1, no. 1, 3–es, 2007.

[5] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity", in *2007 IEEE 23rd international conference on data engineering*, IEEE, 2006, pp. 106–115.

[6] J. Jordon et al., "Synthetic data–what, why and how?", *arXiv preprint arXiv:2205.03257*, 2022.

[7] M. Hernadez, G. Epelde, A. Alberdi, R. Cilla, and D. Rankin, "Synthetic tabular data evaluation in the health domain covering resemblance, utility, and

privacy dimensions", *Methods of information in medicine*, vol. 62, no. S 01, e19–e38, 2023.

[8] D. Arpit et al., "A closer look at memorization in deep networks", in *International conference on machine learning*, PMLR, 2017, pp. 233–242.

[9] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault", in *2016 IEEE international conference on data science and advanced analytics (DSAA)*, IEEE, 2016, pp. 399–410.

[10] J. Drechsler, "Using support vector machines for generating synthetic datasets", in *International conference on privacy in statistical databases*, Springer, 2010, pp. 148–161.

[11] G. Caiola and J. P. Reiter, "Random forests for generating partially synthetic, categorical data.", *Trans. Data Priv.*, vol. 3, no. 1, pp. 27–42, 2010.

[12] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan", *Advances in neural information processing systems*, vol. 32, 2019.

[13] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks", in *Machine learning for healthcare conference*, PMLR, 2017, pp. 286–305.

[14] C. Dwork, "Differential privacy", in *Encyclopedia of Cryptography, Security and Privacy*, Springer, 2025, pp. 649–652.

[15] I. Dinur and K. Nissim, "Revealing information while preserving privacy", in *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2003, pp. 202–210.

[16] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau, "Aim: An adaptive and iterative mechanism for differentially private synthetic data", *arXiv preprint arXiv:2201.12677*, 2022.

[17] M. Hernandez, P. A. Osorio-Marulanda, M. Catalina, L. Loinaz, G. Epelde, and N. Aginako, "Comprehensive evaluation framework for synthetic tabular data in health: Fidelity, utility and privacy analysis of generative models with and without privacy guarantees", *Frontiers in Digital Health*, vol. 7, p. 1 576 290, 2025.

[18] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, "Generation and evaluation of synthetic patient data", *BMC medical research methodology*, vol. 20, no. 1, p. 108, 2020.

[19] 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)*, vol. 42, no. 4, pp. 1–41, 2017.

[20] T. Stadler, B. Oprisanu, and C. Troncoso, "Synthetic data–anonymisation groundhog day", in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1451–1468.

[21] R. B. Nelsen, *An introduction to copulas*. Springer, 2006.

[22] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks", *arXiv preprint arXiv:1806.03384*, 2018.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[24] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans", *Advances in neural information processing systems*, vol. 30, 2017.

[26] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks", in *International conference on machine learning*, Pmlr, 2017, pp. 214–223.

[27] I. J. Goodfellow et al., "Generative adversarial nets", *Advances in neural information processing systems*, vol. 27, 2014.

[28] R. Camino, C. Hammerschmidt, and R. State, "Generating multi-categorical samples with generative adversarial networks", *arXiv preprint arXiv:1807.01202*, 2018.

[29] A. Yahi, R. Vanguri, N. Elhadad, and N. P. Tatonetti, "Generative adversarial networks for electronic health records: A framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories", *arXiv preprint arXiv:1712.00164*, 2017.

[30] Z. Che, Y. Cheng, S. Zhai, Z. Sun, and Y. Liu, "Boosting deep learning risk prediction with generative adversarial networks for electronic health records", in *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2017, pp. 787–792.

[31] J. Jordon, J. Yoon, and M. Van Der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees", in *International conference on learning representations*, 2018.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational bayes", *arXiv preprint arXiv:1312.6114*, 2013.

[33] H. Akrami, S. Aydore, R. M. Leahy, and A. A. Joshi, "Robust variational autoencoder for tabular data with beta divergence", *arXiv preprint arXiv:2006.08204*, 2020.

[34] Z. Qian, R. Davis, and M. Van Der Schaar, "Synthcity: A benchmark framework for diverse use cases of tabular synthetic data", *Advances in neural information processing systems*, vol. 36, pp. 3173–3188, 2023.

[35] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", *Foundations and trends® in theoretical computer science*, vol. 9, no. 3-4, pp. 211–487, 2014.

[36] J. Ullman and S. Vadhan, "Pcps and the hardness of generating private synthetic data", in *Theory of Cryptography Conference*, Springer, 2011, pp. 400–416.

[37] B. Van Breugel, T. Kyono, J. Berrevoets, and M. Van der Schaar, "Decaf: Generating fair synthetic data using causally-aware generative networks", *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 221–22 233, 2021.

[38] A. Alaa, B. Van Breugel, E. S. Saveliev, and M. Van Der Schaar, "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models", in *International conference on machine learning*, PMLR, 2022, pp. 290–306.

[39] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics", *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[40] J. Lin, "Divergence measures based on the shannon entropy", *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[41] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions", *IEEE Transactions on Information theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[42] B. Becker and R. Kohavi, *Adult*, UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5XW20, 1996.

[43] S. Moro, P. Rita, and P. Cortez, *Bank Marketing*, UCI Machine Learning Repository, 2014. DOI: 10.24432/C5K306.

[44] S. Ulianova, *Cardiovascular disease dataset*, Kaggle Dataset, 2019.

[45] G. Ganev, M. S. Muthu Selva Annamalai, S. Mahiou, and E. De Cristofaro, "The importance of being discrete: Measuring the impact of discretization in end-to-end differentially private synthetic data", in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2025, pp. 3236–3250.

[46] T. H. A. Tran, M. L. Wiesner, and M. van Keulen, "Influence of discretization granularity on learning classification models", in *BNAIC/BeNeLearn 2022 Joint International Scientific Conferences on AI and Machine Learning*, 2022.

[47] *SciPy — scipy.org*, https://scipy.org/, [Accessed 27-06-2026].

[48] *Ctgan: Synthetic data generation for single table data*, https://pypi.org/project/ctgan/, [Accessed 27-06-2026].

[49] *Welcome to the SDV! | Synthetic Data Vault — docs.sdv.dev*, https://docs.sdv.dev/sdv, [Accessed 27-06-2026].

[50] *Synthcity: Synthetic data generation and evaluation framework*, https://pypi.org/project/synthcity/, [Accessed 27-06-2026].

[51] *SmartNoise 2014; OpenDP SmartNoise — docs.smartnoise.org*, https://docs.smartnoise.org/, [Accessed 27-06-2026].

[52] *Installation 2014; mbi documentation — private-pgm.readthedocs.io*, https://private-pgm.readthedocs.io/en/latest/installation.html, [Accessed 27-06-2026].

Appendix A Complete Non-Private Synthesizers Variable-Level Boxplots

This appendix presents boxplots for all variables for Hellinger distance and Jensen Shannon distance. The figures complement the selected-variable analysis in Chapter 4. The non private synthesizers included are Gaussian Copula, CTGAN, TVAE, and RTVAE.

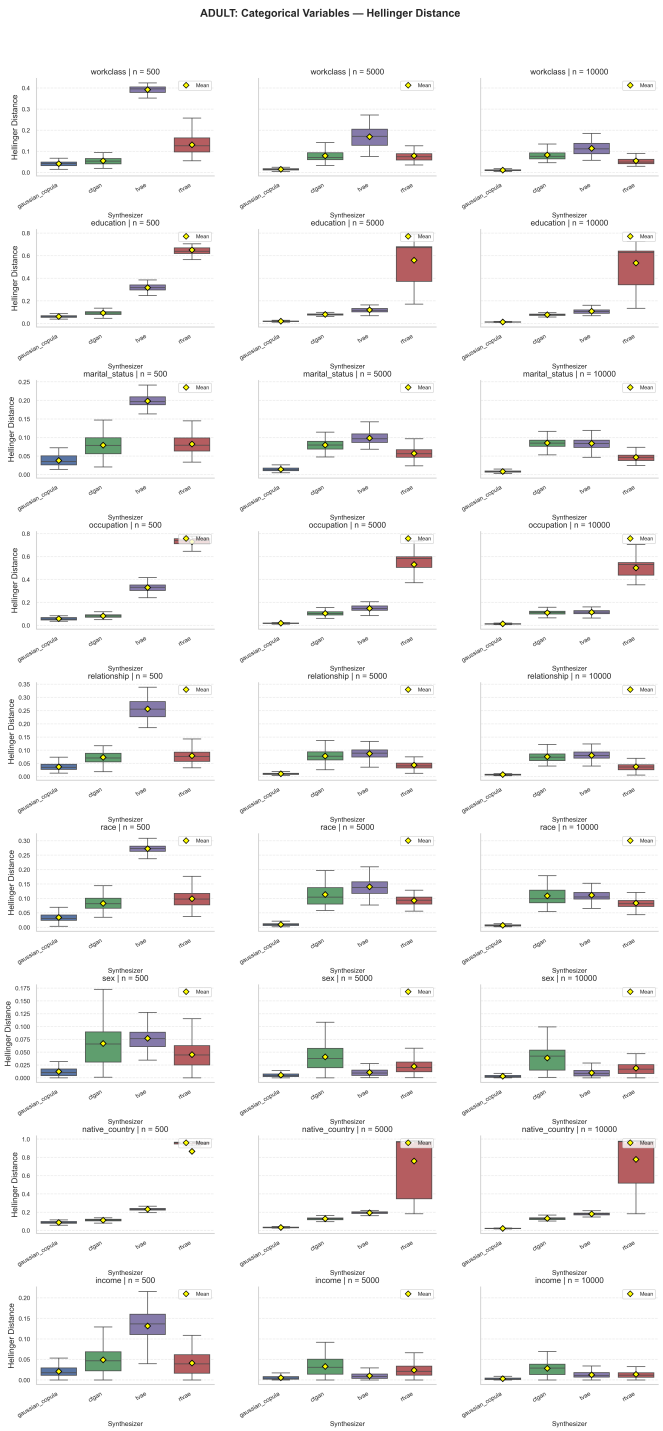

Figure A.1: Hellinger distance results for all variables in the Adult dataset using non-private synthesizers.

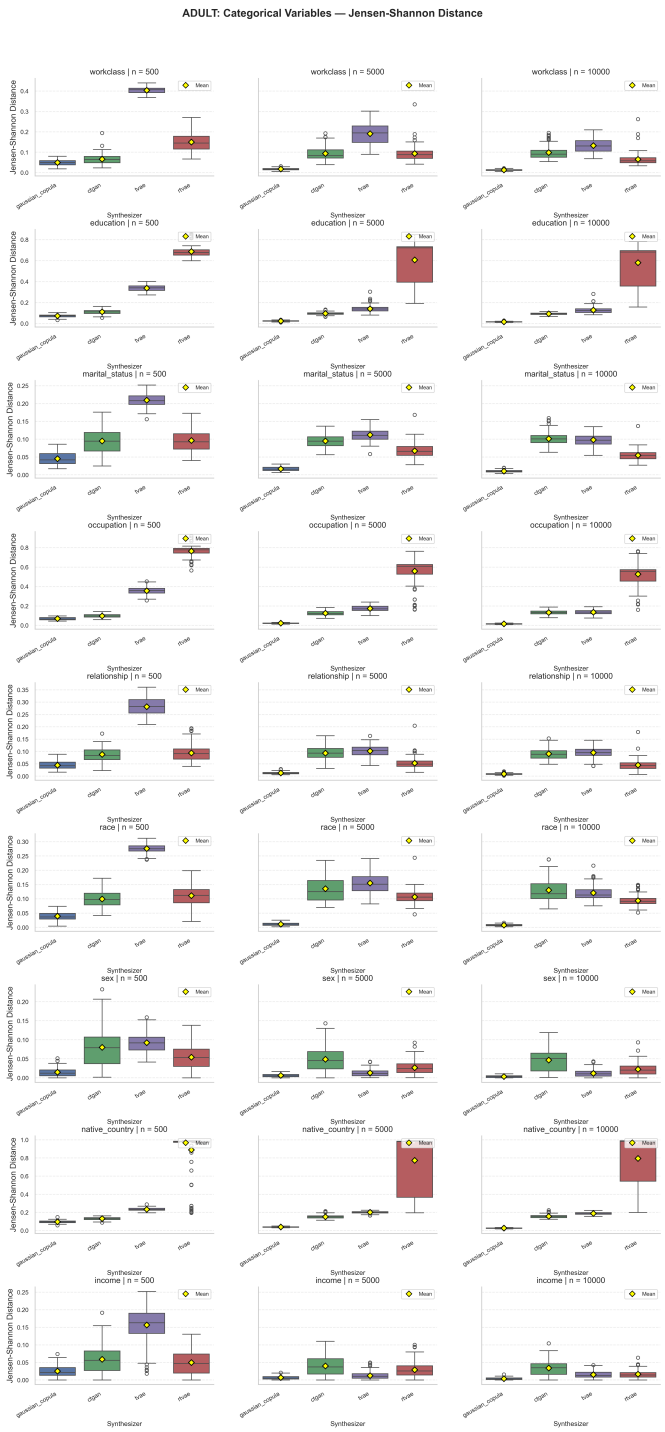

Figure A.2: Jensen Shannon distance results for all variables in the Adult dataset using non-private synthesizers.

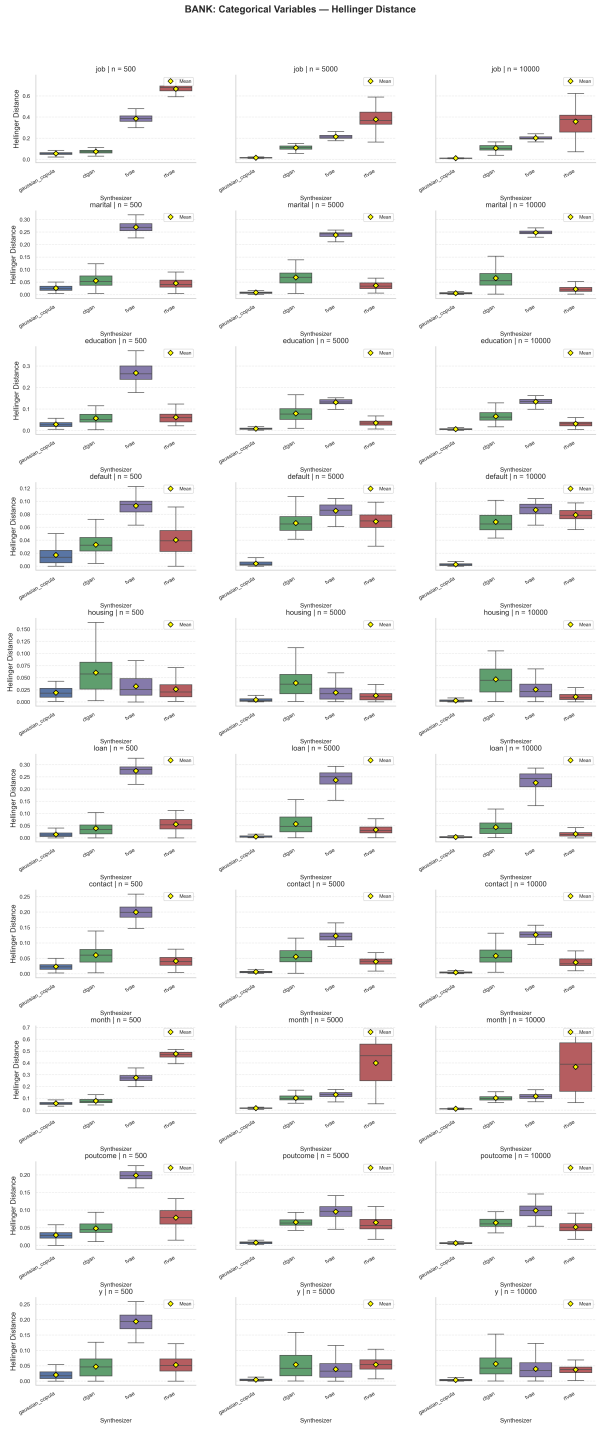

Figure A.3: Hellinger distance results for all variables in the Bank dataset using non-private synthesizers.

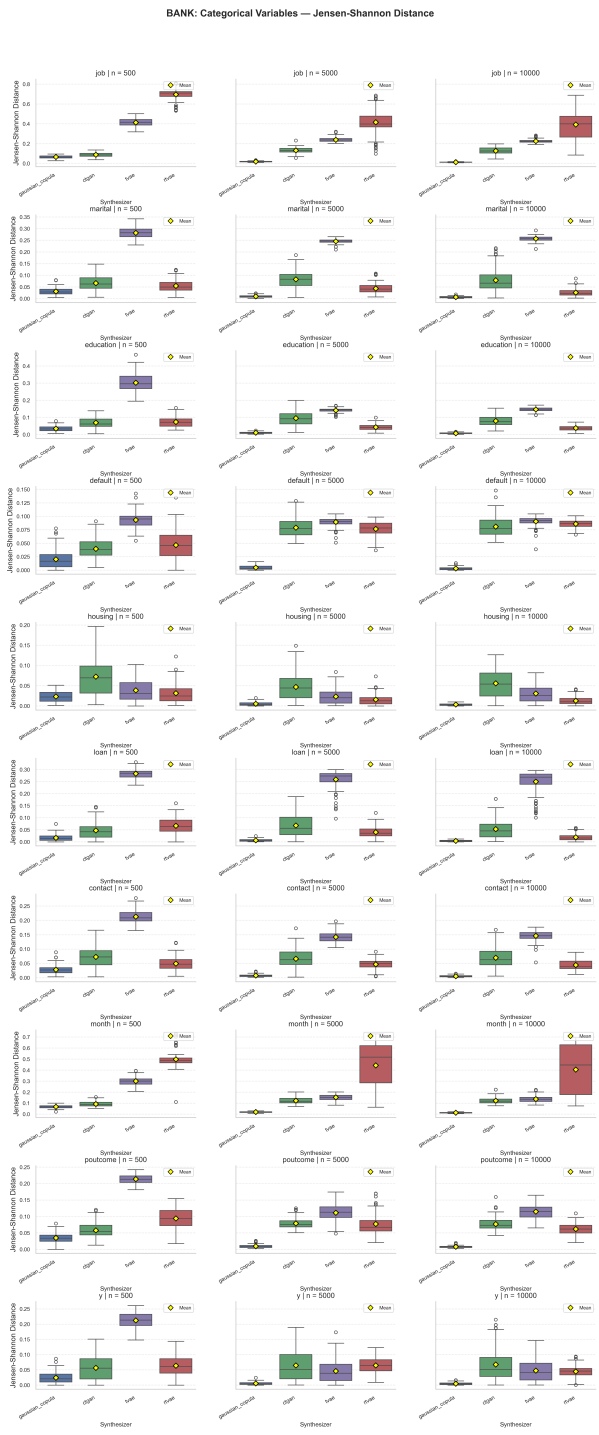

Figure A.4: Jensen–Shannon distance results for all variables in the Bank dataset using non-private synthesizers.

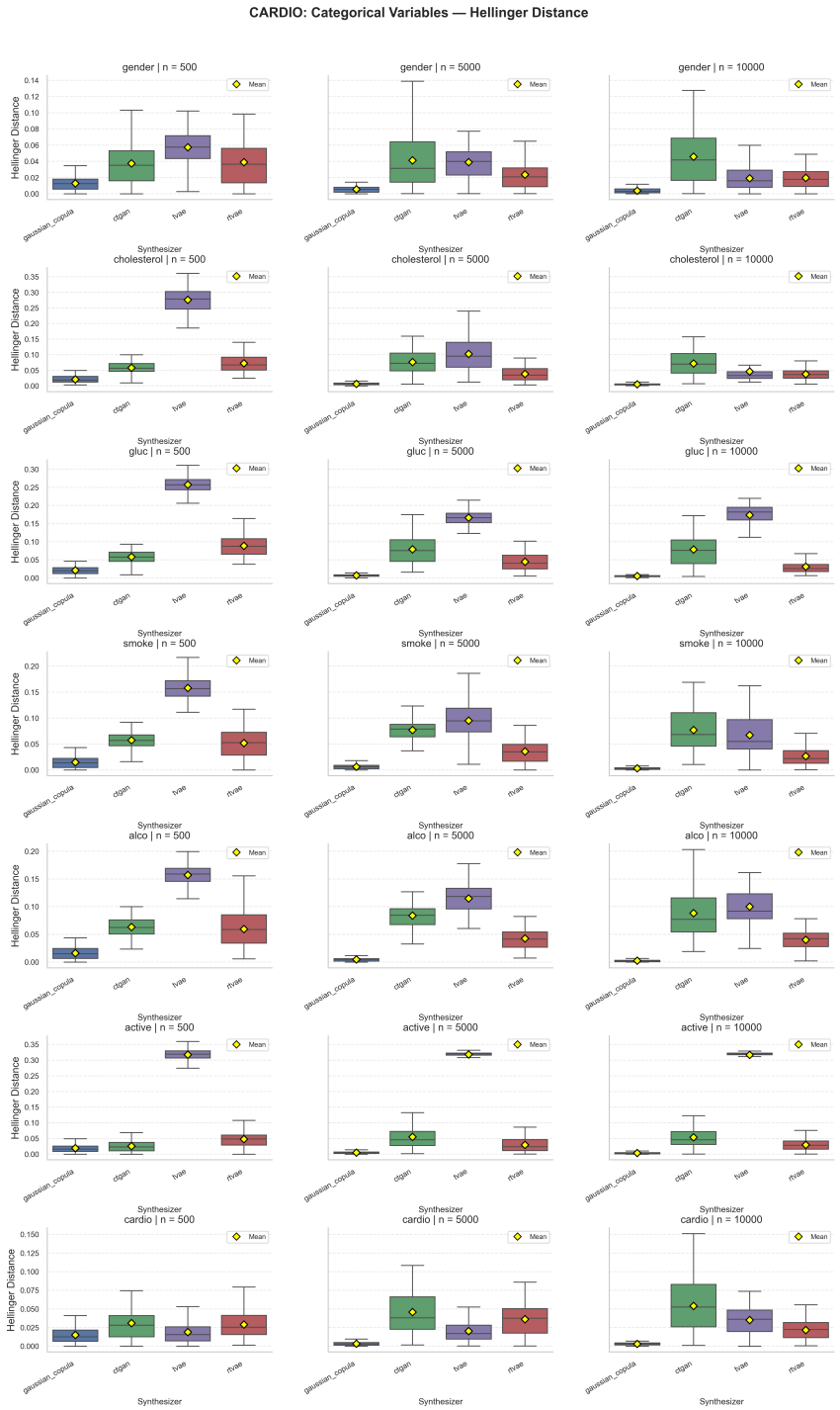

Figure A.5: Hellinger distance results for all variables in the Cardio dataset using non-private synthesizers.

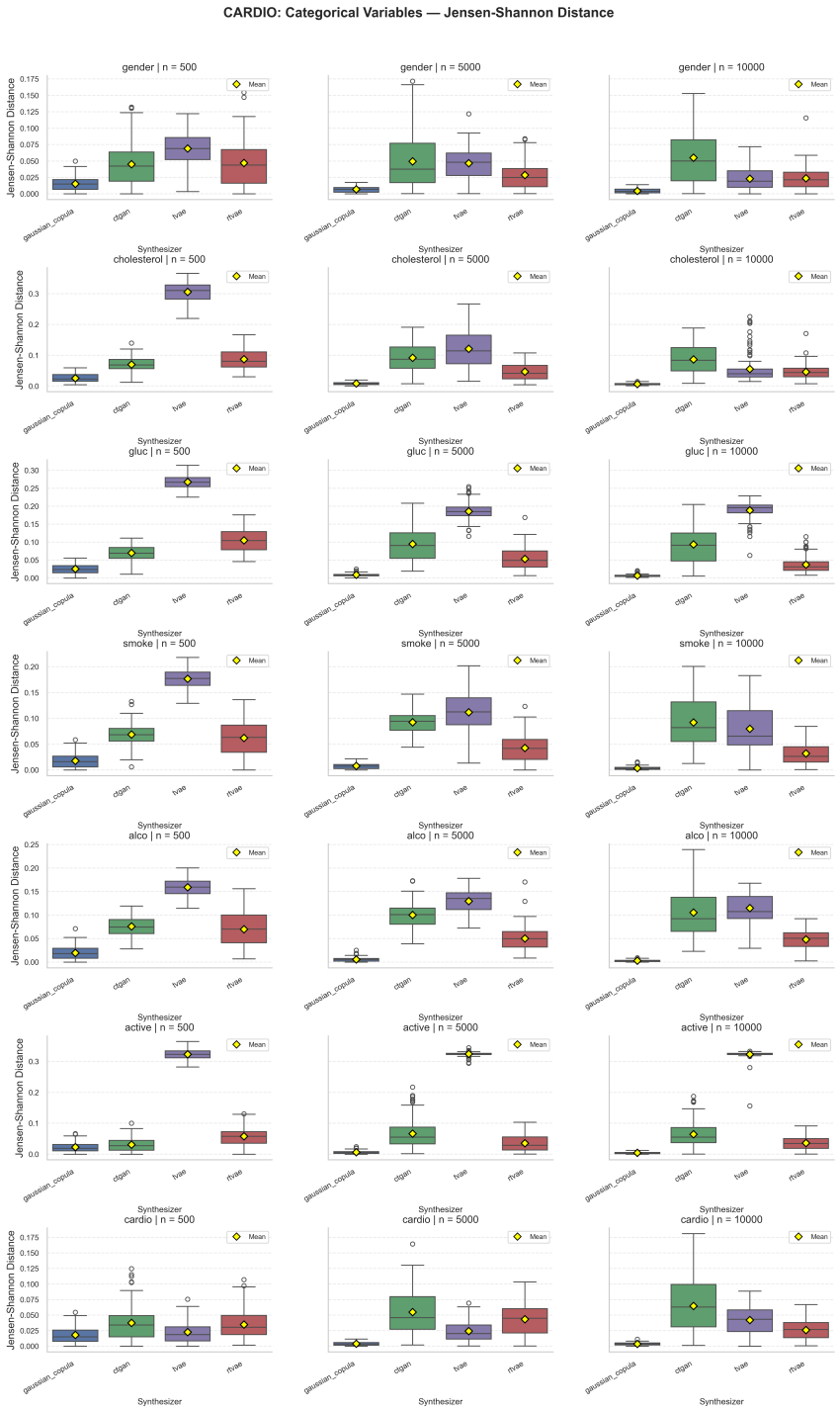

Figure A.6: Jensen Shannon distance results for all variables in the Cardio dataset using non-private synthesizers.