

Dividi et Impera: Enhancing Synthetic Data Fidelity through Data Partitioning

Yordanos Nebiyou Yifru^{1,†}, Salvatore Distefano^{1,*,†}

¹University of Messina

Abstract

Data-intensive application utility is often hampered by data scarcity, quality, heterogeneity, and privacy constraints. This paper introduces a novel, modular framework for synthetic data generation, centered on a preliminary data partitioning stage. The proposed approach decomposes complex datasets into homogeneous subsets, enabling local generative models to capture intricate data characteristics. Our instantiation, Partitioned Gaussian Copula (PGC), combines Hierarchical Variance-Entropy Based Partitioning (HVEP) with Gaussian copulas. Experiments on geometric and real-world tabular datasets demonstrate the PGC effectiveness by machine learning utility and statistical fidelity compared to global models and baselines (CTGAN, TVAE). While PGC shows a slight trade-off in privacy metrics due to localized modeling, its enhanced data representation for complex distributions underscores partitioning as a critical architectural improvement in synthetic data generation.

Keywords

Synthetic Data, Data Quality, Data Balancing, Undersampling, Machine Learning, Privacy, Data Utility, Data Fidelity, Gaussian Copulas, Generative Models.

1. Introduction

In the today data-driven landscape, the development of robust intelligent systems is increasingly challenged by data scarcity, quality deficiencies, and critical privacy concerns. The process of acquiring and annotating real-world data is inherently costly and time-consuming, often yielding insufficient volumes or inconsistent quality that can lead to imprecise machine learning models [1]. Furthermore, stringent regulations like GDPR necessitate privacy-preserving data solutions, while inherent biases in empirical datasets demand methods to ensure algorithmic fairness [2]. Synthetic data emerges as a powerful solution, comprising artificially generated datasets that statistically mimic real data without containing actual sensitive instances. This addresses limitations in volume, quality, balance, and privacy [3].

Synthetic data broadly encompasses partially synthetic data (where sensitive values are replaced), hybrid synthetic data (combining real and synthetic components), and fully synthetic data (entirely artificial datasets preserving statistical properties). Its generation leverages diverse methodologies, including advanced Generative AI Models like GANs, VAEs, and Transformers, alongside statistical modeling and rule-based systems.

The core contribution of this paper lies in introducing a preliminary data partitioning stage within the synthetic data generation process. We propose a novel, general framework that segments datasets into more homogeneous subsets based on statistical or semantic criteria. This methodological shift is central to our approach; rather than training a single, monolithic generative model on an entire, potentially complex dataset, our framework enables the fitting of local generative models to each individual partition. This preliminary partitioning significantly enhances the capacity of existing generative techniques to capture intricate, nonlinear, and localized data dependencies more accurately.

As a concrete instantiation, we apply this framework to Gaussian copulas. This demonstrates a remarkable improvement in their ability to model complex relationships that typically exceed their

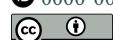
ITADATA2025: The 4th Italian Conference on Big Data and Data Science, September 9–11, 2025, Turin, Italy

*Corresponding author.

†These authors contributed equally.

✉ yordanos.yifru@studenti.unime.it (Y. N. Yifru); salvatore.distefano@unime.it (S. Distefano)

>ID 0000-0003-3448-961X (S. Distefano)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

expressive power, leading to enhanced performance even on challenging data structures. Our approach is characterized by a simple yet effective hierarchical, axis-aligned binary splitting algorithm, guided by variance reduction, which maintains interpretability and computational efficiency. Empirical evaluation across various tabular datasets confirms that this partition-based generation framework yields improved synthetic data utility and competitive privacy guarantees with minimal additional computational overhead, making it a highly effective solution to prevalent data challenges.

2. Synthetic Data

Although synthetic data attracted significant attention in recent years, there is no universally accepted definition in the literature. To cover the diverse range of applications and generation methods, we adopt the following definition inspired by [3].

Definition 1. *Synthetic data is data that has been generated using a purpose-built mathematical model or algorithm, with the aim of solving a (set of) data science task(s).*

Synthetic data is generated by a model, often with the purpose of using it instead of real data.[3] If used responsibly, synthetic data promises to enable learning across datasets when the privacy of the data needs to be preserved; or when data is incomplete, scarce or biased.[3]

One of the most prevalent modalities for synthetic data is image data. Synthetic image data is defined as any image data that is either artificially created by modifying real image data or captured from synthetic environments[4]. The need for robust image data sets for algorithm development and testing has prompted the consideration of synthetic imagery as a supplement to real imagery[5]. Image recognition is one of the most promising applications of synthetic image data. Large-scale annotated data has revolutionized the field of image recognition. However, it is costly and time-consuming to manually collect a large-scale labeled dataset, and recent concerns about data privacy and usage rights further hinder this process[6]. The key advantage of synthesizing image data, and the primary reason that makes the generation of data faster and cheaper, is that a properly set up image synthesis pipeline is capable of automating the data generation and labelling process at a comparatively low cost to manual labour[4].

There are many means to generate synthetic image data with different methodologies suited to different tasks and applications [4] such as Manual Generation, Generative Adversarial Networks, Variational Autoencoders (VAEs), Hybrid Networks, 3D Morphable Models, Parametric Models and so on. VAEs[7] consist of two neural networks and represent a step forward from classical autoencoders. VAE networks have been used in image denoising[8], and image compression [9] but limited for image generation on its own due to its blurry output. Research on the use of VAEs for image generation has transitioned to hybrid models that use both VAEs and GANs[4]. GAN model[10] uses two networks—a generator and a discriminator—that compete adversarially, with the generator aiming to fool the discriminator by producing realistic synthetic data. The more well-known GAN models used to generate image data are BigGAN, CycleGAN, DALL-E 2 (and similar models), DCGAN, and StyleGAN [4]. Overall, GANs have multiple notable use cases in synthetic data generation. The most common use case is to generate data for training other networks[4].

The other modalities for synthetic data is text data. Text generation is the task of automatically generating texts, which maintain specific properties of real texts[11]. Machine-learning-powered text classification models have been widely applied in diverse applications such as detecting biased or toxic language on online platforms and filtering spam emails[12]. The performance of these models is directly related to the volume and quality of the data we have. This poses a huge challenge, as the training data collection and curation process is often costly, time-consuming, and complex[12]. In addition, the issue of privacy has gained increasing attention in natural language processing (NLP)[2]. These make synthetic text generation an increasingly valuable approach, as it allows for the creation of large, diverse, and privacy-preserving datasets that can be used to train and evaluate NLP models effectively.

In the past, models such as latent Dirichlet allocation (LDA), Markov chains (MC) and hidden Markov model (HMM) have been used for generating synthetic text data. However, with the recent advancements

in large language models (LLMs), researchers have started to explore the potential of utilizing LLMs for generating synthetic data tailored to specific tasks and augmenting the training data in low resourced data settings[12].

The final and less explored modalities for synthetic data is tabular data. synthetic tabular data Denotes data that is synthetically produced to replicate the structure and statistical characteristics of real-world tabular data, which is commonly arranged in rows and columns. In this format, columns correspond to features or attributes, such as categorical, numerical, or ordinal variables, while rows represent individual data entries or observations.

When working with scarce or imbalanced datasets generative modeling can be used to augment the data by creating synthetic data points that fill in the gaps. This can help to improve the performance of machine learning models, as they will have more data to train on [13].

Several models have been used for the generation of synthetic tabular data. The traditional approach to the generation of synthetic tabular data includes Bayesian networks[14, 15] and copulas [16, 17]. The modern method for modeling tabular models is based on deep learning. Conditional Tabular Generative Adversarial Network (CTGAN)[18] is a deep learning method to model tabular data. It uses a conditional GAN to capture complex non-linear relationships[19]. [20] introduced the most optimized version of conditional GAN for tabular data. The Tabular Variational Autoencoder (TVAE)[18] is a type of Variational Autoencoder for modeing tabular data.

Synthetic data are being used as a solution to a variety of problems in many domains[3] such as data exchange while preserving privacy, data debiasing for fairness, and data augmentation.

The wide adoption of data-driven machine learning solutions as the prevailing approach to innovate has created a need to share data[3]. When privacy preservation is one of the goal during data sharing, synthetic data offers a potential solution.

Data-driven algorithms are only as good as the data they work with, while data sets, especially social data, often fail to represent minorities adequately[21]. These algorithms may under-perform if trained on pre-existing biases which lay inside data distributions[22]. Showed that data augmentation can reduce classification error for discriminated groups. [23] Further demonstrate the large potential of synthetic data for analyzing and reducing the negative effects of dataset bias on deep face recognition systems.

Data augmentation using synthetic data has emerged as a promising strategy to address challenges related to data scarcity, bias, and fairness in machine learning. By generating realistic yet artificial samples, synthetic data helps expand limited datasets and improves model generalization, especially for underrepresented groups.

3. Proposed Approach

This paper proposes a modular framework for synthetic data generation that combines data partitioning and synthetic data generation models within these partitions to address the challenges of heterogeneity and complexity in datasets. The main idea is to decompose the input dataset into smaller subsets and train dedicated generative models on them, thereby enabling the modeling process to adapt to local data characteristics. The synthetic data generation workflow, shown in Figure 1, relies fundamentally on few stages, i.e., i) exploratory data analysis and preprocessing, ii) partitioning (algorithm selection and application) and iii) synthetic data generation (model selection and application on the partitions). Once generated, the synthetic data quality is assessed, and if it does not meet the quality criteria, the process restarts from partitioning.

3.1. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical initial step that involves understanding the structure, distribution, and quality of the data set before synthetic data generation. By examining data types, identifying patterns or anomalies and assessing diversity and complexity, EDA guides the selection of

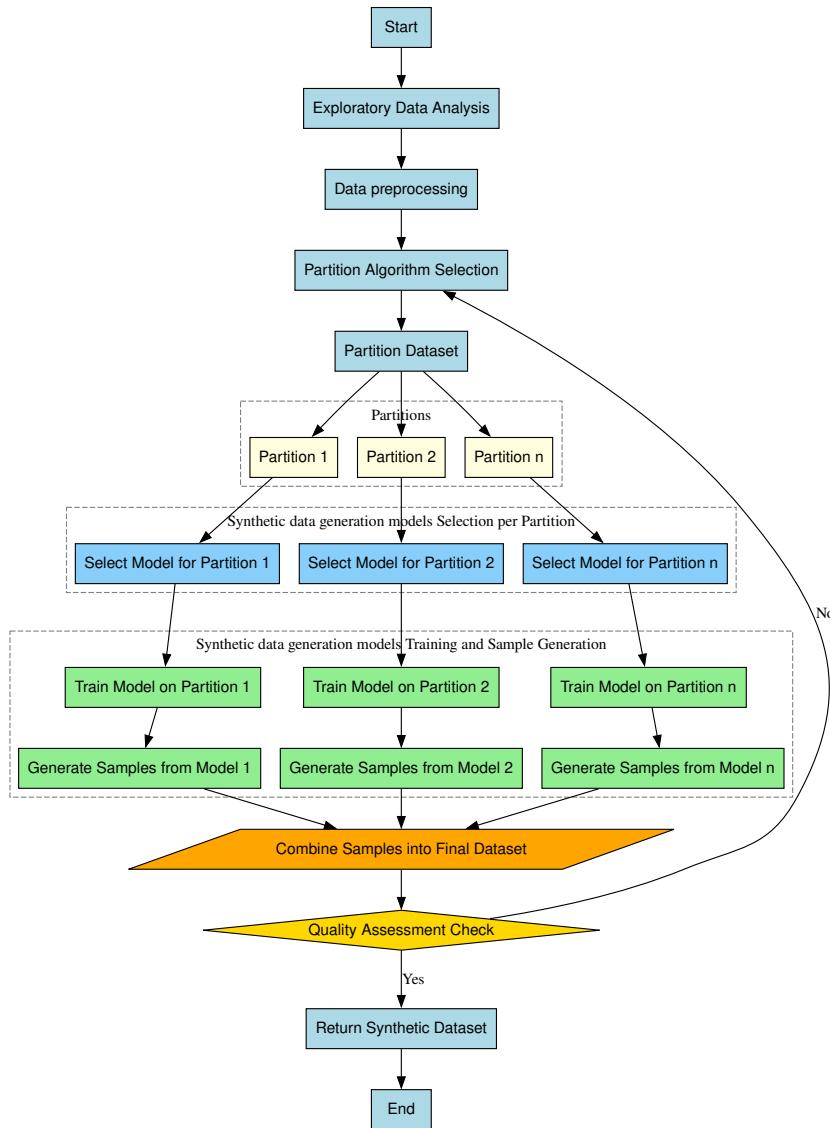


Figure 1: Modular Partition-and-Generate workflow

partitioning strategies and suitable generative models. Skipping this step risks applying inappropriate models that may generate unrealistic or biased synthetic data.

3.2. Data Preprocessing

Data preprocessing prepares the data set for partitioning and modeling by addressing quality and consistency issues. This includes handling missing values, encoding categorical features, detecting and removing outliers, and balancing imbalanced datasets. These steps reduce noise while ensuring that the input data align with the model requirements, ultimately enhancing the fidelity and usefulness of the generated synthetic data.

3.3. Partition algorithm selection

In These step frameworks involve selecting an appropriate partitioning algorithm.

A **Partitioning algorithm** \mathcal{P} can be defined as a function

$$\mathcal{P} : D \rightarrow D_1, D_2, \dots, D_n$$

mapping from the input dataset D to a set of subsets $\{D_1, D_2, \dots, D_n\}$, where $\bigcup_{i=1}^n D_i \subseteq D$ and the number of partitions n can be specified by the user or adaptively determined by the partitioning algorithm based on the data characteristics. Partitioning the dataset D into subsets $\{D_1, D_2, \dots, D_k\}$ aims to isolate regions of the data that exhibit relatively homogeneous characteristics. This simplifies modeling and improves synthetic data quality by allowing models to capture local patterns more effectively.

The choice of a partitioning strategy is highly dependent on the intrinsic properties of the dataset D . Key characteristics influencing this decision include:

- **Data type and modality:** Whether the data is tabular, image-based, text, time-series, or multi-modal can significantly affect the suitable partitioning methods.
- **Statistical properties:** Measures such as variance, entropy, correlation between features, or class imbalance provide insight into the complexity and heterogeneity of the data.
- **Structural characteristics:** The presence of clusters, hierarchical groupings, or other latent structures in the data guides the partitioning approach.
- **Dimensionality:** High-dimensional data may require dimensionality reduction or specialized partitioning algorithms to handle sparsity and noise.

Effective partitioning is central to improving the quality of synthetic data generation, as it enables localized modeling within more homogeneous data subsets. For tabular data, partitions can be created using categorical splits, decision trees, or correlation-based clustering in rank space. These approaches isolate subpopulations with simpler statistical properties, improving model accuracy. Advanced techniques like PCA-based variance reduction and latent space clustering (e.g., via VAEs) further capture complex dependencies by transforming the data into representations where partitions are more meaningful.

For image datasets, partitioning leverages the rich semantic information captured by pretrained CNNs such as ResNet to generate compact embeddings. Clustering in this feature space forms semantically coherent groups suitable for local generation. Hierarchical partitioning or label-based segmentation also helps preserve semantic granularity—e.g., dividing images first by broad classes and then refining into subcategories. When performed in learned latent spaces, clustering can uncover intricate visual structures not easily detected through pixel-level analysis.

Text partitioning utilizes contextual embeddings from transformer models like BERT to group text by semantic similarity. This enables the discovery of topic-based clusters for more focused modeling. Additionally, linguistic features such as syntax or sentence complexity can inform partitions along stylistic or structural lines. Together, semantic and linguistic partitioning strategies produce coherent textual subsets, enhancing the performance and interpretability of generative models tailored to diverse language use cases.

Although the proposed partition-based framework can be applied across diverse data modalities, the choice of partition strategy must be tailored to the structure and semantics of the specific data type. To this end, we introduce a custom *Hierarchical variance-entropy based partitioning* (HVEP) algorithm, specifically designed for tabular data sets. The HVEP algorithm recursively partitions the data into smaller and more homogeneous subsets using axis-aligned binary splits driven by maximizing an aggregated gain metric across all features (e.g., variance reduction for continuous variables or information gain for categorical ones). It builds a binary tree where each internal node corresponds to a split on one feature, and each leaf node represents a final partition that is simple enough to be modeled independently.

3.4. Synthetic data generation

This process includes the main steps for synthetic data generation, and it is driven by a set of generative functions or models \mathcal{M}_i trained independently on each subset D_i and then generating synthetic samples \tilde{D}_i as shown in Figure 1. The \mathcal{M}_i models can be any probabilistic or neural generative model (e.g., Gaussian copula, VAE, GAN), depending on the application requirements. The choice of generative

model in our framework is directly informed by the nature of the partitions created during the data partition phase. Different partitioning strategies yield subsets with varied statistical properties—such as lower variance, more homogeneous correlations, or class-specific features—that enable even simple models to effectively learn from localized data. This piecewise modeling approach leverages the fact that many generative models, particularly those that assume linearity or unimodality, perform poorly on globally complex data but succeed when applied to well-chosen subregions.

For example, Gaussian Copulas assume linear dependencies after Gaussian transformation of marginals and therefore work best on partitions where such linearity holds. Effective partitioning techniques include variance reduction (e.g., PCA + k-means), conditioning on discrete variables (like gender or product category), and tree-based splits that isolate local linear relationships. By aligning the partitions with the copula modeling assumptions, we enable it to capture complex global structures through a composition of locally linear models.

Similarly, Conditional GANs (CGANs) benefit from label-based conditioning, but the effectiveness of this conditioning depends on the granularity of labels. Coarse labels often miss intra-class variation, so our framework introduces hierarchical conditioning, where data is either further clustered within existing labels or pseudo-labeled when no explicit classes exist. This allows CGANs to learn fine-grained conditional distributions, improving diversity and realism of generated data while also helping with class imbalance.

Tabular VAEs (TVAE), while designed for structured data, tend to average out distinct subpopulation behaviors when trained globally. Partitioning the data first—via k-means or Gaussian Mixture Models—allows training of one TVAE per cluster, preserving subgroup-specific dynamics. This ensures that even rare or behaviorally distinct groups are represented accurately in the synthetic data.

3.5. Quality assessment check

After generating synthetic data for each partition—where a separate model is trained per region—we merge the outputs into a unified synthetic dataset. To ensure the overall quality, we apply a post-generation quality assessment step. If the generated data fails to meet the desired quality thresholds, we return to the partition selection phase and refine the partitioning strategy. The specific assessment criteria heavily depend on the intended use case. In privacy-sensitive applications, metrics such as Nearest-Neighbor Distance Ratio (NNDR) or Distance to Closest Record (DCR) can be used to evaluate disclosure risks. These assess how distinguishable synthetic records are from real ones, helping identify potential privacy leaks.

When statistical fidelity is the primary concern, we can measure how well the synthetic data preserve important distributional properties. Marginal distributions can be compared using the Wasserstein distance or Jensen-Shannon divergence, while the correlation distance can be used to assess pairwise dependencies. In use cases where synthetic data are intended to support downstream machine learning tasks, model-centric metrics such as accuracy, F1 score, or AUC can be used. These evaluate whether models trained on synthetic data generalize well to real data, serving as an indirect but practical measure of utility.

4. Case study and experiments

In this section, we present a comprehensive evaluation of our modular synthetic data generation framework. The framework allows for flexible instantiation by combining a partitioning algorithm with a generative model. In the experiments, a Hierarchical Variance-Entropy Based Partitioning (HVEP) algorithm and Gaussian copulas have been adopted as the base generative model. We refer to this instantiation as a Partitioned Gaussian Copula (PGC). For comparison, we also evaluated the performance of a global Gaussian copula (GGC) model that applies the same generative model without partitioning. In addition, we include other standard baselines to provide a broader context and assess general performance across multiple axes.

Our experiments evaluate the effectiveness of partition-based synthetic data generation using Gaussian copulas across two fronts. First, we use geometric synthetic datasets with known nonlinear patterns—such as spirals and trefoils—to visually assess how well different models preserve geometric and topological structures. Second, we apply our approach to four real-world tabular datasets (Adult, Loan, Breast Cancer, and Intrusion) to measure performance across machine learning utility (accuracy, F1-score, AUC), statistical similarity (Wasserstein distance, Jensen-Shannon divergence, correlation distance), and privacy metrics (Nearest Neighbor Distance Ratio and Distance to Closest Record). This comprehensive evaluation reveals the strengths and limitations of our modular framework in capturing complex dependencies while balancing utility, fidelity, and privacy. The synthetic datasets, notebooks, source codes and all the experiments artifacts are available on demand.

4.1. Geometric datasets for structural fidelity

We used two synthetic 3D datasets, spiral and Trefoil Knot, to evaluate structural fidelity. The Spiral dataset consists of two intertwined spiral arms in three-dimensional space (x, y, z), forming a classic nonlinear manifold that challenges models to preserve spatial continuity and curvature. The Trefoil Knot, a single loop knotted curve, introduces a topological challenge because of its non-trivial structure and complex dependencies. Both datasets contain 10,000 samples and serve as visually intuitive benchmarks for assessing a model ability to capture nonlinear and topologically rich patterns.

To evaluate the fidelity of the generated data, we visualized the output of various models: PGC, GGC, CTGAN, TVAE, and CopulaGAN, along with the original 3D data. All neural-based models were trained for 150 epochs. A successful generative model is expected to preserve the global shape, continuity, and density of the original data. Models that fail to do so often suffer from inadequate spatial modeling, limited representational capacity, or overly simplified learning that neglects global structure.

These qualitative observations are supported by quantitative metrics that help to assess how well a model approximates the true data distribution. We report statistical measures including marginal and correlation similarity, Jensen-Shannon divergence (JSD), and Kolmogorov-Smirnov (KS) divergence. Together, these tools offer a comprehensive evaluation of structural fidelity in the generation of synthetic data.

4.1.1. Trefoil

The results shown in Table 1 and Figure 2 indicate that the PGC achieves the lowest Wasserstein distance and the correlation distance, suggesting a superior preservation of both marginal distributions and feature dependencies in the Trefoil dataset. GGC or neural-based generative models show higher distances, reflecting a less accurate fit to the complex structure of the data. This highlights the effectiveness of partitioning strategies in improving synthetic data quality for complex non-linear datasets.

Table 1
Statistical Metrics on Trefoil Dataset

Model	Average WD	Average JSD	Correlation Distance
PGC	0.004	-	0.01
GGC	0.02	-	0.06
CopulaGAN	0.02	-	0.08
CTGAN	0.04	-	0.03
TVAE	0.02	-	0.05

4.1.2. Spiral

The spiral dataset results shown in Table 2 and Figure 3 confirm the PGC effectiveness, which achieves the lowest Wasserstein distance and the best preservation of correlations, matching the GGC but outperforming it in marginal distribution fidelity. Neural-based models such as CopulaGAN, CTGAN,

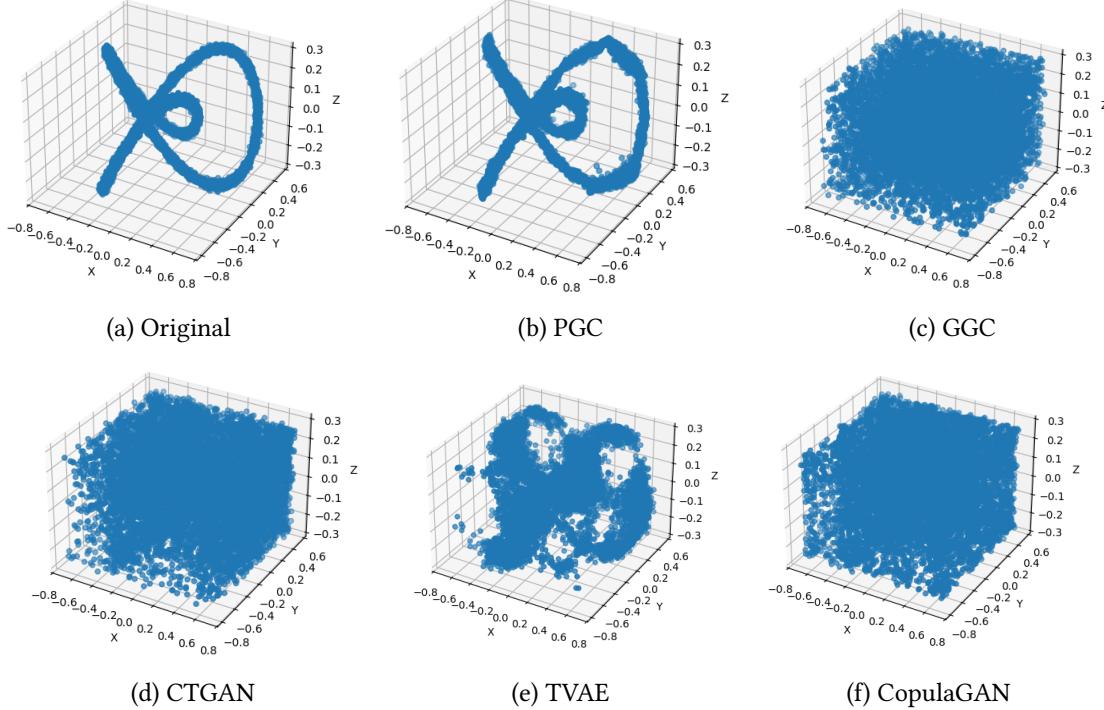


Figure 2: Trefoil Dataset 3D Visual Comparison of Real and Synthetic Samples

and TVAE exhibit higher correlation distances, indicating challenges in accurately capturing the complex dependencies inherent in the spiral structure. In general, these findings underscore the benefits of partitioning for modeling intricate non-linear data distributions.

Table 2
Statistical Metrics on Spiral Dataset

Model	Average WD	Average JSD	Correlation Distance
PGC	0.003	-	0.04
GGC	0.01	-	0.04
CopulaGAN	0.02	-	0.47
CTGAN	0.05	-	0.63
TVAE	0.017	-	0.11

4.2. Real-World Datasets for Utility, Privacy, and Statistical Similarity

We evaluated our method on four real-world tabular datasets. UCI Adult, Breast Cancer Wisconsin, Personal Loan, and KDD Cup 1999. These datasets differ in size and dimensionality, offering a broad testbed for assessing generative performance. The UCI Adult dataset¹ contains 48,842 records and 14 attributes, and consists of anonymous people information such as occupation, age, native country, race, capital gain, capital loss, education, work class and more. The Breast Cancer Wisconsin dataset², with 569 rows and 30 features computed from a digitized image of a fine needle aspirate of a breast mass describing characteristics of the cell nuclei present in the image. The Personal Loan dataset³ includes 5,000 records and 12 attributes, including customer demographic information (age, income,

¹<https://www.kaggle.com/datasets/sagnikpatra/uci-adult-census-data-dataset>

²<https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic>

³<https://www.kaggle.com/datasets/teertha/personal-loan-modeling>

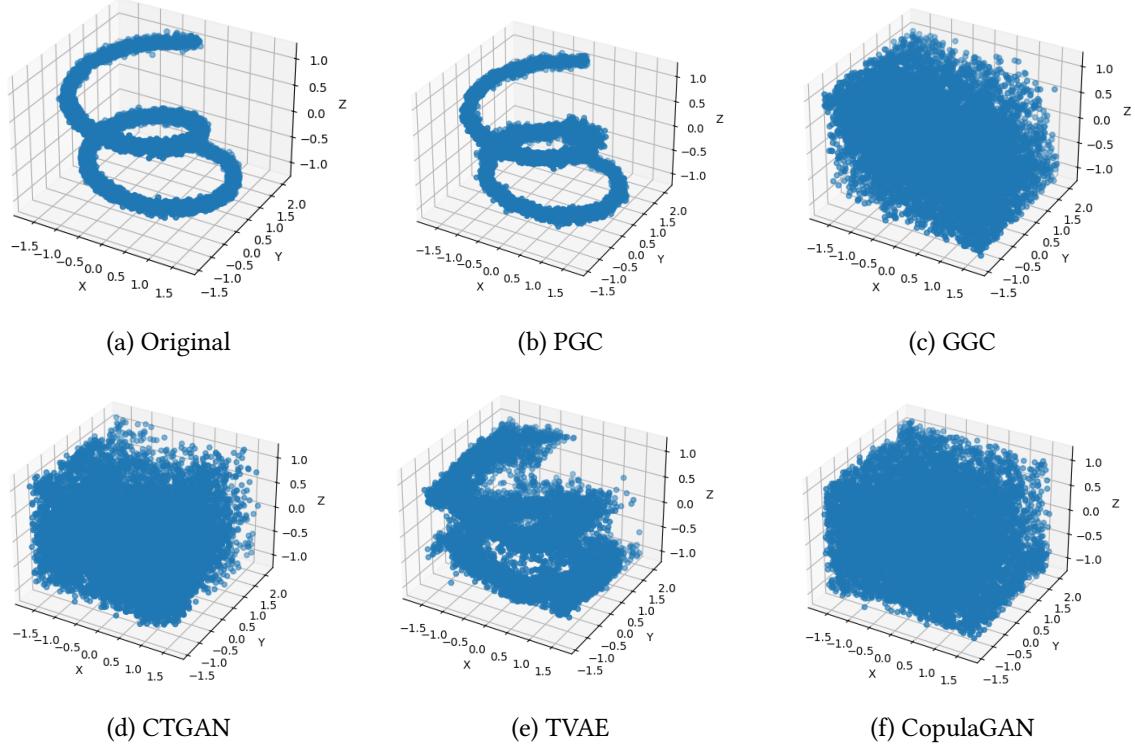


Figure 3: Spiral Dataset 3D Visual Comparison of Real and Synthetic Samples

etc.), relationship with the bank (mortgage, securities account, etc.), and response to the last personal loan campaign. The KDD Cup 1999 dataset⁴, comprising 50,000 records and 41 features, describing network traffic for intrusion detection. The number of training epochs is chosen based on the size and complexity of each dataset, allowing fair comparison between models by ensuring sufficient training without overfitting or underfitting.

For each dataset, we compare the partitioned Gaussian copula (PGC) against its counterpart without partition (GGC). We benchmark it also against established baselines, including CTGAN, TVAE, and CopulaGAN.

4.2.1. Assessment metrics

The evaluation is conducted on three dimensions: (1) machine learning (ML) utility, (2) statistical similarity and (3) privacy preservability. Each dimension offers insights into different aspects of data quality and risk. We assess the **machine learning utility** of the synthetic data by comparing how well it supports downstream model training in comparison to the real dataset. In short, we train standard classifiers on both the real and synthetic training sets and evaluate them on a shared real test set. This setup allows us to observe whether models trained on synthetic data can generalize similarly to those trained on original data. Performance is reported using **accuracy**, **F1-score**, and **AUC**, enabling a fair and interpretable comparison. To perform this evaluation, we first partition the original data set into two parts: training and testing. The training portion is then used as input to the synthetic data generation model, which produces a dataset of equal size, but without exposing real records. We then train 5 machine learning algorithms, decision trees, logistic regression, and MLP, SVM and Random Forest independent of both real and synthetic training sets. In both cases, the evaluation is performed on the same real test set. This setup provides direct insight into whether the knowledge learned from synthetic data is transferable to real-world data distributions. If model performance on the synthetic-trained models is close to the real-trained ones, the generated data are considered highly useful.

⁴<https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

To evaluate the **statistical similarity**, we compute multiple distance-based metrics between the synthetic and real data distributions. For continuous variables, we use the **Wasserstein distance** to quantify how well the empirical distributions align. For categorical attributes, we apply **Jensen-Shannon Divergence**, a symmetric and bounded variant of KL divergence, which assesses the overlap between probability distributions. Furthermore, we measure the **correlation distance** between the feature pair relationships by comparing the absolute differences between the two correlation matrices.

To ensure privacy is not compromised, we use two proximity-based metrics. The **Nearest-Neighbor Distance Ratio (NNDR)** compares how close each synthetic point is to its closest real neighbor versus its closest synthetic neighbor. Higher ratios suggest that synthetic points are more embedded within the synthetic distribution, reducing memorization risk. The **distance from the closest record (DCR)** captures the minimum distance from each synthetic sample to any real data point, with higher values indicating better privacy protection. Together, NNDR and DCR offer a robust assessment of whether synthetic data avoid overly replicating or leaking sensitive information from the original dataset.

The experiment is run 3 times and the average is taken to avoid random fluctuations or measurement noise affecting the results.

4.2.2. Results

Table 3

Average ML Utility Metrics (Accuracy, F1 Score, AUC) Across All Datasets

Model	Accuracy (%)	F1 Score	AUC
PGC	2.96	0.0320	0.0180
CTGAN	6.39	0.1118	0.0800
GGC	31.19	0.4600	0.1690
CopulaGAN	7.00	0.1600	0.0350
TVAE	11.80	0.1500	0.0360

Table 3 shows the averaged differences in ML utility between real and synthetic data in terms of accuracy, F1 score, and AUC. Better synthetic data are expected to have low differences. The results confirm that partitioning enhances the utility of generative models. The PGC model outperformed its counterpart the GGC achieving an average gain of approximately + 43 percentage points (pp) in F1. This substantial improvement highlights the strong compatibility of Gaussian copulas with partitioning. PGC also outperformed well known GAN and VAE based baselines which shows how effective partitioning can be.

Table 4

Average Statistical Metrics Across All Datasets

Model	Average WD	Average JSD	Correlation Distance
PGC	0.0088	0.0249	1.0137
CTGAN	0.0631	0.1024	3.8806
GGC	0.0918	0.0923	2.4465
CopulaGAN	0.0870	0.1057	4.1020
TVAE	0.0462	0.0875	1.9061

Table 4 summarizes statistical similarity metrics - averaged across all data sets. We can see that the PGC consistently achieves the best performance in all statistical metrics, indicating that partitioning preserves the marginal and correlation structure most effectively in this setting.

Table 5 presents privacy evaluation metrics. These metrics assess how distinguishable synthetic records are from real ones - higher values generally suggest better privacy. As expected, the experimental results show that PGC generally exhibits less privacy compared to GGC, since data partitioning often

Table 5

Average Comparison of Models on DCR and NNDR Metrics Between Real and Synthetic Data

Model	DCR	NNDR
PGC	1.3620	0.6924
CTGAN	1.5795	0.7529
GGC	1.6256	0.7254
CopulaGAN	1.4146	0.7592
TVAE	1.4883	0.7936

leads to smaller training subsets, which can increase the risk that synthetic samples resemble real records too closely.

5. Conclusions

This work addresses key challenges in synthetic data generation by proposing a modular framework incorporating a preliminary data partitioning stage. This core contribution enhances generative model capacity by allowing local modeling of homogeneous data subsets. The Partitioned Gaussian Copula (PGC) instantiation, leveraging HVEP and Gaussian copulas, consistently demonstrated superior machine learning utility and statistical fidelity on both complex geometric and diverse real-world tabular datasets, significantly outperforming Global Gaussian copulas (GGC) and established baselines. This confirms that partitioning effectively enables generative models to capture intricate, nonlinear data dependencies. While localized modeling in PGC led to an expected, marginal trade-off in privacy metrics compared to global approaches, the substantial gains in utility and fidelity underscore the framework effectiveness. Future research will integrate formal differential privacy mechanisms within partitions and explore adaptive partitioning for various data modalities.

Acknowledgments

This work has also been supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, project 3D-SECSDE, CUP J33C22002810001, partnership on “SEcurity and RIghts in the CyberSpace” (PE000000014 - program “SERICS”).

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT for grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication content.

References

- [1] A. Figueira, B. Vaz, Survey on synthetic data generation, evaluation methods and gans, *Mathematics* 10 (2022). URL: <https://www.mdpi.com/2227-7390/10/15/2733>. doi:10.3390/math10152733.
- [2] X. Yue, H. A. Inan, X. Li, G. Kumar, J. McAnallen, H. Shajari, H. Sun, D. Levitan, R. Sim, Synthetic text generation with differential privacy: A simple and practical recipe, *arXiv preprint arXiv:2210.14348* (2022).
- [3] J. Jordon, L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, S. N. Cohen, A. Weller, Synthetic data—what, why and how?, *arXiv preprint arXiv:2205.03257* (2022).

- [4] K. Man, J. Chahl, A review of synthetic image data and its use in computer vision, *Journal of Imaging* 8 (2022) 310.
- [5] J. R. Schott, S. D. Brown, R. V. Raqueno, H. N. Gross, G. Robinson, An advanced synthetic image generation model and its application to multi/hyperspectral algorithm development, *Canadian Journal of Remote Sensing* 25 (1999) 99–111.
- [6] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, X. Qi, Is synthetic data from generative models ready for image recognition?, *arXiv preprint arXiv:2210.07574* (2022).
- [7] D. P. Kingma, M. Welling, et al., Auto-encoding variational bayes, 2013.
- [8] D. Im Im, S. Ahn, R. Memisevic, Y. Bengio, Denoising criterion for variational auto-encoding framework, in: *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [9] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, Variational image compression with a scale hyperprior, *arXiv preprint arXiv:1802.01436* (2018).
- [10] I. Goodfellow, et al., Generative adversarial nets, *Advances in neural information processing systems* 27 (2014).
- [11] U. Maqsud, Synthetic text generation for sentiment analysis, in: *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2015, pp. 156–161.
- [12] Z. Li, H. Zhu, Z. Lu, M. Yin, Synthetic data generation with large language models for text classification: Potential and limitations, *arXiv preprint arXiv:2310.07849* (2023).
- [13] D. Manousakas, S. Aydöre, On the usefulness of synthetic tabular data generation, *arXiv preprint arXiv:2306.15636* (2023).
- [14] J. Young, P. Graham, R. Penny, Using bayesian networks to create synthetic data, *Journal of Official Statistics* 25 (2009) 549–567.
- [15] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, X. Xiao, Privbayes: Private data release via bayesian networks, *ACM Transactions on Database Systems (TODS)* 42 (2017) 1–41.
- [16] S. Kamthe, S. Assefa, M. Deisenroth, Copula flows for synthetic data generation, *arXiv preprint arXiv:2101.00598* (2021).
- [17] D. Meyer, T. Nagler, R. J. Hogan, Copula-based synthetic data augmentation for machine-learning emulators, *Geoscientific Model Development* 14 (2021) 5205–5215.
- [18] L. Xu, M. Skouliaridou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, *Advances in neural information processing systems* 32 (2019).
- [19] L. Hansen, N. Seedat, M. van der Schaar, A. Petrovic, Reimagining synthetic tabular data generation through data-centric ai: A comprehensive benchmark, *Advances in neural information processing systems* 36 (2023) 33781–33823.
- [20] Z. Zhao, A. Kunar, R. Birke, H. Van der Scheer, L. Y. Chen, Ctab-gan+: Enhancing tabular data synthesis, *Frontiers in big Data* 6 (2024) 1296508.
- [21] N. Shahbazi, Y. Lin, A. Asudeh, H. Jagadish, Representation bias in data: A survey on identification and resolution techniques, *ACM Computing Surveys* 55 (2023) 1–39.
- [22] V. Iosifidis, E. Ntoutsi, Dealing with bias via data augmentation in supervised learning scenarios, *Jo Bates Paul D. Clough Robert Jäschke* 24 (2018).
- [23] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster, T. Vetter, Analyzing and reducing the damage of dataset bias to face recognition with synthetic data, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.