TVineSynth: A Truncated C-Vine Copula Generator of Synthetic Tabular Data to Balance Privacy and Utility

Elisabeth Griesbauer
Norwegian Computing Center,
University of Oslo,
Integreat - Norwegian Centre for Knowledge-driven Machine Learning

Claudia Czado
Technical University of Munich,
Munich Data Science Institute

Arnoldo Frigessi
University of Oslo,
Integreat - Norwegian Centre for Knowledge-driven Machine Learning

Ingrid Hobæk Haff
University of Oslo,
Integreat - Norwegian Centre for Knowledge-driven Machine Learning

Abstract

We propose TVineSynth, a vine copula based synthetic tabular data generator, which is designed to balance privacy and utility, using the vine tree structure and its truncation to do the trade-off. Contrary to synthetic data generators that achieve DP by globally adding noise, TVineSynth performs a controlled approximation of the estimated data generating distribution, so that it does not suffer from poor utility of the resulting synthetic data for downstream prediction tasks. TVineSynth introduces a targeted bias into the vine copula model that, combined with the specific tree structure of the vine, causes the model to zero out privacy-leaking dependencies while relying on those that are beneficial for utility. Privacy is here measured with membership (MIA) and attribute inference attacks (AIA). Further, we theoretically justify how the construction of TVineSynth ensures AIA privacy under a natural privacy measure for continuous sensitive attributes. When compared to competitor models, with and without DP, on simulated and on real-world data, TVineSynth achieves a superior privacy-utility balance.

1 INTRODUCTION

The availability of diverse, high-quality data has led to tremendous advances in science, technology and society at large, when analysed by means of statistical and machine learning (ML) methods. However, real-world data are in many cases limited, imbalanced or cannot be made public to the research community due to privacy restrictions, obstructing progress especially in bio-medical research. Synthetic data can augment the real data, and as long as they do not disclose private aspects, they can also substitute sensitive real data. Both substituting and augmenting real with synthetic data have proven to be successful in training downstream ML applications (Gao et al., 2023; Morales-García et al., 2023; Shetty et al., 2023; Wang et al., 2023a; Jain et al., 2023; Pezoulas et al., 2023; Ye-Bin et al., 2023; Goldschmidt et al., 2023; Wang et al., 2023b; Saisho et al., 2023; Schaufelberger et al., 2023).

We focus on scenarios where the main concern about the real tabular data is privacy and the downstream ML application is classification or regression. Our work is motivated by two objectives: (i) The synthetic data should retain joint dependence and marginal behavior of the real data, so that a regression method trained on the synthetic data performs comparably well on unseen data as it would have done if trained on the real data (**utility**); (ii) the generative model should not leak sensitive information on an instance of the real data into the synthetic data (**privacy**). While differential privacy (DP) (Dwork et al., 2014) provides a sound approach for privacy preserving generative modelling, the resulting synthetic data have shown to score poorly in terms of utility (Jayaraman and Evans, 2019; Bagdasaryan et al., 2019; Cheng et al., 2021). On the other hand, popular generative models without privacy guarantees, such as generative adversarial networks (GANs) (Goodfellow et al., 2014) or variational autoencoders

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

(VAEs) (Kingma and Welling, 2013), tend to generate realistic, but privacy violating synthetic data (Chen et al., 2020; van Breugel et al., 2023; Andrei et al., 2023). Further, their training is data intensive, making them inappropriate as synthetic data generators for small and moderately sized real data sets.

Contributions We propose TVineSynth, a vine copula based synthetic data generator, designed to balance privacy against utility. In contrast to globally adding noise, as is done to obtain DP guarantees, TVineSynth approximates the data generating distribution by (1) setting the focus of the model on dependencies that are relevant for the prediction task and (2) introducing a targeted bias into dependencies that would otherwise leak sensitive information into the synthetic data.

This is achieved as follows: We propose an algorithm to re-order the features[1] in the real data, to obtain a block structure dependence. Then, we set the vine tree structure such that we achieve (1) and truncate away tree by tree from the vine copula, cutting off privacy leaking dependencies to achieve (2). Re-ordering the real data to obtain a block structured dependence is central in TVineSynth as it amplifies the effect of truncation on privacy, thus making it easier to find a suitable vine tree structure that also keeps high utility.

We conduct an in depth analysis of the privacy of the TVineSynth generated synthetic data under membership and attribute inference attacks (Shokri et al., 2017; Yeom et al., 2018) and of their utility w.r.t. prediction performance. We asses AIA privacy with the mean absolute β-coefficient (MAB), a measure for AIA privacy, that naturally builds on the implementation of AIA attacks by Stadler et al. (2022) and addresses the weaknesses of previously used measures. We theoretically justify the construction of TVineSynth by showing how the truncation of the vine copula and the order of the covariates in the vine tree structure ensure AIA privacy under the MAB. TVineSynth's privacy and utility are compared with those of other generative models with and without privacy guarantees. We show that if privacy *and* utility matter, TVineSynth is preferable over private and non-private competitors.

Why do we not use DP? TVineSynth does not use the concept of DP in its model design. We argue that it is common that the data holder wants to protect specific sensitive features, while regarding the protection of the remaining features as less important. Contrary to how we design TVineSynth, this knowledge about the real data is not exploited in favor of either utility or of privacy when noise is added uniformly on (statistics of) all features to obtain DP guarantees. Real-world medical data is highly complex and inherently noisy and preserving its joint distribution is critical for decision making, which makes the application of DP less suitable in the medical domain. There the most relevant risk to analyse is the risk of identifying a finite set of real patients from a finite synthetic data set generated, which DP does neither address nor provide theoretical bounds for. Finally, DP 'fails to address ethical concerns pertaining to the risk benefit ratio, where minimal risk may be deemed allowable if the societal benefits' from more rapid development of medical treatments are high, (Yoon et al., 2020). DP offers theoretical bounds on the effect of substituting a single training data point on the probability of observing an outcome of an algorithm. However, these bounds become weak to meaningless when a privacy budget is chosen, that is non-prohibitive to utility (Stock et al., 2022). While Ziller et al. (2024) claim that for image data a meaninglessly high ϵ provides sufficient protection against relaxed but realistic privacy attacks, we argue that their results cannot directly be transferred to tabular data and a worst-case privacy assessment through MIAs is indispensable in the medical domain. On top of that, the bounds provided by DP are hard to interpret for real-world applications and risks. Data protection laws, such as the GDPR (GDPR, 2016), do not build on DP, which further highlights the problem of translating DP into practice, (Yoon et al., 2020). While DP translates into a theoretical lower bound on MIA privacy gain (PG) (Yeom et al., 2018), empirically TVineSynth achieves a PG comparable to the DP competitors due to the MLE's robustness in TVineSynth. No theoretical bounds on AIA success have been developed w.r.t. DP yet, and we show that through its model design, TVineSynth can handle this attribute specific risk on par with its DP competitors.

Related Work Xu et al. (2019) extend GANs (Goodfellow et al., 2014) and VAEs (Kingma and Welling, 2013) with a conditional generator and specific preprocessing to obtain their counterparts for tabular data, namely CTGAN and TVAE. Kotelnikov et al. (2023) adapt denoising diffusion probabilistic models to model tabular data. These approaches model the real data closely, but do not exploit model structure to achieve privacy like TVineSynth. Taking privacy into account, Jordon et al. (2018) (PATE-GAN), Xie et al. (2018) (DP-GAN) and Zhang et al. (2017) (PrivBayes) modify non-private GANs and Bayesian networks to fulfill DP, while Donhauser et al. (2024) utilize a particle based approach on privatized marginals (PrivPGD). These models add noise in a global fashion in order to guarantee DP, contrasting the precise model approximation through truncation of a vine copula in TVineSynth.

[1] The terms feature and covariate are used interchangeably in the following.

Another line of work generates synthetic data with copulas, such as Gaussian copulas (Patki et al., 2016; Kumi et al., 2023), Student's t-copulas (Benali et al., 2021), an empirical beta copula as latent space distribution in a pre-trained autoencoder (Coblenz et al., 2023), a DP Gaussian copula by applying DP marginal histograms and correlation matrix (Li et al., 2014) or a copula estimated with normalizing flows (Kamthe et al., 2021). These copulas lack the flexibility of the vine copula and are not tailored towards the privacy needs of the data holder. Generative modeling with vine copulas naturally builds on this line of work. While Chu et al. (2022b) limit themselves to C- and D-vines, Meyer et al. (2021) utilize an R-vine copula for data generation. Moreover, Sun et al. (2019) re-formulate the structure selection in an R-vine as a reinforcement learning problem to increase modelling flexibility, Tagasovska et al. (2019) model high dimensional data using a vine copula as latent space distribution in an autoencoder and Gambs et al. (2021) obtain a DP vine copula by applying DP marginal histograms. While these generative models benefit from the flexibility of vine copulas, they do not balance utility with privacy, as we do by exploiting the vine structure and truncation. Patki et al. (2016) and Qian et al. (2023) offer implementations of several SOTA generative models and evaluation metrics, while Meyer and Nagler (2021) focus specifically on vine copulas, with no attention to privacy.

2 METHODS

2.1 Vine Copula Based Synthetic Data Generation

A vine copula[2] is a probabilistic model that builds on copulas: A d-dimensional copula $C : [0,1]^d \to [0,1]$ is a d-dimensional distribution on the unit cube with uniform marginals and corresponding copula density c. Sklar's theorem (Sklar, 1959) states that any multivariate distribution F can be expressed in terms of a copula C; and if all densities exist, a multivariate density f can be expressed as a product of the corresponding copula density c and marginal densities. Vine copulas (Joe, 1997; Bedford and Cooke, 2001, 2002; Aas et al., 2009; Joe, 2014; Czado, 2019) are hierarchical probabilistic graphical models constructed from univariate distributions and bivariate (conditional) copulas. The vine tree structure $\mathcal{V} = (T_1, \ldots, T_{d-1})$, which is a nested sequence of $d-1$ trees $T_k = (V_k, E_k),\ k \in [d-1]$, serves as a construction plan of the vine copula. Here an edge in T_1 represents a bivariate copula c_{a_e,b_e} of the unconditional pair of random variables $(X_{a_e}, X_{b_e}),\ a_e, b_e \in [d]$, and an edge e in $T_k,\ k \in \{2, \ldots, d-1\}$ represents a bivariate copula $c_{a_e,b_e;D_e}$ of a pair (X_{a_e}, X_{b_e}), conditioned on $k-1$ random variables $X_j,\ j \in D_e \subset [d]$. Taking $\mathcal{V}$ and the pair copulas together, the d-dimensional copula density c can be expressed as a product of (conditional) pair copulas over the edges of the trees in $\mathcal{V}$ giving the *vine copula* $c = \prod_{k\in[d-1]} \prod_{e\in E_k} c_{a_e,b_e;D_e}$.

With Sklar's theorem (Sklar, 1959) the full joint density is then $f = c \cdot f_1 \cdots f_d$. The structure of a vine copula and the corresponding conditioning sets are by construction such that computing the vine copula is iterative along the trees and thus efficient. The univariate margins and the pair copulas can be chosen freely. This makes vine copulas a highly flexible, yet tractable model class, that allows to capture complex dependence structures. A way to reduce a vine copula's capacity to approximate a multivariate distribution, is to truncate the vine copula at a specific tree level $t \in [d-1] := \{1, 2, ..., d-2, d-1\}$. This is equivalent to setting all pair copulas of trees $T_{t+1}, \ldots, T_{d-1}$ to independence.

Definition 2.1 (Truncation of the Vine Copula at Level t)**.** *Let c be a vine copula as given above. We define the vine copula truncated at truncation level $t \in [d-1]$ as:* $\prod_{k\in[t]} \prod_{e\in E_k} c_{a_e,b_e;D_e}$.

Thus, in the resulting vine copula, only trees $T_1, ...T_t$ are left in the model. For $t = d-1$, we obtain the un-truncated vine copula, while for $t = 1$, only the first tree is retained. Special shapes of trees in the vine tree structure lead to certain sub-classes of vines. In particular, in a C-vine, each tree is star-shaped, i.e. contains a fully connected node called root node.

We use vine copulas to generate synthetic data to substitute the private real data in a general regression setting with response variable Y. The synthetic data generated for this case should not leak sensitive information about any real observation (privacy) and at the same time allow training a regression method equally well as would happen on the real data (utility). Our idea is to strike a balance between privacy and utility of the vine copula generated synthetic data, by exploiting weak stochastic dependencies of sensitive covariates with the covariates that are important for the prediction task. Thus, it might not be necessary to protect all covariates equally well by adding noise in a global fashion (which decreases utility), or to capture all dependencies present in the real data (which might impair privacy). Based on these considerations we propose TVineSynth, a framework building on a star-shaped C-vine copula with Y as root node of T_1, to focus early specifically on those dependencies that matter for the prediction task. Further, TVineSynth finds an order $\mathcal{O}^*$ of the d covariates in which they

[2] For an extended introduction to vine copulas please consult Appendix A.

are arranged in the remaining trees of the C-vine, that yields a block structured dependence. Truncation of the resulting C-vine then cuts off privacy leaking dependencies, while maintaining high utility. By truncating the vine copula at a moderate tree level t, we cut away dependencies that might not add to utility, but challenge privacy. Combining the C-vine structure with the appropriate order of the covariates and truncation of the vine copula model in TVineSynth, we obtain a generative model that can be tailored towards the desired privacy and utility requirements for the real data.

2.2 TVineSynth Construction

The construction of TVineSynth consists of three steps:

(1) Execute Algorithm 1[3] to determine the order $\mathcal{O}^*$ in which the covariates of the real data enter the C-vine copula.

(2) For the specific order $\mathcal{O}^*$ of the covariates in step (1), generate synthetic data from the C-vine at all candidate truncation levels, and for each truncation level assess their privacy and utility.

(3) Find the truncation that offers optimal privacy-utility balance to the user by consulting the privacy-utility plot.

Steps (1) to (3) are executed by the data holder and are not made public; only the resulting synthetic data fulfilling the data holder's privacy and utility demands is published.

In Algorithm 1 in step (1), user knowledge about sensitive covariates is considered together with the empirical dependence properties of the real data in order to find an order $\mathcal{O}^*$ of the d covariates in which they will enter the C-vine copula model, such that truncation of the C-vine cuts off privacy leaking dependencies, while maintaining high utility. A theoretical justification of Algorithm 1 is given in Section 2.6 and further details on the algorithm are provided in Appendix C. The order $\mathcal{O}^*$, together with the vine tree structure $\mathcal{V}$ of the star-shaped C-vine, then determines the cascade of pair copulas of conditional distributions across the hierarchy of vine trees, see Proposition C.1. In each order $\mathcal{O}^*$ we require the response Y to be the center of the first tree in the C-vine. More specifically, let $X_{(1)}, \dots, X_{(d)}$ be the covariates in the chosen order. Then T_1 of the C-vine models the pairwise dependence between Y and each of $X_{(1)}, \dots, X_{(d)}$, T_2 the pairwise dependence between $X_{(d)}$ and each of $X_{(1)}, \dots, X_{(d-1)}$, conditioning on Y, T_3 the pairwise dependence between $X_{(d-1)}$ and each of $X_{(1)}, \dots, X_{(d-2)}$, conditioning on Y and $X_{(d)}$, and so on until the last tree T_d, which captures the pairwise dependence between $X_{(2)}$ and $X_{(1)}$, conditioning on $Y, X_{(d)}, \dots, X_{(3)}$. In Appendix C, we explore how orders other than $\mathcal{O}^*$ affect privacy.

For a selected ordering $\mathcal{O}^*$ of the covariates, the C-vine is estimated at user-defined maximal truncation level $t_{max} \leq d$. We advise to set $t_{max} := d + 1 - j$ or lower, where j is the position of the first sensitive feature to appear in the center node of a tree of the C-vine according to $\mathcal{O}^*$, as tree levels that model pairwise (conditional) dependencies with a sensitive feature and all other features should not be considered, see Section 2.6 for further theoretical grounding, and for large d (e.g. $d = 500$) we recommend to set $t_{max} << d$ because of uncertainty in the parameter estimation. Then for user-defined candidate truncation levels $t \in T \subset [t_{max}]$[4] the C-vine truncated at level $t \in T \setminus \{t_{max}\}$ is obtained by setting pair copulas of tree levels $t + 1$ and above to independence, i.e. removing tree after tree from the model. This means that for obtaining the C-vines of all candidate truncation levels $t \in T$ the sensitive real data only needs to be accessed once, namely for estimating the un-truncated C-vine. More precisely, let $(X, \boldsymbol{y}) \in \mathbb{R}^{n \times (d+1)}$ denote the real data where $X := (x_{ij}) \in \mathbb{R}^{n \times d}$, $i \in [n]$, $j \in [d]$, is the matrix of n realizations of the random vector $(X_1, \dots, X_d)$ and $\boldsymbol{y} := (y_1, \dots, y_n)^T$ is the vector of n realizations of the random variable Y. Then the vine copula model g with truncation level t is fit to the real data resulting in $\hat{g} := g\big((X, \boldsymbol{y}); \mathcal{V}, t\big)$ and the synthetic data $(Z, \boldsymbol{w}) \in \mathbb{R}^{n \times (d+1)}$ are sampled from $\hat{g}$. We use the estimation and sampling algorithms introduced in Dissmann et al. (2013) and implemented by Nagler and Vatter (2023). The vine copula model is estimated in an iterative, hierarchical fashion: Proceeding tree by tree, a greedy maximum spanning tree search with pairwise association measure as edge weights is conducted and parametric pair copulas corresponding to the edges are estimated with MLE and selected with AIC (Akaike, 1998).

After synthetic data have been generated from the C-vine truncated at each $t \in T$, their privacy P_t and utility U_t are assessed in step (2) using the methods explained hereafter. This results in points (U_t, P_t) for truncation levels $t \in T$ in the privacy-utility plot of step (3). Here, U_t is a measure of prediction performance over several synthetic data sets generated from the C-vine with specific ordering and truncation level t (see below for details), whereas P_t is either the median

[3] The implementation of Algorithm 1 as well as experiments can be found at: `https://github.com/ElisabethGriesbauer/T-Vine-Synth`.

[4] This can for example be every 5th truncation level, i.e. $T := \{1, 5, 10, 15, 20, 26\}$ for $d = 26$.

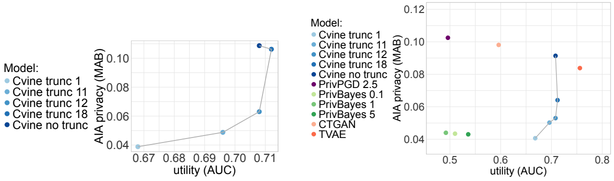
Algorithm 1 Finding Order $\mathcal{O}^*$

```
**Input:** $(X, \boldsymbol{y})$, initial order $\mathcal{O}^0 = (X_1, ..., X_d, Y)$ with $X_{d+1} := Y$, pairwise association measure $\rho : \mathbb{R}^{n\times 2} \to \mathbb{R}$,[5] pairwise association threshold $\rho^* > 0$, sensitive covariates $X_{j^*}$ with $j^* \in S \subset [d]$
**Output:** order $\mathcal{O}^*$
set $\mathcal{O}^*_{d+1} := Y$
compute $\rho_{j,k} := \rho(\boldsymbol{x}_j, \boldsymbol{x}_k)$ for $j \in [d]$ , $k > j$
set $K := \{k \in [d] : \ |\rho_{j^*,k}| > \rho^*$ for $j^* \in S\}$, the set of variables highly associated with sensitive features
for $j \in \{1, ..., |S|\}$ do
    set $\mathcal{O}^*_j := X_{j^*}$ with $j^* \in S$
end for
order $|\rho_{k,j^*}|$ for $k \in K$ and $j^* \in S$ in descending order $|\rho_{(1)}|, ..., |\rho_{(|K|)}|$
for $j \in \{1, ..., |K|\}$ do
    set $\mathcal{O}^*_{|S|+j} := X_k$ if $|\rho_{(j)}| = |\rho_{k,j^*}|$ with $k \in K$
end for
$r := 1$
for $j \in [d] \setminus (S \cup K)$ do
    $\mathcal{O}^*_{|S|+|K|+r} := X_j$
    $r := r + 1$
end for
```

MAB of an AIA or the median PG of a MIA over several synthetic data sets generated from the model and several runs of the privacy attack, see definitions below. Due to Theorem 2.3, it is necessary to evaluate all truncation levels t in the candidate set T. Figure 1 shows a plot of (U_t, P_t), where higher values along each axis indicate a better privacy or utility. The privacy-utility plot allows to observe a trajectory of how the privacy-utility trade-off of a C-vine with a specific ordering develops with its truncation level. Adding the results of competitor models, the privacy-utility plot allows to take a well-informed decision on the TVineSynth model, offering the desired privacy-utility balance. In Appendix D we further elaborate on how finding the best truncation level according to user demands can be formalized as an optimization problem. Considerations on the computational complexity of TVineSynth can be found in Appendix B.

2.3 Competitor Models

TVineSynth is benchmarked against PrivBayes (Zhang et al., 2017) and PrivPGD (Donhauser et al., 2024), which offer DP guarantees, and CTGAN and TVAE (Xu et al., 2019), which do not provide any DP guarantees, but are designed to resemble the real data as closely as possible. For details on the competitor models and their choice see Appendix F.



(a) Sensitive feature X_6. (b) Sensitive feature X_1.

Figure 1: Privacy-utility plot of synthetic data generated with a C-vine truncated at $t \in \{1, 11, 12, 18\}$ and no truncation (a) and competitors (b) from simulated real data. For AIA privacy, the MAB and for utility the median over 50 synthetic data sets are reported.[6] Parameters of the generative models and privacy attacks can be found in Appendix J.

2.4 Utility

We generate synthetic data to substitute private real data in a general regression task with response variable Y. This includes classification when Y is binary. Fitting a vine copula is more challenging on discrete than on continuous data[7]. Therefore we focus on a binary classification task. For assessing the utility of the synthetic data we compare Train on Synthetic - Test on Real (TSTR) to Train on Real - Test on Real (TRTR). Let $(X^*, \boldsymbol{y}^*)$ be a hold-out, real test data set of size n_{test} that was not used to learn the generative models. Let $f : \mathbb{R}^d \to \{0, 1\}$ be a classifier and $\hat{f}$ be its estimate from the real data $(X, \boldsymbol{y})$. Let $\hat{\boldsymbol{y}}^*$ be the prediction of the classifier f estimated from $(X, \boldsymbol{y})$ applied to the test data $(X^*, \boldsymbol{y}^*)$ and let $\hat{\boldsymbol{w}}^*$ be the prediction of the classifier f estimated from the synthetic data $(Z, \boldsymbol{w})$ applied to $(X^*, \boldsymbol{y}^*)$. The utility of the synthetic data is assessed by comparing $\hat{\boldsymbol{w}}^*$ to $\hat{\boldsymbol{y}}^*$ through comparing $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ and $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{y}}^*)$, the area under the receiver operating characteristic curve (AUC). This allows us to analyse how the performance of the classifier on real test data changes when it is trained on synthetic instead of real data.

2.5 Privacy

The privacy of the synthetic data is assessed through a membership and an attribute inference attack (MIA and AIA) (Shokri et al., 2017; Yeom et al., 2018). We follow the framework of Stadler et al. (2022), who model these attacks as privacy games between an attacker

[5] We recommend choosing a pairwise association measure that is scale invariant, such as Kendall's τ.

[6] Boxes and whiskers are not displayed in the privacy-utility plots, as they can already be found in Figures 10 (AIA privacy) and 14 (utility) in Appendix L.3, and to simplify visual inspection of the figure.

[7] The copula is uniquely defined only on the Cartesian product of the ranges of the marginal distributions for a discrete Y, (Panagiotelis et al., 2012).

and a challenger, the data holder.

Membership Inference Attack (MIA) In a MIA, the attacker aims to infer from $(Z, \boldsymbol{w})$ whether a target observation $(\boldsymbol{x}_t^T, y_t)$ is part $(X, \boldsymbol{y})$. The attacker has access to a reference data set $(X, \boldsymbol{y})_{ref}$ coming from the same distribution as the real data, and knows the size of the real and synthetic data (both n) and which generative model class is used. Then the attacker repeatedly samples data sets of fixed size from the reference data, adds the target observation half of the time and trains the generative model on them. After that, the attacker samples several synthetic data sets from each trained model and labels them according to whether the target observation has been added to the training data or not. A classifier is trained on the labeled synthetic data sets to estimate whether the target observation was part of the real data. The MIA game is repeated $N \in \mathbb{N}$ times.

Attribute Inference Attack (AIA) In an AIA, the attacker aims to infer the sensitive feature value x_{t,j^*} of a target observation $(\boldsymbol{x}_t^T, y_t)$ from $(Z, \boldsymbol{w})$ for some sensitive feature X_{j^*}, $j^* \in S \subset [d]$. The attacker has access to $(X, \boldsymbol{y})_{ref}$, a reference data set of fixed size coming from the same distribution as the real data, and knows which generative model class is used. Then the attacker trains the generative model on the reference data and samples $n_{synth} \in \mathbb{N}$ synthetic data sets from the estimated model. Subsequently, the attacker standardizes[8] the synthetic data to obtain $(\tilde{Z}, \tilde{\boldsymbol{w}})$, fits a linear regression model on the non-sensitive features of $(\tilde{Z}, \tilde{\boldsymbol{w}})$ with $\tilde{\boldsymbol{z}}_{j^*}$ as response and issues a guess $\hat{\tilde{x}}_{t,j^*}$ based on real $(\tilde{\boldsymbol{x}}_{t,-j^*}^T, y_t)$ that was standardized by the data holder. The AIA game is repeated $N \in \mathbb{N}$ times.

Choice of Sensitive Features The definition of sensitive features is based on domain knowledge and legal considerations such as GDPR (GDPR, 2016). Sensitive features involve personal information about health, demography, financial situation, behaviors, etc. that, if available to adversaries can be used to cause harm to data subjects or related people (Ohm, 2014). For the case that domain knowledge is lacking, Yoon et al. (2020) propose a definition of sensitive features: They consider features as sensitive, if they allow identification of an individual with high probability, for example because the feature values are extreme or rare.

Measures of Privacy As a measure of privacy protection against MIAs we use the *privacy gain (PG)* w.r.t. a given target observation, as proposed in Stadler et al. (2022). The PG is defined as the 'reduction in the attacker's advantage when given access to the synthetic data instead of the real data', where $PG \in [0, 2]$ and $PG = 1$ indicates best possible privacy.

The definition of the PG for AIAs provided by Stadler et al. (2022) does not make sense for continuous sensitive features, see Appendix G. Olatunji et al. (2023) propose to use the MSE in this case, which measures distance between the attacker's guess and the actual sensitive feature value. However, the MSE may be low just because the actual sensitive feature value is close to the sensitive feature's mean and not because the non-sensitive features inform the sensitive feature in the synthetic data, see Appendix G for an example. The influence of a covariate in a regression model (non-sensitive feature) on the dependent variable (sensitive feature) can be assessed by the magnitude of its regression coefficient. The *mean absolute β-coefficient (MAB)* summarizes how much the non-sensitive features inform the sensitive feature when the target observation was part of the generative model training in one number and naturally builds on how AIAs are commonly implemented, such as by Stadler et al. (2022).

Definition 2.2 (Mean Absolute β-Coefficient, MAB)**.** *Let an attacker perform an AIA according to Stadler et al. (2022). Then in a given run m of the game, with X_{j^*} as the sensitive feature, a linear Gaussian regression is fitted by ordinary least squares to each of the l standardized synthetic data sets $\boldsymbol{V}_l = (\tilde{Z}_l, \tilde{\boldsymbol{w}}_l)$. This results in the coefficients $\hat{\boldsymbol{\beta}}^{(j^*)}_{m,l} = (\hat{\beta}^{(j^*)}_{1,m,l}, \dots, \hat{\beta}^{(j^*)}_{d,m,l})^T$, with $m \in [N]$ and $l \in [n_{synth}]$. The MAB_{j^*} for sensitive covariate X_{j^*}, $j^* \in S \subset [d]$ is defined as:*

$$MAB_{j^*} := \frac{1}{dNn_{synth}} \sum_{k\in[d]} \sum_{m\in[N]} \sum_{l\in[n_{synth}]} |\hat{\beta}^{(j^*)}_{k,m,l}| \, . \tag{1}$$

The intercept $\hat{\beta}^{(j^*)}_{0,m,l}$ is not included in the definition of the MAB. The MAB will only be low if the AIA is unsuccessful. In Appendix G we present an extension of the MAB to measure worst-case AIA privacy.

Choice of Target Observations The PG is defined w.r.t. a *single target observation* of the real data. Thus, the results of an MIA do not only depend on the synthetic data, but also on the choice of target observation. To provide a realistic privacy evaluation we follow Stadler et al. (2022) and pick two sets of target observations: outlying targets outside the 95% quantile and randomly sampled targets. Although a set of target observations is needed to conduct an AIA, the MAB is independent of the choice of target observation, see Definition 2.2.

[8] By standardizing $\boldsymbol{x} \in \mathbb{R}^n$ to obtain $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \dots, \tilde{x}_n)^T$ we refer to $\tilde{x}_i := \frac{x_i - \bar{x}}{\left(\frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})^2\right)^{\frac{1}{2}}}$ with $\bar{x} := \frac{1}{n}\sum_{i=1}^n x_i$.

2.6 Theoretical Justification of TVineSynth

In the theorems below, we provide a theoretical justification for the TVineSynth construction. Let $(\boldsymbol{x}_i^T, y_i)$, $i \in [n]$ be i.i.d. samples of $(\boldsymbol{X}^T, Y)$, that follow a C-vine distribution with parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)$, where $\boldsymbol{\theta}_t$ are the parameters of tree number t[9], $\boldsymbol{X}$ is arranged according to the order of the C-vine and Y is binary with $P(Y = 1) = \pi_Y$ and $X_j \sim U(0, 1)$, $j \in [d]$. Further, let ψ be the log-odds ratio for a given observation $\boldsymbol{x}$ of $\boldsymbol{X}$, i.e.

$$\begin{aligned} \psi &= \psi(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d; \boldsymbol{x}) && (2) \\ &= \log \frac{P(Y = 1|\boldsymbol{X} = \boldsymbol{x}; \pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)}{P(Y = 0|\boldsymbol{X} = \boldsymbol{x}; \pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)} \ . && (3) \end{aligned}$$

It is easily shown that ψ is of the form:

$$\psi = \sum_{t=1}^{d} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t; \boldsymbol{x}) \ , \tag{4}$$

where the ψ_t's are given by:

$$\psi_1 = \log \frac{\pi_Y}{1 - \pi_Y} + \sum_{j=1}^{d} \log \frac{f_{j|y}(x_j|1)}{f_{j|y}(x_j|0)} \ , \tag{5}$$

and

$$\psi_t = \sum_{j=1}^{d+1-t} \log \frac{c^1_{j,d+2-t;d+3-t,\ldots,d,y}}{c^0_{j,d+2-t;d+3-t,\ldots,d,y}} \ , \tag{6}$$

with $t \in \{2, \ldots, d\}$ where $c^k_{j,d+2-t;d+3-t\ldots d,y}$ is evaluated at $(\boldsymbol{x}, y) = (\boldsymbol{x}, k)$. Moreover, let:

$$\hat{\psi} = \psi(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_d; \boldsymbol{x}) \ , \tag{7}$$

where $(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_d)$ are the maximum likelihood estimators of $(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)$, and

$$\tilde{\psi}^\tau = \sum_{t=1}^{\tau} \psi_t(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_t; \boldsymbol{x}) \tag{8}$$

be the estimator of ψ from the C-vine truncated at level τ.

Theorem 2.3. *Under these assumptions, it holds for large enough n that:*

$$MSE(\hat{\psi}) = E\Big[(\hat{\psi} - \psi)^2\Big] = \frac{1}{n} \cdot \boldsymbol{v}^T \boldsymbol{J}^{-1} \boldsymbol{v} + \mathcal{O}\left(\frac{1}{n}\right) \ ,$$

$$MSE(\tilde{\psi}^\tau) = \left(\sum_{t=\tau+1}^{d} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t; \boldsymbol{x}) \right)^2 + \frac{1}{n} \cdot \left(\boldsymbol{v}^{1\ldots\tau}\right)^T \boldsymbol{J}^{1\ldots\tau,1\ldots\tau} \boldsymbol{v}^{1\ldots\tau} + \mathcal{O}\left(\frac{1}{n}\right) \ ,$$

with:

$$\boldsymbol{v} = \frac{\partial \psi}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)} \ , \tag{9}$$

$$\boldsymbol{v}^{1\ldots\tau} = \frac{\partial \sum_{t=1}^{\tau} \psi_t}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau)} \ , \tag{10}$$

$$\begin{aligned} \boldsymbol{J} = -E\Big[&\frac{\partial^2}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d) \partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)^T} && (11) \\ &\log f(\boldsymbol{X}, Y)\Big] && (12) \end{aligned}$$

and $\boldsymbol{J}^{1\ldots\tau,1\ldots\tau}$ is the upper left sub-matrix of $\boldsymbol{J}^{-1}$ corresponding to the parameters $(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau)$.

This means that as the size n of the training data increases, the MSE of the estimated log-odds ratio for the full vine vanishes, while the one for the τ-truncated vine is dominated by the squared bias. Hence, for large n the utility of the truncated vine is lower than that of the full one. However, the bias does not necessarily increase monotonically as τ decreases, i.e. as more trees are truncated away. This is due to the fact that it consists of sums of log-differences of pair copula densities, the sign of which will vary with the copula families, parameters and $\boldsymbol{x}$. Further, the variance term of the MSE will typically be smaller for the truncated vine, meaning that for smaller n the utility of the full vine is not necessarily higher than that of a τ-truncated one, as seen e.g. in Figure 1. This is one of the reasons why we recommend to go through all, or at least several, truncation levels in order to find the best one.

Assume now that the order of the columns of $\boldsymbol{V}$, defined in Definition 2.2, is the same as order of the variables in the C-vine, where we omit the subscripts m and l for simplicity. Also, note that a multivariate normal distribution may be expressed as a C-vine with only Gaussian pair copulas, combined with normal margins.

Theorem 2.4. *Assume that each row of $\boldsymbol{V}$ follows a standard $(d + 1)$-variate normal distribution with correlation matrix $\boldsymbol{\rho}$. Then $\hat{\boldsymbol{\beta}}$ follows a d-variate normal distribution with:*

$$E[\hat{\boldsymbol{\beta}}^{(j^*)}] = \boldsymbol{\beta}^{(j^*)} = \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{\rho}_{[d+1]\setminus\{j^*\},j^*} \tag{13}$$

and covariance matrix:

$$Var(\hat{\boldsymbol{\beta}}^{(j^*)}) = (\sigma^{(j^*)})^2 (\boldsymbol{V}^T_{[d+1]\setminus\{j^*\}} \boldsymbol{V}_{[d+1]\setminus\{j^*\}})^{-1} \tag{14}$$

with:

$$\begin{aligned} (\sigma^{(j^*)})^2 = 1 - \boldsymbol{\rho}^T_{[d+1]\setminus\{j^*\},j^*} &\boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} && (15) \\ &\boldsymbol{\rho}_{[d+1]\setminus\{j^*\},j^*} \ . && (16) \end{aligned}$$

[9] The index t, that in prior sections was used to denote the truncation level, is in this section used as a running index for trees; the truncation level will instead be denoted by τ.

Let now the C-vine of $\boldsymbol{V}$ *be truncated at level* $\tau \leq d+1-j^*$ *and let* $\boldsymbol{\beta}^{(j^*)}_{(\tau)}$ *be the coefficient corresponding to the C-vine truncated at level* τ*. Then:*

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\;1\ldots d-\tau} = \mathbf{0}\;, \tag{17}$$

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\;d+1-\tau\ldots d} = \boldsymbol{\rho}^{-1}_{d+2-\tau\ldots d+1,d+2-\tau\ldots d+1}\boldsymbol{\rho}_{d+2-\tau\ldots d+1,j^*} \tag{18}$$

and:

$$(\sigma^{(j^*)}_{(\tau)})^2 = 1 - \boldsymbol{\rho}^{T}_{d+2-\tau\ldots d+1,j^*}\boldsymbol{\rho}^{-1}_{d+2-\tau\ldots d+1,d+2-\tau\ldots d+1} \tag{19}$$

$$\boldsymbol{\rho}_{d+2-\tau\ldots d+1,j^*}\;\;. \tag{20}$$

Theorem 2.5. *Under the same assumptions as Theorem 2.4, if* $\boldsymbol{\rho}$ *has a block structure with* $\rho_{kl} = 0$, $\forall(k,l)$ *with* $k \in (K \cup S)$ *and* $l \in [d+1] \setminus (K \cup S)$*, where* K *and* S *are as defined in Algorithm 1, and the C-vine of* $\boldsymbol{V}$ *is truncated at level* $\tau \leq d+1-|K|-|S|$*, then* $\boldsymbol{\beta}^{(j^*)}_{(\tau)} = \mathbf{0}$.

Proofs of Theorems 2.3, 2.4 and 2.5 are given in Appendix H.

This means that if the C-vine is truncated somewhere below the tree where the sensitive feature appears in the center node, some of the $\hat{\beta}_k$s will have mean 0, and will thus tend to be small, which reduces the attacker's ability to guess the value of the sensitive variable, and improves the protection of privacy. Further, the number of $\hat{\beta}_k$s with mean 0 increases by 1 for each tree that is truncated away. If in addition the correlation matrix of the C-vine follows a block structure, where the block containing the sensitive features is approximately uncorrelated with the remaining block(s), then all $\hat{\beta}_k$s will have mean (approximately) 0 already at truncation level $\tau = d+1-|K|-|S|$, where the first variable not in the sensitive block appears in the center of the tree. This gives a high protection of privacy, without truncating away too many trees, thus increasing the potential for high utility. This is exactly the purpose of ordering the C-vine according to Algorithm 1. Without the block structure, one might have to truncate away all trees to obtain the same protection of privacy, which would correspond to removing all dependencies between the variables, and a correspondingly minuscule utility. Note that the choice of ρ^* in Algorithm 1 affects the block structure of the correlation matrix $\boldsymbol{\rho}$ and thus the βs. A larger ρ^* leads to a smaller K, and potentially more correlated blocks, which reduces the protection of privacy.

3 RESULTS

Simulated Data We simulate a real data set to study the effect of truncation and $\mathcal{V}$ on privacy and utility, see Appendix L for details. AIA results of the C-vine confirm Theorem 2.5 as the MAB jumps at the truncation level expected from the block structure of the real data's correlation matrix. Truncated at the level corresponding to the position of the sensitive covariate in $\mathcal{O}^*$, TVineSynth offers AIA and MIA privacy as good as PrivBayes and superior to CTGAN, TVAE and PrivPGD and a utility superior to CTGAN and especially to PrivBayes and PrivPGD, and comparable to TVAE, see Figure 1b. For more detailed results, please consult Appendix L.3.

3.1 Real-world Data

We apply TVineSynth to the real-world SUPPORT2 data containing patients suffering from various conditions (Harrell, 2022b). The binary response Y indicates if a patient died during the study. Covariates *crea* and *totcst* are selected as sensitive features, see Appendix M for details.

Privacy: Attribute Inference Attack In accordance with the block correlation matrix of the real data, see Figure 23 in Appendix M, after applying Algorithm 1 and Theorem 2.5, TVineSynth provides high AIA privacy when the C-vine is truncated below level 15, outperforming TVAE and PrivPGD and comparable to CTGAN. For truncation at level 10 and lower (*totcst*) and at level 1 (*crea*) the C-vine's MAB is as low as for PrivBayes, see Figure 2a. Moving from truncation at level 20 to no truncation the MAB_{totcst} of the C-vine changes its trend and decreases. This is because the un-truncated C-vine starts to model noise in the real data. Figure 2c confirms this, showing a decrease in utility for the un-truncated C-vine. Comparing to the generative models' utility, Figure 2c, we observe that the privacy protection offered by PrivBayes and CTGAN comes at the cost of utility. The PrivPGD exhibits a surprisingly high MAB. For this reason we additionally consulted the AIA's estimated β-coefficients. These exhibit a mean close to 0, indicating moderate privacy protection, but a high variation, which explains the PrivPGD's high MAB. This is confirmed by the MSE which for outlying targets is moderate to high, see Appendix M.4. Hence the PrivPGD seems quite unstable compared to the other synthetic data generators between different runs of the AIA.

Privacy: Membership Inference Attack The PG of C-vine generated synthetic data is around 1 with low variation for all truncation levels, indicating optimal MIA privacy for outlying (orange) and randomly sampled (blue) targets, Figure 2b. The C-vine's PG is seemingly independent of truncation level because the estimation of the un-truncated C-vine with Maximum Likelihood (ML) is robust w.r.t. adding/removing a single observation to the real data.[10] As a consequence, also a C-vine truncated at level $t < d$ shows the same robustness as the un-truncated C-vine. These results compare to PrivBayes for $\epsilon \in \{0.1, 1, 5\}$ and PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$. The PG of CTGAN is about 1 at median, but exhibits a high variation over different observations and repetitions of the MIA. The TVAE provides very low MIA privacy with a PG of around 0. This indicates that the TVAE generated synthetic data reproduce the SUPPORT2 data too detailed, harming privacy.

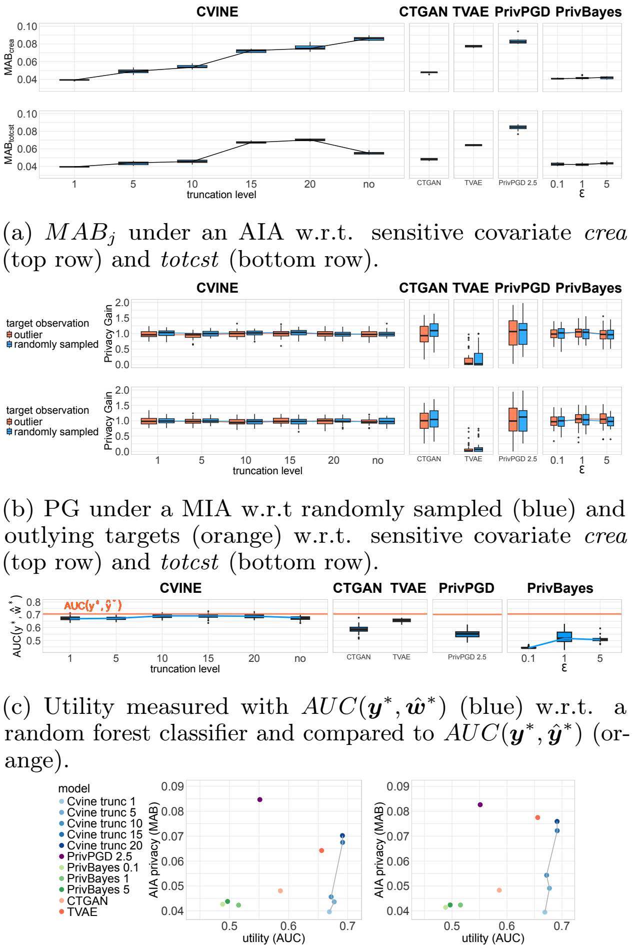
Utility For evaluating utility, 50 synthetic data sets of the same size as the real data ($n = 884$) are generated from each model, a random forest classifier is trained on each of them and tested on hold-out test data ($n_{test} = 220$). The C-vine generated synthetic data consistently outperform synthetic data generated from a CTGAN and PrivPGD and by far PrivBayes for all truncation levels, yielding an $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ almost as high as $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{y}}^*) \approx 0.71$, Figure 2c. Only the TVAE performs comparable to the C-vine. Considering its low PG and high MAB in Figures 2a and 2b, the TVAE violates the privacy by modeling the real data too closely. The C-vine, contrarily, captures the dependencies in the real data without compromising privacy.

Privacy-Utility Plots If we truncate at level 10 or lower for sensitive covariate *totcst* and at level 5 or lower for *crea*, TVineSynth generated synthetic data offer a privacy-utility balance superior to that of the competitors, Figure 2d. See Appendix M.6 for further results.

Statistical Fidelity In terms of the statistical fidelity and discrepancy between real and synthetic joint and marginal distributions TVineSynth outperforms its competitors, see Appendix M.7.

4 CONCLUSION

We present TVineSynth, a synthetic tabular data generator based on a truncated C-vine to balance privacy and utility and theoretically justify its construction. Experiments show that TVineSynth offers a privacy-utility trade-off superior to that of competitors. While TVineSynth is not limited to supervised ML tasks, the vine structure might have to be changed for applications such as clustering. Further work could focus on improving scalability, as inference on a vine copula is computationally difficult for more than 500 dimensions, and evaluating the synthetic data also w.r.t. fairness and explainability of predictions.



(a) MAB_j under an AIA w.r.t. sensitive covariate *crea* (top row) and *totcst* (bottom row).

(b) PG under a MIA w.r.t randomly sampled (blue) and outlying targets (orange) w.r.t. sensitive covariate *crea* (top row) and *totcst* (bottom row).

(c) Utility measured with $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ (blue) w.r.t. a random forest classifier and compared to $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{y}}^*)$ (orange).

(d) Privacy-utility plot w.r.t. AIA and sensitive features *totcst* (left) and *crea* (right).

Figure 2: MAB (a), PG (b), utility (c) and privacy-utility plots (d) of synthetic data generated with a C-vine for different truncation levels, CTGAN, TVAE, PrivPGD ($\epsilon = 2.5$, $\delta = 10^{-5}$) and PrivBayes ($\epsilon \in \{0.1, 1, 5\}$). Boxplots are obtained from 10 game iterations in the AIA and MIA, 50 synthetic data sets in the utility evaluation. Model and privacy attack parameters can be found in Appendix J.

Acknowledgments

This work was supported by the Research Council of Norway, Integreat - Norwegian Centre for knowledge-driven machine learning, project number 332645. The work of C. Czado is supported in part by the Deutsche Forschungsgemeinschaft (DFG CZ-86/6-1 CZ-86/10-1).

[10] The robustness of ML estimation depends on the sample size of the (real) data. Thus, for lower sample sizes than the ones used here, we would expect to see a MIA PG that varies more with truncation level.

References

Tachycardia. American College of Cardiology, `https://www.acc.org`. Accessed: 2024-11-11.

Tachypnea. American Thoracic Society, `https://www.thoracic.org`. Accessed: 2024-11-11.

Blood urea nitrogen (bun). `https://www.mayocliniclabs.com/test-catalog/overview/81793#Overview`. Accessed: 2024-11-08.

Harmonisation of reference intervals, pathology harmony. `https://www.acb.org.uk/static/eea82309-7e93-417d-9b8991b0c65c52fb/Pathology-Harmony-biochemistry.pdf`. Accessed: 2024-11-11.

Synthetic data privacy evaluation framework. `https://github.com/spring-epfl/synthetic_data_release`. Accessed: 2023-02-10.

Ctgan: Conditional gan for generating synthetic tabular data. `https://github.com/sdv-dev/CTGAN`. Accessed: 2023-12-06.

Saps ii. `https://clincalc.com/IcuMortality/SAPSII.aspx`. Accessed: 2024-11-06.

Synthetic data vault. `https://github.com/sdv-dev/SDV`. Accessed: 2023-12-06.

Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998.

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022.

Alexandra-Georgiana Andrei, Ahmedkhan Radzhabov, Ioan Coman, Vassili Kovalev, Bogdan Ionescu, and Henning Müller. Overview of imageclefmedical gans 2023 task—identifying training data "fingerprints" in synthetic biomedical images generated by gans for medical image security. In *CLEF2023 Working Notes, CEUR Workshop Proceedings*, 2023.

Kunihiro Baba, Ritei Shibata, and Masaaki Sibuya. Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, 46(4):657–664, 2004.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

Tim Bedford and Roger M Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32(1):245–268, 2001.

Tim Bedford and Roger M Cooke. Vines–a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.

Fodil Benali, Damien Bodénès, Nicolas Labroche, and Cyril de Runz. Mtcopula: Synthetic complex data generation using copula. In *23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*, pages 51–60, 2021.

Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

Victoria Cheng, Vinith M Suriyakumar, Natalie Dullerud, Shalmali Joshi, and Marzyeh Ghassemi. Can you fake it until you make it? impacts of differentially private synthetic data on downstream classification fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 149–160, 2021.

Amanda MY Chu, Chun Yin Ip, Benson SY Lam, and Mike KP So. Statistical disclosure control for continuous variables using an extended skew-t copula. *Applied Stochastic Models in Business and Industry*, 38(1):96–115, 2022a.

Amanda MY Chu, Chun Yin Ip, Benson SY Lam, and Mike KP So. Vine copula statistical disclosure control for mixed-type data. *Computational Statistics & Data Analysis*, 176:107561, 2022b.

Maximilian Coblenz, Oliver Grothe, and Fabian Kächele. Learning nonparametric high-dimensional generative models: The empirical-beta-copula autoencoder. *arXiv preprint arXiv:2309.09916*, 2023.

Claudia Czado. Analyzing dependent data with vine copulas. *Lecture Notes in Statistics, Springer*, 222, 2019.

Jeffrey Dissmann, Eike C Brechmann, Claudia Czado, and Dorota Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69, 2013.

Konstantin Donhauser, Javier Abad, Neha Hulkund, and Fanny Yang. Privacy-preserving data release leveraging optimal transport and particle gradient descent. *arXiv preprint arXiv:2401.17823*, 2024.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4): 211–407, 2014.

Hazel Finney, David J Newman, and Christopher P Price. Adult reference ranges for serum cystatin c, creatinine and predicted creatinine clearance. *Annals of clinical biochemistry*, 37(1):49–59, 2000.

Sébastien Gambs, Frédéric Ladouceur, Antoine Laurent, and Alexandre Roy-Gaumond. Growing synthetic data through differentially-private vine copulas. *Proc. Priv. Enhancing Technol.*, 2021(3):122–141, 2021.

Cong Gao, Benjamin D Killeen, Yicheng Hu, Robert B Grupp, Russell H Taylor, Mehran Armand, and Mathias Unberath. Synthetic data accelerates the development of generalizable learning-based algorithms for x-ray image analysis. *Nature Machine Intelligence*, pages 1–15, 2023.

General Data Protection Regulation GDPR. General data protection regulation. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC*, 2016.

Jens Goldschmidt, Elisabeth Moser, Leonard Nitzsche, Rudolf Bierl, and Jürgen Wöllenstein. Improving the performance of artificial neural networks trained on synthetic data in gas spectroscopy–a study on two sensing approaches: Approaches to overcome data scarcity when utilizing artificial neural networks in quantitative gas analysis. *tm-Technisches Messen*, (0), 2023.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Elisabeth Griesbauer. Vine copula based synthetic data generation for classification. 2022.

Frank Harrell. Support2 data set. `https://archive.ics.uci.edu/dataset/880/support2`, 2022a. UCI Machine Learning Repository. Accessed: 2024-01-07.

Frank Harrell. Support2 data set. `https://hbiostat.org/data/`, 2022b. Vanderbilt University Department of Biostatistics. Accessed: 2024-01-07.

Donna D Ignatavicius, M Linda Workman, and Cherie Rebar. *Medical-Surgical Nursing-E-Book: Concepts for Interprofessional Collaborative Care.* Elsevier Health Sciences, 2017.

Anubhav Jain, Nasir Memon, and Julian Togelius. Fair gans through model rebalancing with synthetic data. *arXiv preprint arXiv:2308.08638*, 2023.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

Harry Joe. Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, pages 120–141, 1996.

Harry Joe. *Multivariate models and multivariate dependence concepts.* CRC press, 1997.

Harry Joe. *Dependence modeling with copulas.* CRC press, 2014.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Sanket Kamthe, Samuel Assefa, and Marc Deisenroth. Copula flows for synthetic data generation. *arXiv preprint arXiv:2101.00598*, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

William A Knaus, Douglas P Wagner, Elizabeth A Draper, Jack E Zimmerman, Marilyn Bergner, Paulo G Bastos, Carl A Sirio, Donald J Murphy, Ted Lotring, Anne Damiano, et al. The apache iii prognostic system: risk prediction of hospital mortality for critically iii hospitalized adults. *Chest*, 100 (6):1619–1636, 1991.

William A Knaus, Frank E Harrell, Joanne Lynn, Lee Goldman, Russell S Phillips, Alfred F Connors, Neal V Dawson, William J Fulkerson, Robert M Califf, Norman Desbiens, et al. The support prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of internal medicine*, 122(3):191–203, 1995.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

Sandra Kumi, Maxwell Hilton, Charles Snow, Richard K Lomotey, and Ralph Deters. sleepsynth: Evaluating the use of synthetic data in health digital twins. In *2023 IEEE International Conference on Digital Health (ICDH)*, pages 121–130. IEEE, 2023.

Dorota Kurowicka and Roger Cooke. A parameterization of positive definite matrices in terms of partial correlation vines. *Linear Algebra and its Applications*, 372:225–251, 2003.

Jean-Roger Le Gall, Stanley Lemeshow, and Fabienne Saulnier. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Jama*, 270(24):2957–2963, 1993.

E.L. Lehmann. *Elements of Large-Sample Theory*. Springer, 1999.

Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Advances in database technology: proceedings. International conference on extending database technology*, volume 2014, page 475. NIH Public Access, 2014.

John William McEvoy, Cian P McCarthy, Rosa Maria Bruno, Sofie Brouwers, Michelle D Canavan, Claudio Ceconi, Ruxandra Maria Christodorescu, Stella S Daskalopoulou, Charles J Ferro, Eva Gerdts, et al. 2024 esc guidelines for the management of elevated blood pressure and hypertension: Developed by the task force on the management of elevated blood pressure and hypertension of the european society of cardiology (esc) and endorsed by the european society of endocrinology (ese) and the european stroke organisation (eso). *European heart journal*, 45(38): 3912–4018, 2024.

David Meyer and Thomas Nagler. Synthia: Multidimensional synthetic data generation in python. *Journal of Open Source Software*, 6(65):2863, 2021.

David Meyer, Thomas Nagler, and Robin J Hogan. Copula-based synthetic data generation for machine learning emulators in weather and climate: application to a simple radiation model. *Geosci. Model Dev. Discuss.(GMDD)*, pages 1–21, 2021.

Juan Morales-García, Andrés Bueno-Crespo, Fernando Terroso-Sáenz, Francisco Arcas-Túnez, Raquel Martínez-España, and José M Cecilia. Evaluation of synthetic data generation for intelligent climate control in greenhouses. *Applied Intelligence*, pages 1–17, 2023.

Thomas Nagler and Thibault Vatter. *rvinecopulib: High Performance Algorithms for Vine Copula Modeling*, 2023. R package version 0.6.3.1.1.

Roger B Nelsen. *An introduction to copulas*. Springer Science & Business Media, 2007.

Paul Ohm. Sensitive information. *S. Cal. L. Rev.*, 88: 1125, 2014.

Iyiola E Olatunji, Anmar Hizber, Oliver Sihlovec, and Megha Khosla. Does black-box attribute inference attacks on graph neural networks constitute privacy risk? *arXiv preprint arXiv:2306.00578*, 2023.

Anastasios Panagiotelis, Claudia Czado, and Harry Joe. Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499):1063–1072, 2012.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410. IEEE, 2016.

Vasileios C Pezoulas, Themis P Exarchos, Nikolaos S Tachos, Andreas Goules, Athanasios G Tzioufas, and Dimitrios I Fotiadis. Boosting the performance of malt lymphoma classification in patients with primary sjögren's syndrome through data augmentation: a case study. 2023.

Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities. *arXiv preprint arXiv:2301.07573*, 2023.

Lyrad K Riley and Jedda Rupert. Evaluation of patients with leukocytosis. *American family physician*, 92(11):1004–1011, 2015.

Osamu Saisho, Keiichiro Kashiwagi, Sakiko Kawai, Kazuki Iwahana, and Koki Mitani. Sandbox ai: We don't trust each other but want to create new value efficiently through collaboration using sensitive data. In *Adjunct Proceedings of the 2023 ACM International Joint Conference on Pervasive and Ubiquitous Computing & the 2023 ACM International Symposium on Wearable Computing*, pages 68–72, 2023.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Matthias Schaufelberger, Reinald Peter Kühle, Andreas Wachter, Frederic Weichel, Niclas Hagen, Friedemann Ringwald, Urs Eisenmann, Jürgen Hoffmann, Michael Engel, Christian Freudlsperger, et al. Impact of data synthesis strategies for the classification of craniosynostosis. *arXiv preprint arXiv:2310.10199*, 2023.

Shashank Shetty, VS Ananthanarayana, and Ajit Mahale. Data augmentation vs. synthetic data generation: An empirical evaluation for enhancing radiology image classification. In *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6. IEEE, 2023.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1451–1468, 2022.

Pierre Stock, Igor Shilov, Ilya Mironov, and Alexandre Sablayrolles. Defending against reconstruction attacks with r\'enyi differential privacy. *arXiv preprint arXiv:2202.07623*, 2022.

Jakob Stöber and Claudia Czado. Pair copula constructions. In *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*, pages 185–230. Singapore: World Scientific Publishing, 2nd edition, 2017.

Yi Sun, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Learning vine copula models for synthetic data generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5049–5057, 2019.

Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. Copulas as high-dimensional generative models: Vine copula autoencoders. *Advances in neural information processing systems*, 32, 2019.

Graham Teasdale and Bryan Jennett. Assessment of coma and impaired consciousness: a practical scale. *The Lancet*, 304(7872):81–84, 1974.

Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. *arXiv preprint arXiv:2302.12580*, 2023.

Alex X Wang, Stefanka S Chukova, Colin R Simpson, and Binh P Nguyen. Data-centric ai to improve early detection of mental illness. In *2023 IEEE Statistical Signal Processing Workshop (SSP)*, pages 369–373. IEEE, 2023a.

Jiyao Wang, Nicha C Dvornek, Lawrence H Staib, and James S Duncan. Learning sequential information in task-based fmri for synthetic data augmentation. *arXiv preprint arXiv:2308.15564*, 2023b.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, Nayeong Kim, Suha Kwak, and Tae-Hyun Oh. Exploiting synthetic data for data imbalance problems: Baselines from a data perspective. *arXiv preprint arXiv:2308.00994*, 2023.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.

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

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4): 1–41, 2017.

Alexander Ziller, Tamara T Mueller, Simon Stieger, Leonhard F Feiner, Johannes Brandt, Rickmer Braren, Daniel Rueckert, and Georgios Kaissis. Reconciling privacy and accuracy in ai for medical imaging. *Nature Machine Intelligence*, 6(7):764–774, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] See Section 2.2 and Appendix C.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] See Appendix B for the computational complexity of TVineSynth.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] See the submitted zip file containing the anonymized code.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] See Section 2.6 where we give the full set of assumptions made to theoretically justify TVineSynth.
   (b) Complete proofs of all theoretical results. [Yes] See Appendix H.
   (c) Clear explanations of any assumptions. [Yes] See Section 2.6.

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] See submitted code and Appendices J, L.2 and M.3.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] See Appendices J, L and M.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] See captions of figures for description of error bars.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] See Appendix K.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes] See Section 3.1 and Appendix M.

   (b) The license information of the assets, if applicable. [Yes] See submitted zip file.

   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] See Appendix L and submitted zip file.

   (d) Information about consent from data providers/curators. [Not Applicable] The data set used is published online, see Appendix M.

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] The data used is published and does not contain offensive content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable] No crowdsourcing or research with human subjects was conducted.

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] No crowdsourcing or research with human subjects was conducted.

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] No crowdsourcing or research with human subjects was conducted.

Appendix

Table of Contents

A An Introduction to Vine Copulas

This introduction to vine copulas is based on Griesbauer (2022) which again is based on Czado (2019). In the latter more details can be found. Vine copulas build on the concept of copulas.

A.1 Copulas

Definition A.1. *Let $d \in \mathbb{N}$. The function $C : [0,1]^d \to [0,1]^d$ is a d-dimensional copula if it is a d-dimensional cumulative distribution function with uniform marginal distributions $U[0,1]$.*

So for the random vector $(U_1, \ldots U_d)$ taking on values $(u_1, \ldots, u_d) \in [0,1]^d$ it is:

$$C(u_1, \ldots, u_d) = P(U_1 \leq u_1, \ldots, U_d \leq u_d)\ . \tag{21}$$

Theorem A.2 (Sklar's Theorem)**.** *Let $\boldsymbol{X}$ be a d-dimensional random vector with distribution function F and marginal distributions $F_1, \ldots F_d$. Then F can be expressed as:*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d))\ , \quad (x_1, \ldots, x_d) \in \mathbb{R}^d\ . \tag{22}$$

where C is a copula. If F is absolutely continuous, the copula C is unique. We then say that the copula C is corresponding to the distribution F. In the case of absolute continuity all densities exist and we can express the joint density f of $\boldsymbol{X}$ as:

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \cdot f_1(x_1) \cdot \ldots \cdot f_d(x_d)\ . \tag{23}$$

Conversely, let C be the d-dimensional copula corresponding to the joint distribution function F of $\boldsymbol{X}$ with marginal distributions $F_1, \ldots F_d$. Then we can express C as:

$$C(u_1, \ldots, u_d) = F(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)) \tag{24}$$

with copula density:

$$c(u_1, \ldots, u_d) = \frac{f(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d))}{f_1(F_1^{-1}(u_1)) \cdot \ldots \cdot f_d(F_d^{-1}(u_d))}\ . \tag{25}$$

Sklar's Theorem, Sklar (1959) provides the link between the copula on the d-dimensional hypercube and the probability distribution of the random vector $(X_1, \ldots, X_d)$. Equation (23) illustrates how the joint density f of a random vector $(X_1, \ldots, X_d)$ can be split into the joint copula density, which captures the dependence structure of $X_1, \ldots X_d$, and the marginal densities $f_1, \ldots f_d$.

The inverse Sklar's Theorem A.2 gives the construction of the *elliptical copulas*, to which the Gauss copula belongs.

Definition A.3 (bivariate Gauss copula)**.** *Let $\Phi_2(\cdot, \cdot; \rho)$ be the 2-dimensional standard normal distribution with mean vector $\boldsymbol{\mu} = 0$ and correlation parameter $\rho \in (0,1)$, and let $\Phi^{-1}(\cdot)$ be the quantile function of the univariate standard normal distribution. Then by Sklar's Theorem A.2 we obtain the bivariate Gauss copula by:*

$$C(u_1, u_2; \rho) = \Phi_2(\Phi^{-1}(u_1), \Phi^{-1}(u_2); \rho)\ . \tag{26}$$

The Clayton, Gumbel, Frank and Joe copulas belong to the class of *Archimedean copulas*, which is covered in more detail for example in Nelsen (2007).

Pair Copula Construction (PCC)

The set of multivariate copulas to choose from, i.e. elliptical and Archimedean copulas, is rather limited and constrained in modeling flexibility. However, complex high-dimensional dependence structures call for more flexible multivariate copulas. Aas et al. (2009), which the following subsection is based on, decompose a multivariate density by using a cascade of bivariate building blocks: pair copulas. Knowing how to decompose a multivariate distribution, this approach can be reversed in order to construct multivariate copulas and distribution functions respectively. These are flexible and their construction is simple. This is the idea of *pair copula construction.*

The following notation is defined:

Definition A.4. *Let $\boldsymbol{X}_D \in \mathbb{R}^d$ be a random vector and $\boldsymbol{x}_D \in \mathbb{R}^d$, let $i, j, d \in \mathbb{N}$ and $D \subset \mathbb{N}$ with $i, j \notin D$ and $|D| = d$. Let $F_{ij\,|\,D}(\cdot, \cdot \,|\, \boldsymbol{X}_D = \boldsymbol{x}_D)$ be the conditional distribution of (X_i, X_j) given that $\boldsymbol{X}_D = \boldsymbol{x}_D$. The copula distribution associated with $F_{ij\,|\,D}(\cdot, \cdot \,|\, \boldsymbol{X}_D = \boldsymbol{x}_D)$ is denoted by:*

$$C_{ij;D}(\cdot, \cdot; \boldsymbol{x}_D) \; .$$

If existing, its corresponding density is denoted by:

$$c_{ij;D}(\cdot, \cdot; \boldsymbol{x}_D) \; .$$

Let $\boldsymbol{X} = (X_1, X_2, X_3)$ be a random vector with joint density function f_{123} and marginal density functions f_1, f_2 and f_3. Using conditioning we can rewrite the joint density function:

$$f_{123}(x_1, x_2, x_3) = f_{1|23}(x_1 \,|\, x_2, x_3) f_{2|3}(x_2 \,|\, x_3) f_3(x_3) \; , \tag{27}$$

with:

$$f_{2|3}(x_2 \,|\, x_3) = \frac{f_{23}(x_2, x_3)}{f_3(x_3)} \; , \tag{28}$$

$$f_{1|23}(x_1 \,|\, x_2, x_3) = \frac{f_{123}(x_1, x_2, x_3)}{f_{23}(x_2, x_3)} = \frac{f_{13|2}(x_1, x_3 \,|\, x_2)}{f_{3|2}(x_3 \,|\, x_2)} \; . \tag{29}$$

By Sklar's Theorem A.2 we know, that:

$$f_{23}(x_2, x_3) = c_{23}(F_2(x_2), F_3(x_3)) f_2(x_2) f_3(x_3) \; , \tag{30}$$

and thus (28) becomes:

$$f_{2|3}(x_2 \,|\, x_3) := \frac{f_{23}(x_2, x_3)}{f_3(x_3)} = c_{23}(F_2(x_2), F_3(x_3)) f_2(x_2) \; . \tag{31}$$

In the same manner we obtain (29):

$$\begin{aligned} f_{1|23}(x_1 \,|\, x_2, x_3) &= \frac{f_{13|2}(x_1, x_3 \,|\, x_2)}{f_{3|2}(x_3 \,|\, x_2)} \\ &= \frac{c_{13;2}(F(x_1 \,|\, x_2), F(x_3 \,|\, x_2); x_2) f_{1|2}(x_1 \,|\, x_2) f_{3|2}(x_3 \,|\, x_2)}{f_{3|2}(x_3 \,|\, x_2)} \\ &= c_{13;2}(F(x_1 \,|\, x_2), F(x_3 \,|\, x_2); x_2) f_{1|2}(x_1 \,|\, x_2) \\ &= c_{13;2}(F(x_1 \,|\, x_2), F(x_3 \,|\, x_2); x_2) c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) \; . \end{aligned} \tag{32}$$

Combining (31) and (32) we can decompose (27) into a product of pair copulas and marginal distributions:

$$\begin{aligned} f_{123}(x_1, x_2, x_3) = & \, c_{13;2}(F(x_1 \,|\, x_2), F(x_3 \,|\, x_2); x_2) \\ & \, c_{12}(F_1(x_1), F_2(x_2)) \, c_{23}(F_2(x_2), F_3(x_3)) \\ & \, f_1(x_1) \, f_2(x_2) \, f_3(x_3) \; . \end{aligned} \tag{33}$$

Remark A.5.

- *The decomposition with conditioning in (27) is not unique. Neither is therefore (33). In general we could reorder (X_1, X_2, X_3) in $3! = 6$ ways. However, in a pair copula there is no distinction made between the first and the second argument, i.e. $c_{ij}(u_i, u_j) = c_{ji}(u_j, u_i)$. That is why we end up with three distinct decompositions in the form of (33).*

- *We see, that $c_{13;2}(\cdot, \cdot; x_2)$, the pair copula associated with the conditional distribution of (X_1, X_3) given $X_2 = x_2$ depends on the value x_2 of X_2. We stick to the terminology in Czado (2019) and speak of a pair copula decomposition, if the copulas associated with conditional distributions are allowed to depend on the value of the conditioning variable, i.e. here $X_2 = x_2$. If we ignore this dependence, which in our case would be equivalent to:*

  $$\forall x_2 \in \mathbb{R} : \quad c_{13;2}(u_1, u_3; x_2) = c_{13;2}(u_1, u_3), \quad u_1 \in [0,1], \; u_3 \in [0,1] \; ,$$

  we make the simplifying assumption in three dimensions. In general it assumes, that copulas associated with conditional distributions do not depend on the value(s) of the conditioning variable(s). If we assume the simplifying assumption, we can reverse the decomposition approach and view the simplified version of (33) as the construction of the three dimensional density f_{123} from pair copula densities, conditional distributions and marginal densities. In this case we speak of pair copula construction.

- *Obviously the construction 3-dimensional example above can be generalized to higher dimensions. There we encounter conditional marginal densities, which can be expressed as:*

  $$f(x \,|\, \boldsymbol{v}) = c_{xv_j;\boldsymbol{v}_{-j}}(F(x \,|\, \boldsymbol{v}_{-j}), F(v_j \mid \boldsymbol{v}_{-j})) \cdot f(x \,|\, \boldsymbol{v}_{-j}) \; , \tag{34}$$

  with $\boldsymbol{v} \in \mathbb{R}^d$ and $\boldsymbol{v}_{-j}$ the sub-vector of $\boldsymbol{v}$ with the jth component left out. The second factor of (34) can again be factorized with (34). This illustrates the iterative nature of the construction. Finally, with the result of Joe (1996), that:

  $$\begin{aligned} \forall j : \quad F(x \,|\, \boldsymbol{v}) &= \left. \frac{\partial C_{x,v_j;\boldsymbol{v}_{-j}}\big(F(x \,|\, \boldsymbol{v}_{-j}), u\big)}{\partial u} \right|_{u=F(v_j \mid \boldsymbol{v}_{-j})} \\ &=: \frac{\partial C_{x,v_j;\boldsymbol{v}_{-j}}\big(F(x \,|\, \boldsymbol{v}_{-j}), F(v_j \,|\, \boldsymbol{v}_{-j})\big)}{\partial F(v_j \,|\, \boldsymbol{v}_{-j})} \; , \end{aligned} \tag{35}$$

  the construction or decomposition respectively is completed. Here h-functions help to simplify the notation of conditional distributions and copulas.

  Definition A.6. *For a bivariate copula C_{uv} the corresponding h-function is defined for all $(u, v) \in [0,1]^2$ as:*

  $$h_{u \,|\, v}(u \,|\, v) := \frac{\partial}{\partial v} C_{uv}(u, v) \; . \tag{36}$$

  Clearly (35) holds for any continuous distribution F and thus also for the bivariate copula distribution C_{uv}. With $C(u) = u$ for any $u \in [0,1]$ and the copula C it follows that:

  $$C_{u \,|\, v}(u \,|\, v) \overset{(35)}{=} \frac{\partial}{\partial v} C_{uv}(u, v) \overset{A.6}{=} h_{u \,|\, v}(u \,|\, v) \; . \tag{37}$$

A.2 Regular Vines

In Section A.1 we saw that a d-dimensional probability distribution function can be constructed from or decomposed into bivariate building-blocks, pair copulas. For a specific d-dimensional probability distribution there exist several pair copula constructions, a subset of them satisfying the *proximity condition* introduced in the following.

Bedford and Cooke (2001) and Bedford and Cooke (2002) introduced *regular vines (R-vines)* and the *R-vine specification* to efficiently represent the pair copula constructions satisfying the proximity condition. The *R-vine specification* captures the structure of the pair copula construction: each bivariate copula is associated with an edge in a sequence of nested trees, the *R-vine*. The families, rotations and parameters of the bivariate copulas may be stored in matrices. This compact notation facilitates the estimation and sampling procedures on R-vines. Bedford and Cooke (2001) and Bedford and Cooke (2002) also show, that each R-vine specification stands for a unique d-dimensional distribution F.

Definition A.7 (Vine, regular vine, regular vine tree sequence)**.** *A set of trees* $\mathcal{V} = (T_1, ..., T_{d-1})$ *is a vine on* d *elements if:*

(i) T_1 *is a tree with edge set* E_1 *and node set* $V_1 = \{1, ..., d\}$*.*

(ii) For $i \in \{2, ..., (d-1)\}$ *it holds that* T_i *is a tree with edge set* E_i *and node set* $V_i = E_{i-1}$*.*

$\mathcal{V}$ *is an regular vine (R-vine) or regular vine tree sequence (R-vine tree sequence) if additionally the so called proximity condition holds:*

(iii) For $i \in \{2, ..., (d-1)\}$ *and* $\{a, b\} \in E_i$ *with* $a = \{a_1, a_2\}$ *and* $b = \{b_1, b_2\}$ *we have that* $|a \cap b| = 1$*.*

Remark A.8. *The proximity condition makes sure that nodes* a *and* b *are only then joined by an edge in tree* T_i *if they share a common node in tree* T_{i-1}*, where* $a, b \in E_{i-1}$*.*

Among the R-vines there are (among others) the two sub-classes of C-vines and D-vine. They distinguish themselves through a special structure each tree in $\mathcal{V}$ takes on.

Definition A.9 (C-vine, D-vine)**.** *An R-vine tree sequence* $\mathcal{V}$ *on* d *elements is called:*

(i) D-vine, if for each node v *of each tree* $T_i \in \mathcal{V},\ i \in [d-1]$ *it holds that* $deg(v) \leq 2$*,*

(ii) C-vine, if in each tree $T_i \in \mathcal{V},\ i \in [d-1]$ *there is one unique node* v *with* $deg(v) = d - i$ *which is called root node.*

Below is the tree sequence of a C-vine on 5 elements without node and edge labels. The root node in each star-shaped tree is coloured.

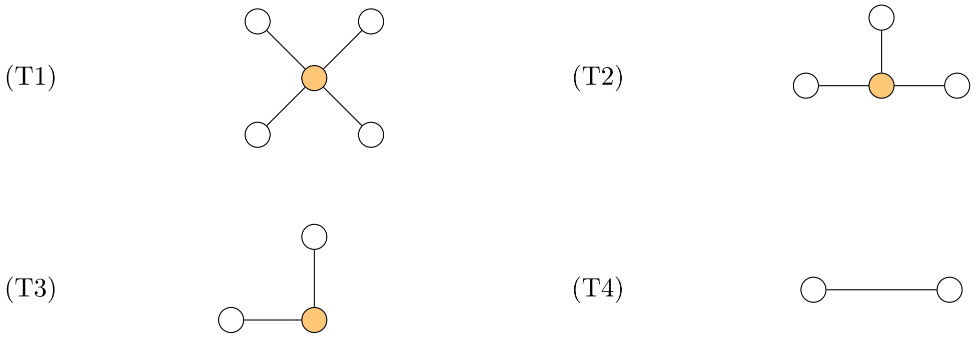


Figure 3: A C-vine on 5 elements.

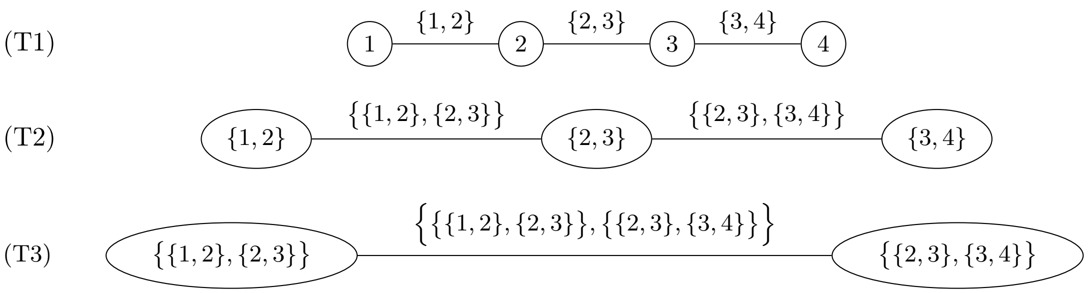


Figure 4: A D-vine on 4 elements.

In Figure 4 an R-vine tree sequence $\mathcal{V}$ on $d = 4$ elements is displayed. Note that $\mathcal{V}$ is a D-vine.

This notation for edges and nodes is hard to read and use. By Bedford and Cooke (2002) and results of Kurowicka and Cooke (2003) the edges of each tree can be uniquely identified by two *conditioned nodes* and a set of *conditioning nodes*.

Definition A.10 (Complete union, conditioning set, conditioned set)**.** *Let $\mathcal{V}$ be an R-vine tree sequence. The complete union U_e of the edge $e \in E_i$ is defined as:*

$$U_e := \{j \in V_1 \mid \exists e_1 \in E_1, ..., e_{i-1} \in E_{i-1} \quad s.th. \quad j \in e_1 \in ... \in e_{i-1} \in e\}. \tag{38}$$

The set:

$$D_e := U_a \cap U_b \tag{39}$$

is called conditioning set D_e of an edge $e = \{a, b\}$ and the conditioned sets $\mathcal{C}_{e,a}$, $\mathcal{C}_{e,b}$ and $\mathcal{C}_e$ are given by:

$$\mathcal{C}_{e,a} := U_a \setminus D_e \ , \quad \mathcal{C}_{e,b} := U_b \setminus D_e \quad and \quad \mathcal{C}_e := \mathcal{C}_{e,a} \cup \mathcal{C}_{e,b} \ . \tag{40}$$

With the notation introduced above we obtain the following conditioning sets:

$$\begin{aligned} T_1: &\qquad D_{\{1,2\}} = \emptyset \ , \quad D_{\{2,3\}} = \emptyset \ , \quad D_{\{3,4\}} = \emptyset \ , \\ T_2: &\qquad D_{\{\{1,2\},\{2,3\}\}} = \{2\} \ , \quad D_{\{\{2,3\},\{3,4\}\}} = \{3\} \ , \\ T_3: &\qquad D_{\{\{\{1,2\},\{2,3\}\},\{\{2,3\},\{3,4\}\}\}} = \{2, 3\} \ . \end{aligned}$$

and the following conditioned sets:

$$\begin{aligned} T_1: &\qquad \mathcal{C}_{\{1,2\}} = \{1, 2\} \ , \quad \mathcal{C}_{\{2,3\}} = \{2, 3\} \ , \quad \mathcal{C}_{\{3,4\}} = \{3, 4\} \ , \\ T_2: &\qquad \mathcal{C}_{\{\{1,2\},\{2,3\}\}} = \{1, 3\} \ , \quad \mathcal{C}_{\{\{2,3\},\{3,4\}\}} = \{2, 4\} \ , \\ T_3: &\qquad \mathcal{C}_{\{\{\{1,2\},\{2,3\}\},\{\{2,3\},\{3,4\}\}\}} = \{1, 4\} \ . \end{aligned}$$

Figure 5 displays a tree sequence with the new notation which is more readable. It still describes the R-vine tree sequence uniquely up to permutation and order of the elements of the set $\mathcal{C}_e$.[11]

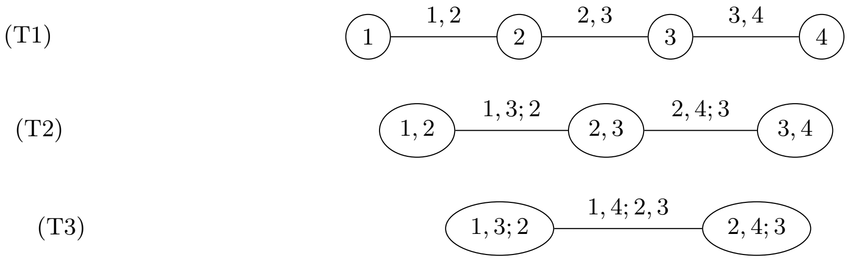


Figure 5: A D-vine on 4 elements with the notation of Definition A.10.

[11] As a convention we order the elements of $\mathcal{C}_e$ in ascending as done in Figure 5.

Definition A.11 (Constraint set)**.** *The constraint set $\mathcal{CV}$ for the R-vine tree sequence $\mathcal{V}$ is defined as:*

$$\mathcal{CV} := \left\{ (\mathcal{C}_{e,a}, \mathcal{C}_{e,b}; D_e) \mid e = \{a, b\},\ e \in E_i \quad \mathit{for} \quad i \in [d-1] \right\} . \tag{41}$$

Here the edge $e = (\mathcal{C}_{e,a}, \mathcal{C}_{e,b}; D_e)$ of the R-vine tree sequence will often be abbreviated by $e = (e_a, e_b; D_e)$.

Here the constraint set is:

$$\left\{ (1,2), (2,3), (3,4), (1,3;2), (2,4;3), (1,4;2,3) \right\} .$$

Note that the curly braces of the conditioned and conditioning sets are left out. This is not completely precise nor consistent to Definition A.11, but facilitates notation. In Figure 5 the round braces are left out as well.

Kurowicka and Cooke (2003) show that any edge of an R-vine tree sequence can be identified by its conditioning or conditioned sets.

Definition A.12 (R-vine specification)**.** *The triple $(\boldsymbol{F}, \mathcal{V}, B)$ is called R-vine specification if:*

(i) $\boldsymbol{F} = (F_1, ..., F_d)$ *is a vector of continuous and invertible distribution functions,*

(ii) $\mathcal{V}$ *is an R-vine tree sequence on d elements and*

(iii) $B := \left\{ C_e \mid e \in E_i \quad \mathit{for} \quad i \in [d-1] \right\}$ *is the set of bivariate copulas C_e with E_i the edge set of tree T_i of the R-vine tree sequence $\mathcal{V}$.*

By this definition each edge $e \in E_i$ of a tree T_i in $\mathcal{V}$ corresponds to a bivariate copula C_e.

Definition A.13 (Realizing an R-vine specification, Regular vine distribution)**.** *A joint distribution F of the random vector $\boldsymbol{X} = (X_1, ..., X_d)$ is said to realize an R-vine specification $(\boldsymbol{F}, \mathcal{V}, B)$ or have a regular vine distribution respectively, if C_e is the bivariate copula of $X_{\mathcal{C}_{e,a}}$ and $X_{\mathcal{C}_{e,b}}$ given $\boldsymbol{X}_{D_e}$ for each edge $e = \{a, b\} \in E_i$ and the marginal distribution of X_i is F_i for $i \in [d]$.*

Remark A.14 (Simplifying assumption)**.** *The assumption that for each edge e of $\mathcal{V}$ the bivariate copula C_e does not depend on the value $\boldsymbol{x}_{D_e}$ the conditioning random vector $\boldsymbol{X}_{D_e}$ takes on is called simplifying assumption.*

Theorem A.15. *Let $(\boldsymbol{F}, \mathcal{V}, B)$ be an R-vine specification on d elements where all pair copulas $C_e \in B$ satisfy the simplifying assumption and have densities c_e. There is a unique distribution F that realizes this R-vine specification with density:*

$$f_{1,...d}(x_1, ..., x_d) = \prod_{i=1}^{d} f_i(x_i) \cdot \tag{42}$$

$$\prod_{i=1}^{d-1} \prod_{e \in E_i} c_{\mathcal{C}_{e,a}, \mathcal{C}_{e,b}; D_e} \left(F_{\mathcal{C}_{e,a}|D_e}(x_{\mathcal{C}_{e,a}} | \boldsymbol{x}_{D_e}), F_{\mathcal{C}_{e,b}|D_e}(x_{\mathcal{C}_{e,b}} | \boldsymbol{x}_{D_e}) \right) , \tag{43}$$

where f_i denote the densities of F_i.

Proof. The proof of theorem can be found in Bedford and Cooke (2001) and Bedford and Cooke (2002). □

Definition A.16 (Regular vine copula)**.** *A (regular) vine copula is a regular vine distribution, where all margins are uniformly distributed on [0, 1].*

Estimating Vine Copulas

In order to estimate a vine copula, first the marginal distributions $F_j,\ j \in [d]$ are estimated. Then the vine tree structure $\mathcal{V}$ and the pair copulas B are selected and estimated tree by tree using Dißmann's Algorithm Dissmann et al. (2013) presented in Algorithm 2, which is a maximum spanning tree algorithm to select $\mathcal{V}$ while maximizing the sum of the edge weights - the absolute Kendall's τ value of the two adjacent random variables.

Algorithm 2 Dißmann's algorithm of Dissmann et al. (2013)

```
Input: $n \in \mathbb{N}$ i.i.d. realizations of the random vector $(X_1, \ldots, X_d)$, i.e. $(x_{i1}, \ldots, x_{id})_{i \in [n]}$
Output: $\mathcal{V}$ and $B$ of R-vine copula specification
Calculate the empirical Kendall's $\tau$ value $\hat{\tau}_{j,\,k}$ for all possible variable pairs $(j, k)$, $1 \leq j < k \leq d$.
Select the spanning tree that maximizes the sum of absolute empirical Kendalls's $\tau$ values, i.e.:
    $$T_1 = \arg \max_{T=(V,E) \text{ in spanning tree}} \sum_{e=(j,k) \in E} |\hat{\tau}_{j,\,k}| \,.$$
For each edge $(j, k)$ in the selected spanning tree, select a copula ad estimate the corresponding parameter(s). Then generate pseudo-observations $\hat{u}_{i,j\,|\,k} := \hat{F}_{j\,|\,k}(x_{ij}\,|\,x_{ik})$ and $\hat{u}_{i,k\,|\,j} := \hat{F}_{k\,|\,j}(x_{ik}\,|\,x_{ij})$, $i \in [n]$ using Equation (35) with the fitted copula $\hat{C}_{jk}$.
for $l \in \{2, \ldots, d-1\}$ do
    For all conditional variable pairs $(j,\, k\,;\, D)$ that can be part of tree $T_l$, i.e. all edges fulfilling the proximity condition (iii) of Definition A.7: calculate the empirical Kendall's $\tau$ value $\hat{\tau}_{j,\,k\,;\,D}\left(\hat{u}_{i,j\,|\,k \cup D}, \hat{u}_{i,k\,|\,j \cup D}\right)$. Denote these edges in the set $E_l^*$.
    Among these edges, select the spanning tree that maximizes the sum of absolute empirical Kendall's $\tau$ values, i.e.:
        $$T_l = \arg \max_{T=(V,E) \text{ in spanning tree with } E \subset E_l^*} \sum_{e=(j,\,k\,;\,D) \in E} |\hat{\tau}_{j,\,k\,;\,D}| \,.$$
    For each edge $(j,\, k\,;\, D)$ in the selected spanning tree $T_l$, select a conditional copula and estimate the corresponding parameter(s). Then generate pseudo-observations $\hat{u}_{i,j\,|\,k \cup D} := \hat{F}_{j\,|\,k \cup D}(x_{ij}\,|\,x_{ik}, x_{iD})$ and $\hat{u}_{i,k\,|\,j \cup D} := \hat{F}_{k\,|\,j \cup D}(x_{ik}\,|\,x_{ij}, x_{iD})$, $i \in [n]$ using Equation (35) with the fitted copula $\hat{C}_{jk;D}$.
end for
```

B Computational Complexity of TVineSynth

B.1 Estimating a C-Vine

It is assumed that the user pre-defines the order of the features, thus the full R-vine matrix of the assumed C-vine is given. It is also assumed that a set of k candidate parametric pair copula families to choose from is specified. Pair copula parameters are estimated with MLE and pair copulas are selected using AIC (Akaike, 1998). In Dißmann's algorithm (Dissmann et al., 2013) which is implemented by Nagler and Vatter (2023) and most commonly used for R-vine model selection, the maximum spanning tree selection is omitted due to the pre-specified R-vine matrix. This leaves us with $\frac{d \cdot (d-1)}{2}$ edges in the un-truncated vine copula model and thus $\frac{d \cdot (d-1)}{2}$ pair copulas to select and estimate. For each edge all k pair copula candidates are estimated with ML and their AIC is computed. Both involve n terms in the log-likelihood evaluation, the MLE involves an optimization that depends on the number of parameters of the current pair copula family[12]. The MLE's computational cost also depends on the optimizer chosen and we can assume that it is constant w.r.t. n, d and k. Then selecting a pair copula out of the k candidates requires finding the minimal AIC among k values which can be solved in $\mathcal{O}(k)$. In total this gives us a computational complexity of $\mathcal{O}(nd^2k)$. As the number of candidate pair copula families usually is small (Nagler and Vatter (2023) implement 10 parametric pair copula families and their rotations excluding the independence copula), k can be considered a constant itself giving a complexity of $\mathcal{O}(nd^2)$.

B.2 Sampling from a C-Vine

The computational complexity of sampling one observation from a C-vine on d variables using Algorithm 6.4 in Czado (2019) taken from Stöber and Czado (2017) is $\mathcal{O}(d^2)$. This gives $\mathcal{O}(nd^2)$ for sampling n observations.

[12] The parametric pair copula families implemented by Nagler and Vatter (2023) which have been used in this work, have up to 2 parameters.

B.3 Computational Complexity of TVineSynth

Regarding the computational complexity of TVineSynth the following points need to be considered:

1. **TVineSynth is estimated only *once*:** Let $T \subset [d]$ be the subset of truncation levels considered. Then in a full run of TVineSynth, the C-vine is not estimated $|T|$ times on the real data, but only *once* at the maximal desired truncation level $t_{max} := maxT$ (may it be $t_{max} = d$, so no truncation or $t_{max} < d$). For all subsequent $t \in T \setminus \{t_{max}\}$ the vine copula is not re-estimated, but obtained by simply setting all pair copulas in tree levels $t' \in T \setminus [t]$ to independence. The computational cost comes from fitting a C-vine once for t_{max} and sampling the estimated C-vine at levels $t \in T$. The computational complexity of estimating and sampling from the C-vine is $\mathcal{O}(nd^2)$ each, see Appendices B.1 and B.2.

2. **AIA privacy and utility evaluation are cheap, MIA is expensive:** The computational costs of utility evaluation and AIA are unproblematic, since both are based on simple model architectures (e.g. linear regression). It is mainly the MIA that drives the computational cost of the privacy evaluation. However, for sufficiently large real data, the vine copula estimation is robust to adding/removing a single observation in the model estimation, as performed under MIA (see the MIA results in Sections 3.1 and Appendix L.3.2). If the MIA PG of a C-vine truncated at level t_{max} is (close to) 1 with little variation, then it will also be so for lower truncation levels. Therefore, the MIA privacy evaluation can be reduced to one truncation level.

3. **Limited number of truncation levels considered:** As illustrated in the real data example it is not at all necessary to perform a privacy and utility evaluation for all possible truncation levels $t \in [d]$. It is sufficient to evaluate every 5th or 10th truncation level. In addition, tree levels that model pairwise (conditional) dependencies with a sensitive feature and all other features should not be considered. This means that we can set $t_{max} := d + 1 - j$ where j is the position of the sensitive feature that enters the C-vine first according to order $\mathcal{O}^*$. So for $d = 26$ and $j = 6$ then $t_{max} := 21$. In sum only a limited set of candidate truncation levels $T \subset [t_{max}]$ has to be considered.

4. **$t_{max} << d$ for high-dimensional real data:** If the real data are high-dimensional, e.g. $d > 400$, we recommend truncating the vine copula model early, because of the statistical uncertainty in the model estimation, so for example $t_{max} := 50$.

5. **All competitors require human-in-the-loop tuning:** Finally, in TVineSynth there is only the truncation level to tune while for the competitor models, specifically CTGAN and TVAE, several hyperparameters need to be tuned (no. epochs, batch size, dimension of the latent space, . . .).

C TVineSynth: Order of Covariates

Let $X_1, ..., X_d$ be the covariates and Y be the response in a prediction task. We propose Algorithm 1 to determine the order $\mathcal{O}^*$ in which the covariates enter the C-vine model, that balances the trade-off between protection against loss of privacy and utility. The order $\mathcal{O}^*$ together with the vine tree structure $\mathcal{V}$ determines the pair copulas and the tree levels they belong to in the C-vine.

Proposition C.1. *Let the order $\mathcal{O}^*$ of the covariates $X_k,\ k \in [d]$ and response Y and let the vine tree structure $\mathcal{V}$ (e.g. in form of a R-vine matrix, see Appendix L.1) be given. Then $\mathcal{O}^*$ together with $\mathcal{V}$ determine which pairwise (conditional) dependencies of $X_k,\ k \in [d]$ and Y are modeled in the C-vine copula.*

Proof. Given the order $\mathcal{O}^*$ and the vine tree structure $\mathcal{V}$, the C-vine is unique. □

This means that $\mathcal{O}^*$ and $\mathcal{V}$ determine which pairwise (conditional) dependencies are cut off from the model when truncating the C-vine at level t. Thus, the definition of $\mathcal{O}^*$ should be such that privacy leaking dependencies are cut off early while those important for the prediction task are cut off when truncating at a very low tree level.

Algorithm 1 In any considered order $\mathcal{O}$ the response Y is in the center of the star-shaped tree at level 1, the response Y is placed at position $(d + 1)$. First, we compute a matrix of pairwise association measures, using for example Pearson correlation or pairwise Kendall's τ. Let $S \subset [d]$ be the set of sensitive covariates. For any sensitive features $X_{j^*},\ j^* \in S$ in turn, we find the covariates $X_k,\ k \in [d] \setminus S$ that show an absolute pairwise

association $|\rho_{j^*,k}|$ above a user defined threshold $\rho^* > 0$ and denote their indices in the set $K_{j^*} \subset [d] \setminus S$. Let $K := \bigcup_{j^* \in S} K_{j^*}$.

The set K depends on the threshold ρ^* on the association measure, for example Pearson correlation greater than $\rho^* := 0.6$ in absolute value for all sensitive covariates (though one can use different thresholds for each sensitive covariate). The more conservative, i.e. lower we set this threshold, the more protected the sensitive covariates will be during truncation. Initializing the algorithm with different values of ρ^* results in different orders $\mathcal{O}^*$ that can be compared through pairwise association or privacy plots on the synthetic data resulting from $\mathcal{O}^*$ at different truncation levels.

The covariates $X_k,\ k \in K$ are the ones leaking most private information on the sensitive features $X_{j^*},\ j^* \in S$. Consequently, disregarding those pairwise dependencies has the highest positive impact on privacy protection. For this reason $X_{j^*},\ j^* \in S$ and $X_k,\ k \in K$ should enter the C-vine copula in a group in the final trees, which are the ones that are truncated first. This means placing them on low indices in the order: for the permutation $\sigma : [d] \to [d]$ giving order $\mathcal{O}^*$ we have $\sigma(k) << d$ for $k \in K$. By grouping the sensitive covariate with the covariates informing it most in the order $\mathcal{O}^*$, we introduce a block structure in the matrix of pairwise association measures. As a result the pairwise (conditional) dependence between $X_{j^*},\ j^* \in S$ and $X_k,\ k \in K$ is modeled in higher tree levels of the C-vine. Specifically, if the positions of X_{j^*} and X_k are $\sigma(j^*)$ and $\sigma(k)$ in $\mathcal{O}^*$ and w.l.o.g. we assume $\sigma(j^*) < \sigma(k)$ in $\mathcal{O}$, then their pairwise dependence conditioned on $X_{(\sigma(k)+1)}, ...X_{(d)}$ is modeled in the $d + 2 - \sigma(k)$th tree in the C-vine. It can be truncated away with truncation level $d + 1 - \sigma(k)$ which will be moderate for $\sigma(k) << d$. Thus the pairwise (conditional) dependence between $X_{j^*},\ j^* \in S$ and $X_k,\ k \in K$ is cut away with a low cost on utility. Results in Sections 3 and 3.1 suggest that it suffices to enforce *conditional* independence between $X_{j^*},\ j^* \in S$ and $X_k,\ k \in K$ to achieve privacy protection. Finally, the appropriateness of the chosen order $\mathcal{O}^*$ is confirmed by plotting the matrix of pairwise association of the C-vine generated synthetic data: the correlation structure is more and more reproduced with increasing truncation level, see Figures 9 and 23. We summarize our procedure in Algorithm 1.

We illustrate our algorithm with the SUPPORT2 example. Figure 6 displays the matrices of pairwise Kendall's τ for covariates of synthetic data generated by a C-vine. Here we use an order $\mathcal{O}_{\text{feature importance}}$ such that a covariate enters the C-vine the earlier the higher its feature importance is. As feature importance measure, we used the mean decrease Gini of a random forest classifier estimated on the real data. We observe that the structure of pairwise association of the real data is almost fully reproduced in the synthetic data already at truncation level 5 of the C-vine. This is a much lower truncation level than compared to when the covariates are ordered following the privacy preserving ordering, as from the Algorithm 1, see Figure 23. AIA results w.r.t. the sensitive feature *totcst* on data ordered according to $\mathcal{O}_{\text{feature importance}}$ in Figure 7 further confirm that $\mathcal{O}_{\text{feature importance}}$ is inferior to the approach suggested above. When we compare the MAB in Figure 7 obtained from $\mathcal{O}_{\text{feature importance}}$ to the AIA results in Figure 2a obtained from an order $\mathcal{O}^*$ as proposed above, we notice that the MAB drastically increases already 10 trees levels earlier (at truncation level 5 as opposed to 15).

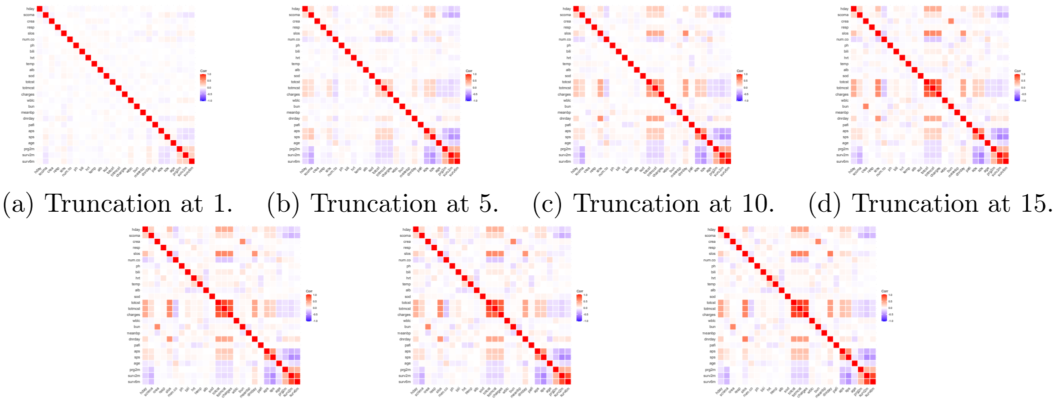
D TVineSynth: Finding the Best Truncation Level As Optimization Task

The selection of the truncation level $t \in T$ giving the best (in terms of the data holder's demands) privacy-utility balance can be optimized if the data holder has a way to place privacy and utility on the same scale. Let P'_t be the privacy score (MIA or AIA) and U'_t the utility score obtained from a C-vine with truncation level $t \in T$. Normalize P'_t and U'_t to obtain $P_t \in [0, 1]$ and $U_t \in [0, 1]$ where higher values for P_t and U_t correspond to better privacy and better utility respectively. Build a privacy-utility score $PU_t := \alpha P_t + (1-\alpha)U_t$ where $\alpha \in [0, 1]$ is user defined to trade-off between privacy and utility. Then finding the best truncation level is a discrete optimization problem in the truncation level t. This is not an easy optimization, as we lack convexity, but it is feasible.

E Limitations

We summarize TVineSynth's main limitations:

- **Privacy guarantees vs. empirical evaluation:** We provide an empirical privacy evaluation of TVineSynth and compare to competitor models. We explain how our algorithm aims to balance privacy and utility. However, TVineSynth does not offer theoretical privacy guarantees in the style of DP. The key idea of



(a) Truncation at 1. (b) Truncation at 5. (c) Truncation at 10. (d) Truncation at 15.

(e) Truncation at 20. (f) No truncation. (g) Real SUPPORT2 data.

Figure 6: The matrices of pairwise Kendall's τ of continuous covariates in synthetic data generated with a C-vine for truncation at levels $t \in \{1, 5, 10, 15, 20\}$ and no truncation when the covariates in the real data are ordered according to $\mathcal{O}_{\text{feature importance}}$ based on feature importance (mean decrease Gini) in a random forest classifier trained on the real SUPPORT2 data. The structure of pairwise association of the real data is almost fully reproduced in the synthetic data already at truncation level 5 of the C-vine. Details on the estimation of the C-vine can be found in Section 2.2 and Appendix J.

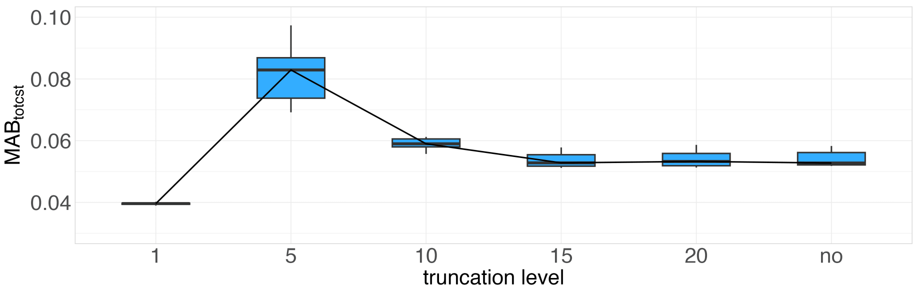


Figure 7: Results of an AIA w.r.t. an order $\mathcal{O}_{\text{feature importance}}$ of the covariates that is based on feature importance (mean decrease Gini) in a random forest classifier trained on the real SUPPORT2 data. The AIA is conducted w.r.t. sensitive covariate *totcst* measured by MAB_{totcst}. Synthetic data are generated with a C-vine for different truncation levels. Results are reported as box plots over 10 AIA game iterations. Parameters of the generative models and privacy attacks can be found in Appendix J.

TVineSynth is to achieve privacy by introducing a targeted bias into the generative model instead of adding noise in a global fashion, which in many cases renders the synthetic data useless for downstream ML applications. For this reason we do not base the design of TVineSynth on DP.

- **Choice of truncation level:** The choice of the level of truncation that provides the preferred balance between privacy and utility is deliberately left to the data owner's decision. This is to account for the fact that privacy requirements vary by context and application and need to be thoroughly weighed against utility demands by data holders, potential users and policy makers. Appendix D discusses how the choice of truncation level can be further automatized.

- **Controlling privacy-utility trade-off:** It is hard to precisely control the privacy-utility trade-off of TVineSynth generated data with the truncation level of the C-vine. For making a well-informed choice the privacy and utility of the vine copula generated data should be evaluated at all truncation levels $t \in T$. Especially the MIA evaluation is costly. However, the candidate truncation levels T can be chosen to minimize the computational cost:

  - Set $t_{max} := d + 1 - j$ or lower where j is the position of the sensitive feature that enters the C-vine first according to $\mathcal{O}^*$, as tree levels that model pairwise (conditional) dependencies with a sensitive feature

  and all other features should not be considered. For d large, set $t_{max} << d$ due to uncertainty in the parameter estimation.
  - It is not necessary to consider all $t < t_{max}$. Instead it suffices to for example set $T := \{1, 5, 10, 15, 25\}$ where $d = 30$. For chosen T it is however necessary to evaluate the vine copula generated data at all $t \in T$.

- **Computational complexity of TVineSynth and its limits for high-dimensional data:** Even though estimating and sampling from a C-vine has moderate computational complexity and TVineSynth is designed to find the optimal privacy-utility balance efficiently, see considerations in Appendix B, evaluating the utility and especially (MIA) privacy of the synthetic data at truncation levels $t \in T$ is computationally demanding. The computational cost can be reduced by making T smaller, but this gives a less nuanced picture of the privacy-utility trade-off. Additionally, estimating a vine copula on data with dimension $d > 500$ becomes computationally challenging. For such settings vine copulas have to be combined with dimension reduction techniques in TVineSynth.

F Competitor Models

F.1 Competitor Models

We benchmark TVineSynth against the following competitor models:

Private Bayes (PrivBayes) Zhang et al. (2017) propose a Bayesian network that satisfies DP guarantees. For a chosen k, they first construct a k-degree Bayesian network in an ϵ_1-differentially private fashion by introducing a score function in the greedy Bayes algorithm. Then, they generate the conditional distributions corresponding to the Bayesian network by injecting Laplacian noise to obtain ϵ_2-DP. The resulting Private Bayes model is $(\epsilon_1 + \epsilon_2)$-differentially private. We use the implementation provided in Stadler et al. (2022), which is patched to fulfill its differentially privacy guarantees.

Private Particle Gradient Descent (PrivPGD) Donhauser et al. (2024) propose a differentially private marginal based generative model that utilizes particle gradient descent. After privately selecting which marginal distributions to estimate, the selected marginal distributions estimated on the real data are privatized with the Gaussian mechanism and transformed to a compact Euclidean space, the embedded space. In the embedded space particles are propagated such that their empirical marginal distribution minimizes the sliced Wasserstein distance, an optimal-transport based divergence, to the embedded privatized marginal distribution of the real data. Note that model estimation and data generation are done in one go and not in two separate steps. As a consequence, PrivPGD does not require model selection or parameter tuning and makes it robust to hyperparameter variation. At the same time it is not possible to sample additional data from PrivPGD once it was estimated, but the full model has to be run again. PrivPGD requires discrete input data and generates discrete synthetic data guaranteeing (ϵ, δ)-DP. This means that, if not discrete, the real data need to be discretized before inputting them into the model and the synthetic data need to be reverted to the original scale afterwards. In order to provide DP, either the real data need to be discrete themselves already or, if not the case, discretization and reversion need to be done in a DP manner. In their experiments on continuous real data Donhauser et al. (2024) solve the discretization by binning of the real data and the reversion by back-transforming the synthetic data to bin means of the real covariates. For this the covariate ranges are inferred from the real data and stored for the reversion process, through which PrivPGD loses its DP guarantees.

In order to retain the DP guarantees also for continuous real data, we therefore infer covariate ranges from a source independent of the real data.

Conditional Tabular GAN (CTGAN) and Tabular Variational Autoencoder (TVAE) To tackle the numerous problems of tabular data when constructing generative adversarial networks (GANs), such as mixed data types, non-Gaussian and multimodal distributions and class imbalance for discrete covariates, Xu et al. (2019) propose the Conditional Tabular GAN (CTGAN). Starting from a GAN, the authors introduce a conditional generator with a modified loss function to account for imbalanced classes and mode-specific normalization to account for non-Gaussian and multimodal distributions. The authors also introduce Tabular Variational Autoencoders (TVAE) by applying the same preprocessing and modified loss functions to a variational autoencoder. We use the wrapper provided in the SDV library around the implementation in the CTGAN library.

R-Vine Copula In an R-vine copula (Joe, 1997; Bedford and Cooke, 2001, 2002; Aas et al., 2009; Joe, 2014; Czado, 2019) the vine tree structure is not pre-specified as in a C- or D-vine, but selected with the Dißmann's algorithm proposed by (Dissmann et al., 2013). An R-vine is therefore the most general and flexible class of vine copulas. Meyer and Nagler (2021) implement a python package for R-vine copula based synthetic data generation.

F.2 Discussing the Choice of Competitor Models

TVineSynth is compared to generative models that focus on preserving privacy of subjects in the real data by providing DP guarantees, and to generative models that focus on reproducing the underlying distribution of the real data closely without offering any formal privacy guarantees. We choose to compare TVineSynth with DP and non-DP competitor models in order to assess which generative model performs best in a context where privacy *and* utility matter.

We compare TVineSynth with CTGAN and TVAE (Xu et al., 2019). They are well established and commonly used generative models for tabular data. CTGAN and TVAE do not offer formal privacy guarantees but focus on generating synthetic data that closely resemble the real data.

PrivBayes is a DP generative model that belongs to the class of graphical probabilistic models as vine copulas do. For this reason we chose PrivBayes as a DP competitor to TVineSynth. Additionally, we chose PrvPGD DP as competitor model as it represents the state-of-the-art for private generative modeling.

Lastly, TVineSynth is compared to an R-vine copula, the most general and flexible vine copula. We do this to assess which impact setting the order of the covariates with Algorithm 1 and setting the vine tree structure to be a C-vine in TVineSynth has on the privacy and utility compared to when both are selected freely in an R-vine. TVineSynth is not compared to the copula-based approaches proposed by Patki et al. (2016), Kumi et al. (2023), Benali et al. (2021), Kamthe et al. (2021) and Chu et al. (2022a) as the latter belong to the same model class as R-vines, but are simpler, less flexible models. We do not compare TVineSynth with the models proposed by Coblenz et al. (2023) and Tagasovska et al. (2019) as we are in a setting where dimension reduction using autoencoders is not necessary to enable modeling the data. For the model proposed by Sun et al. (2019) there is no code available which prohibited a comparison with TVineSynth.

Future work could further compare TVineSynth to tabular denoising diffusion models proposed by Kotelnikov et al. (2023).

G Measures of Privacy

G.1 Privacy Gain (PG)

As a measure of privacy preservation of the synthetic data Stadler et al. (2022) use the PG achieved when publishing a synthetic data set in place of the real given a target observation. The PG is defined as the 'reduction in the attacker's advantage when given access to the synthetic data instead of the real data':

$$PG := Adv\big((X, \boldsymbol{y}), (\boldsymbol{x}_t^T, y_t)\big) - Adv\big((Z, \boldsymbol{w}), (\boldsymbol{x}_t^T, y_t)\big) \ , \tag{44}$$

with target observation $(\boldsymbol{x}_t^T, y_t)$. For MIA the advantages $Adv^{MIA}(\cdot)$ from real and synthetic data are defined as:

$$Adv^{MIA}\big((X, \boldsymbol{y}), (\boldsymbol{x}_t^T, y_t)\big) := P_R(\hat{s}_t = 1 | s_t = 1) - P_R(\hat{s}_t = 1 | s_t = 0) \ , \tag{45}$$

and:

$$Adv^{MIA}\big((Z, \boldsymbol{w}), (\boldsymbol{x}_t^T, y_t)\big) := P_S(\hat{s}_t = 1 | s_t = 1) - P_S(\hat{s}_t = 1 | s_t = 0) \ , \tag{46}$$

respectively, where:

$$s_t := \begin{cases} 1 \, , & (\boldsymbol{x}_t^T, y_t) \text{ is in } (X, \boldsymbol{y}) \\ 0 & \text{else} \end{cases} \ . \tag{47}$$

The attacker's guess is $\hat{s}_t$ and P_R (P_S) indicates that the attacker's guess is based on the real (synthetic) data. Obviously, $Adv^{MIA}\big((X, \boldsymbol{y}), (\boldsymbol{x}_t^T, y_t)\big) = 1$ as the attacker can look up in the real data whether the target observation is present. Together with the theoretical bounds given in Yeom et al. (2018):

$$Adv^{MIA}\big((Z, \boldsymbol{w}), (\boldsymbol{x}_t^T, y_t)\big) \leq e^{\epsilon} - 1 \, , \tag{48}$$

we get that for a differentially private generative model the center is bounded by:

$$PG^{MIA} \geq 2 - e^{\epsilon} \, . \tag{49}$$

For AIA, Stadler et al. (2022) define the advantages $Adv^{AIA}(\cdot)$ from real and synthetic data as:

$$Adv^{AIA}\big((X, \boldsymbol{y}), (\boldsymbol{x}_{t,-j^*}^T, y_t)\big) := P_R(\hat{x}_{t,j^*} = x_{t,j^*} | s_t = 1) - P_R(\hat{x}_{t,j^*} = x_{t,j^*} | s_t = 0) \, , \tag{50}$$

and:

$$Adv^{AIA}\big((Z, \boldsymbol{w}), (\boldsymbol{x}_{t,-j^*}^T, y_t)\big) := P_S(\hat{x}_{t,j^*} = x_{t,j^*} | s_t = 1) - P_S(\hat{x}_{t,j^*} = x_{t,j^*} | s_t = 0) \, , \tag{51}$$

respectively, where $(\boldsymbol{x}_{t,-j^*}^T, y_t)$ is a sub-vector of $(\boldsymbol{x}_t^T, y_t)$ indicating that the sensitive feature value x_{t,j^*} for some $j^* \in S \subset [d]$ of $(\boldsymbol{x}_t^T, y_t)$ is unknown to the attacker and $\hat{x}_{t,j^*}$ is the attacker's estimate of x_{t,j^*}.

G.2 Mean Squared Error (MSE)

The definition of $Adv^{AIA}(\cdot)$ by Stadler et al. (2022) in Equations 50 and 51 is correct for sensitive features X_{j^*} taking on finitely many values. For the continuous case it is wrong, as $P(\hat{x}_{t,j^*} = x_{t,j^*} | s_t = s) = 0$ for any guess $\hat{x}_{t,j^*}$ and taking densities instead, as done in the implementation by Stadler et al. (2022) provided on github, is also incorrect. Olatunji et al. (2023) instead suggest to compute the mean squared error (MSE) to asses the success of an AIA. It may be calculated by generating K samples from the synthetic data generator g and K bootstrap samples of the real data, standardizing them by subtracting the mean and dividing by the standard deviation and then computing:

$$MSE_R(x_{t,j^*} | s_t = s) := \frac{1}{K} \sum_{k=1}^{K} \big(\hat{x}_{t,j^*}(R)^{(k)} - x_{t,j^*}\big)^2 \tag{52}$$

$$MSE_S(x_{t,j^*} | s_t = s) := \frac{1}{K} \sum_{k=1}^{K} \big(\hat{x}_{t,j^*}(S)^{(k)} - x_{t,j^*}\big)^2 \tag{53}$$

where $\hat{x}_{t,j^*}(S)^{(k)}$ is the attacker's guess based on the kth standardized synthetic data set sampled from the vine copula, $\hat{x}_{t,j^*}(R)^{(k)}$ is the attacker's guess based on the kth standardized bootstrap sample from the real data, $k \in [K]$ and s_t is defined as in (47).

However, the MSE gives an incomplete picture of a generative model's AIA privacy: A high MSE indicates that the attacker guesses a value $\hat{x}_{t,j^*}$ which is on average far from the actual sensitive feature value x_{t,j^*} in squared error loss. Thus, privacy protection w.r.t. AIA is high. Concluding from a low MSE that the AIA privacy is low is however not generally correct. This is illustrated in the following example on simulated real data: Figure 21 shows a low MSE for all sensitive covariates if the target observations are randomly sampled (in blue). This is because the randomly sampled target observations are closer to the center of the marginal distribution of the respective sensitive covariate X_{j^*}. We find that the attacker's guess is merely the mean of the corresponding sensitive covariate X_{j^*}. In terms of the attacker's regression model estimated on the synthetic data this means that the all the regression coefficients X_k with $k \in [d] \setminus \{j^*\}$ are approximately 0. Hence, $X_k, \; k \in [d] \setminus \{j^*\}$ do not inform the sensitive covariate X_{j^*} and the attacker learns no privacy leaking dependencies but only general statistics from the synthetic data. This case therefore poses no privacy risk. See further details on the example in Appendix L.3.7.

G.3 Mean Absolute β-Coefficients (MAB)

The previous example made clear that a low MSE does not necessarily indicate low AIA privacy. Instead we define the *mean absolute β-coefficient (MAB)*. The definition of the MAB is based on the AIA game proposed by

Stadler et al. (2022). There it is assumed that the attacker knows the generative model class used to generate synthetic data. In each of the N game iterations the attacker gets access to a subsample of the real data of fixed size. On this subsample the attacker fits the generative model and generates n_{synth} synthetic data sets. On each synthetic data set the attacker then estimates a regression model regressing the sensitive covariate on the non-sensitive covariates. Thus, in a whole AIA privacy evaluation for a fixed sensitive feature X_{j^*} we obtain regression coefficients $\beta^{(j^*)}_{k,m,l}$ with $k \in [d]$ is the index of all other covariates/features (N.B.: The intercept term $\hat{\beta}^{(j^*)}_{0,m,l}$ is not included in the definition of the MAB.), $m \in [N]$ is the index of the game iteration, and $l \in [n_{synth}]$ runs over all generated synthetic data sets. We then define the MAB as in Definition 2.2:

$$MAB_{j^*} := \frac{1}{dNn_{synth}} \sum_{k\in[d]} \sum_{m\in[N]} \sum_{l\in[n_{synth}]} |\hat{\beta}^{(j^*)}_{k,m,l}| \ . \tag{3}$$

Lower values for MAB indicate that covariates $X_k,\ k \neq j^*$ inform the sensitive covariate X_{j^*} less. As opposed to the MSE, the definition in Equation 2.2 is independent of a target observation and thus quantifies AIA privacy in terms of the generative model.

We discuss how the MAB might behave in the case of collinearity of the covariates $X_k,\ k \neq j^*$. Then we could encounter a scenario in which the MAB *and* the MSE take on a high values. A high MAB value lets us conclude that the covariates $X_k,\ k \neq j^*$ inform the sensitive feature X_{j^*} well hinting on privacy leakage, while a high MSE on the contrary indicates that the attacker's guess is on average far from the actual sensitive feature value in squared error loss. Speaking in hypothesis testing terms this case represents a type II, where the MAB indicates privacy leakage when in fact the MSE confirms that there is not.

G.4 Worst-Case Absolute β-Coefficients (WCAB)

Exchanging the mean in the MAB with the maximum we obtain the worst-case absolute β-coefficients (WCAB):

$$WCAB_{j^*} := \max\{|\hat{\beta}^{(j^*)}_{k,m,l}| : \ k \in [d],\ m \in [N],\ l \in [n_{synth}]\} \ . \tag{54}$$

The WCAB gives a worst-case evaluation of the AIA privacy for all individuals following the idea of the worst-case guarantees provided by DP (Dwork et al., 2014).

G.5 Mean R^2 (MR2)

Finally, the degree of privacy required has to be decided by the data holder and varies from application to application. The MAB uses estimated β-coefficients of each feature in the relevant regression model and has therefore a scale which is difficult to interpret. Instead the R^2 can be used, which gives the percentage of variance explained by a regression model and is therefore more interpretable. Like for the MAB, an average over the R^2 values in all performed regressions can be computed and we call it MR2, Mean R^2. It can be shown that regression coefficients of features entering the C-vine late start to vanish with increasing truncation. Therefore the number of degrees of freedom in the regression model varies for different truncation levels of the C-vine. This requires adjusting the MR2 for different number of degrees of freedom according to the truncation level and makes the MR2 harder to compare accross generative models. For this reason we focus on the MAB instead of the MR2.

H Proofs of Theoretical Results Concerning the Utility and Privacy of TVineSynth

Proof of Theorem 2.3. For the log-odds ratio, we have:

$$P(Y=1|\boldsymbol{X}=\boldsymbol{x}) = \frac{\pi_Y f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=1;\boldsymbol{\theta})}{\pi_Y f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=1;\boldsymbol{\theta}) + (1-\pi_Y) f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=0;\boldsymbol{\theta})} \tag{55}$$

and:

$$P(Y=0|\boldsymbol{X}=\boldsymbol{x}) = \frac{(1-\pi_Y) f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=0;\boldsymbol{\theta})}{\pi_Y f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=1;\boldsymbol{\theta}) + (1-\pi_Y) f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=0;\boldsymbol{\theta})} \ , \tag{56}$$

so that:

$$\psi(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d; \boldsymbol{x}) = \log \frac{\pi_Y}{1-\pi_Y} + \log \frac{f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=1;\boldsymbol{\theta})}{f_{\boldsymbol{x}|y}(\boldsymbol{x}|Y=0;\boldsymbol{\theta})} \ . \tag{57}$$

Further, we have:

$$f_{\boldsymbol{x}|y}(\boldsymbol{x}|y) = f_{d|y}(x_d|y) \cdot f_{d-1|d,y}(x_{d-1}|x_d, y) \cdot \ldots \cdot f_{1|2,\ldots,d,y}(x_1|x_2, \ldots, x_d, y) \tag{58}$$

where, omitting arguments for simplicity:

$$f_{d-1|d,y} = \frac{f_{d-1,d|y}}{f_{d|y}} = \frac{c_{d-1,d;y} f_{d-1|y} f_{d|y}}{f_{d|y}} = c_{d-1,d;y} f_{d-1|y} \tag{59}$$

$$f_{d-2|d-1,d,y} = \frac{f_{d-2,d-1|d,y}}{f_{d-1|d,y}} = \frac{c_{d-2,d-1;d,y} f_{d-2|d,y} f_{d-1|d,y}}{f_{d-1|d,y}} = c_{d-2,d-1;d,y} f_{d-2|d,y}, \tag{60}$$

where, correspondingly to $f_{d-1|d,y}$, we obtain $f_{d-2|d,y} = c_{d-2,d;y} f_{d-2|y}$, so that:

$$f_{d-2|d-1,d,y} = c_{d-2,d-1;d,y} c_{d-2,d;y} f_{d-2|y} \ . \tag{61}$$

Further:

$$f_{d-3|d-2,d-1,d,y} = \frac{f_{d-3,d-2|d-1,d,y}}{f_{d-2|d-1,d,y}} = \frac{c_{d-3,d-2;d-1,d,y} f_{d-3|d-1,d,y} f_{d-2|d-1,d,y}}{f_{d-2|d-1,d,y}} \tag{62}$$

$$= c_{d-3,d-2;d-1,d,y} f_{d-3|d-1,d,y} \ , \tag{63}$$

where, correspondingly to $f_{d-2|d-1,d,y}$, we obtain $f_{d-3|d-1,d,y} = c_{d-3,d-1;d,y} c_{d-3,d;y} f_{d-3|y}$, so that:

$$f_{d-3|d-2,d-1,d,y} = c_{d-3,d-2;d-1,d,y} c_{d-3,d-1;d,y} c_{d-3,d;y} f_{d-3|y} \ . \tag{64}$$

Continuing this we obtain:

$$f_{1|2,\ldots,d,y} = c_{1,2;3,\ldots,d,y} \cdot \ldots \cdot c_{1,d;y} f_{1|y} \ . \tag{65}$$

Hence,

$$f_{\boldsymbol{x}|y}(\boldsymbol{x}|y) = \prod_{j=1}^{d} f_{j|y} \prod_{t=2}^{d} \prod_{j=1}^{d+1-t} c_{j,d+2-t;d+3-t,\ldots,d,y} \ , \tag{66}$$

which is so that:

$$\psi = \log \frac{\pi_Y}{1-\pi_Y} + \sum_{j=1}^{d} \log \frac{f_{j|y}(x_j|1)}{f_{j|y}(x_j|0)} + \sum_{t=2}^{d} \sum_{j=1}^{d+1-t} \log \frac{c^1_{j,d+2-t;d+3-t,\ldots,d,y}}{c^0_{j,d+2-t;d+3-t,\ldots,d,y}} = \sum_{t=1}^{d} \psi_t \ , \tag{67}$$

where $c^k_{j,d+2-t;d+3-t,\dots,d,y}$ is evaluated at $(\boldsymbol{x}, y) = (\boldsymbol{x}, k)$, with:

$$\psi_1 = \log \frac{\pi_Y}{1-\pi_Y} + \sum_{j=1}^{d} \log \frac{f_{j|y}(x_j|1)}{f_{j|y}(x_j|0)} \tag{68}$$

and:

$$\psi_t = \sum_{j=1}^{d+1-t} \log \frac{c^1_{j,d+2-t;d+3-t,\dots,d,y}}{c^0_{j,d+2-t;d+3-t,\dots,d,y}} \;, \quad t \in \{2, \dots, d\} \;. \tag{69}$$

When truncating the C-vine at level τ, then:

$$f_{\boldsymbol{x}|y}(\boldsymbol{x}|y) = \prod_{j=1}^{d} f_{j|y} \prod_{t=2}^{\tau} \prod_{j=1}^{d+1-t} c_{j,d+2-t;d+3-t,\dots,d,y} \;, \tag{70}$$

so that the corresponding log-odds ratio is given by $\psi^\tau = \sum_{t=1}^{\tau} \psi_t$.

Further:

$$\begin{aligned}
f_{j|y}(x_j|y) = \frac{\partial}{\partial x_j} F_{j|y}(x_j|y) &= \frac{\partial}{\partial x_j} \frac{P(X_j \leq x_j, Y = y)}{P(Y = y)} && (71)\\
&= \frac{\partial}{\partial x_j} \frac{P(X_j \leq x_j, Y \leq y) - P(X_j \leq x_j, Y \leq y-1)}{P(Y = y)} && (72)\\
&= \frac{\partial}{\partial x_j} \frac{C_{j,y}\big(F_j(x_j), F_Y(y)\big) - C_{j,y}\big(F_j(x_j), F_Y(y-1)\big)}{P(Y = y)} && (73)\\
&= \frac{1}{P(Y = y)} \Big(h_{y|j}\big(F_Y(y)|F_j(x_j)\big) - h_{y|j}\big(F_Y(y-1)|F_j(x_j)\big) \Big) f_j(x_j) \;, && (74)
\end{aligned}$$

where $h_{y|j} = \frac{\partial C_{j,y}}{\partial F_j(x_j)}$, and since $X_j \sim U(0,1)$, for $j \in [d]$:

$$f_{j|y}(x_j|y) = \begin{cases} \frac{1}{1-\pi_Y} h_{y|j}\big(1-\pi_Y|F_j(x_j)\big) \;, & y = 0 \\ \frac{1}{\pi_Y}\Big(1 - h_{y|j}\big(1-\pi_Y|F_j(x_j)\big)\Big) \;, & y = 1 \;, \end{cases} \tag{75}$$

and the arguments of the pair copulas are given by (Joe, 1997):

$$\begin{aligned}
&F_{k|d+2-t,\dots,d,y}(x_k|x_{d+2-t}, \dots, x_d, y) = && (76)\\
&\quad \frac{\partial C_{k,d+2-t;d+3-t,\dots,d,y}\big(F_{k|d+3-t,\dots,d,y}(x_k|x_{d+3-t}, \dots, x_d, y), F_{d+2-t|d+3-t,\dots,d,y}(x_{d+2-t}|x_{d+3-t}, \dots, x_d, y)\big)}{\partial F_{d+2-t|d+3-t,\dots,d,y}(x_{d+2-t}|x_{d+3-t}, \dots, x_d, y)} \;, && (77)
\end{aligned}$$

for $k \in [d+1-t]$ and $t \in [d]$.

We see that distributions $f_{j|y}$ depend on π_Y, but also on the parameter of the copula $C_{j,y}$ of the first tree of the C-vine. This means that the first term ψ_1 of the log-odds ratio depends on π_Y and the copula parameters $\boldsymbol{\theta}_1$ of the first tree. The remaining terms ψ_t, $t \in \{2, \dots, d\}$ are functions of the pair copulas in trees 2 to d, that have conditional distributions as arguments, which are computed recursively, as shown above. This means that ψ_t depends on the parameters $\boldsymbol{\theta}_t$ of the pair copulas of tree number t, but also on the parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{t-1}$ from the previous trees, as well as π_Y, though the recursion.

Under the usual regularity assumptions, consult for instance Lehmann (1999), we have for large n that the maximum likelihood estimator is:

$$\begin{pmatrix} \hat{\pi}_Y \\ \hat{\boldsymbol{\theta}} \end{pmatrix} = \begin{pmatrix} \pi_Y \\ \boldsymbol{\theta} \end{pmatrix} + \boldsymbol{J}^{-1} \bar{\boldsymbol{U}}_n + \mathcal{O}_P\left(\frac{1}{\sqrt{n}}\right) , \tag{78}$$

where:

$$\bar{\boldsymbol{U}}_n = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)} \log f(\boldsymbol{X}_i, Y_i) \tag{79}$$

and:

$$\sqrt{n} \bar{\boldsymbol{U}}_n \xrightarrow{d} \mathcal{N}_{|\boldsymbol{\theta}|+1}(\boldsymbol{0}, \boldsymbol{J}) \tag{80}$$

and:

$$\boldsymbol{J} = -E\Big[\frac{\partial^2}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d) \partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)^T} \log f(\boldsymbol{X}, Y)\Big] , \tag{81}$$

and the delta method gives:

$$\begin{aligned} \hat{\psi} &= \psi(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_d; \boldsymbol{x}) \qquad (82)\\ &= \psi(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d; \boldsymbol{x}) + \frac{1}{\sqrt{n}} \boldsymbol{v}^T \boldsymbol{J}^{-1} \sqrt{n} \bar{\boldsymbol{U}}_n + \mathcal{O}_P\left(\frac{1}{\sqrt{n}}\right), \qquad (83) \end{aligned}$$

where $\boldsymbol{v} = \frac{\partial \psi}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)}$.

Hence, for large n:

$$MSE(\hat{\psi}) = E\big[(\hat{\psi} - \psi)^2\big] = \frac{1}{n} \cdot \boldsymbol{v}^T \boldsymbol{J}^{-1} \boldsymbol{v} + \mathcal{O}\left(\frac{1}{n}\right) . \tag{84}$$

Further, when we truncate the C-vine at level $\tau \leq d-1$, we simply set all pair copulas from level $\tau+1$ to d to independence, but the models parameters $(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_\tau)$ of the truncated model are not re-estimated. Thus:

$$\begin{aligned} \tilde{\psi}^\tau &= \sum_{t=1}^{\tau} \psi_t(\hat{\pi}_Y, \hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_t; \boldsymbol{x}) \qquad (85)\\ &= \sum_{t=1}^{\tau} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t; \boldsymbol{x}) + \frac{1}{\sqrt{n}} \left(\frac{\partial \sum_{t=1}^{\tau} \psi_t}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d)}\right)^T \boldsymbol{J}^{-1} \sqrt{n} \boldsymbol{U}_n + \mathcal{O}_P\left(\frac{1}{\sqrt{n}}\right) \qquad (86)\\ &= \psi + \left(\sum_{t=1}^{\tau} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t; \boldsymbol{x}) - \psi\right) \qquad (87)\\ &+ \frac{1}{\sqrt{n}} \begin{pmatrix} \boldsymbol{v}^{1\ldots\tau} \\ \boldsymbol{0} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{J}^{1\ldots\tau,1\ldots\tau} & \boldsymbol{J}^{1\ldots\tau,\tau+1\ldots d} \\ \boldsymbol{J}^{\tau+1\ldots d,1\ldots\tau} & \boldsymbol{J}^{\tau+1\ldots d,\tau+1\ldots d} \end{pmatrix} \sqrt{n} \boldsymbol{U}_n + \mathcal{O}_P\left(\frac{1}{\sqrt{n}}\right) , \qquad (88) \end{aligned}$$

where:

$$\boldsymbol{v}^{1\ldots\tau} = \frac{\partial}{\partial(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau)} \sum_{t=1}^{\tau} \psi_t \tag{89}$$

and the diagonal blocks $\boldsymbol{J}^{1\ldots\tau,1\ldots\tau}$ and $\boldsymbol{J}^{\tau+1\ldots d,\tau+1\ldots d}$ of $\boldsymbol{J}^{-1}$ correspond to the double derivatives with respect to $(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau)$ and $(\boldsymbol{\theta}_{\tau+1}, \ldots, \boldsymbol{\theta}_d)$, respectively, and the off-diagonal blocks $\boldsymbol{J}^{1\ldots\tau,\tau+1\ldots d}$ and $\boldsymbol{J}^{\tau+1\ldots d,1\ldots\tau}$ to the derivative with respect to $(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_\tau)$ and $(\boldsymbol{\theta}_{\tau+1}, \ldots, \boldsymbol{\theta}_d)$. This means that for large n:

$$\begin{aligned} MSE(\tilde{\psi}^\tau) &= \left(\sum_{t=1}^{\tau} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_i; \boldsymbol{x}) - \psi\right)^2 + \frac{1}{n} \cdot \left(\boldsymbol{v}^{1\ldots\tau}\right)^T \boldsymbol{J}^{1\ldots\tau,1\ldots\tau} \boldsymbol{v}^{1\ldots\tau} + \mathcal{O}\left(\frac{1}{n}\right) && (90) \\ &= \left(\sum_{t=\tau+1}^{d} \psi_t(\pi_Y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_i; \boldsymbol{x})\right)^2 + \frac{1}{n} \cdot \left(\boldsymbol{v}^{1\ldots\tau}\right)^T \boldsymbol{J}^{1\ldots\tau,1\ldots\tau} \boldsymbol{v}^{1\ldots\tau} + \mathcal{O}\left(\frac{1}{n}\right) . && (91) \end{aligned}$$

□

Proof of Theorem 2.4. Since the rows of $\boldsymbol{V}$ are independent and follow a standard $(d+1)$-variate normal distribution with correlation matrix $\boldsymbol{\rho}$, we know that for each row i:

$$\begin{aligned} & V_{ij^*} | \boldsymbol{V}_{i,[d+1]\setminus\{j^*\}} = \boldsymbol{v}_{i,[d+1]\setminus\{j^*\}} && (92) \\ & \sim \mathcal{N}(\boldsymbol{\rho}^T_{[d+1]\setminus\{j^*\},j^*} \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{v}_{i,[d+1]\setminus\{j^*\}}, 1 - \boldsymbol{\rho}^T_{[d+1]\setminus\{j^*\},j^*} \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{\rho}_{[d+1]\setminus\{j^*\},j^*}) , && (93) \end{aligned}$$

so that we may write:

$$V_{ij^*} = \boldsymbol{\rho}^T_{[d+1]\setminus\{j^*\},j^*} \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{v}_{i,[d+1]\setminus\{j^*\}} + \varepsilon_i = (\boldsymbol{\beta}^{(j^*)})^T \boldsymbol{v}_{i,[d+1]\setminus\{j^*\}} + \varepsilon_i, \tag{94}$$

with $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, (\sigma^{(j^*)})^2)$ and:

$$(\sigma^{(j^*)})^2 = 1 - \boldsymbol{\rho}^T_{[d+1]\setminus\{j^*\},j^*} \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{\rho}_{[d+1]\setminus\{j^*\},j^*} \ . \tag{95}$$

Then, it follows from the properties of the ordinary least squares estimator that $\hat{\boldsymbol{\beta}}^{(j^*)}$ follows a d-variate normal distribution with mean:

$$\boldsymbol{\beta}^{(j^*)} = \boldsymbol{\rho}^{-1}_{[d+1]\setminus\{j^*\},[d+1]\setminus\{j^*\}} \boldsymbol{\rho}_{[d+1]\setminus\{j^*\},j^*} \ , \tag{96}$$

and covariance matrix:

$$(\sigma^{(j^*)})^2 (\boldsymbol{V}^T_{[d+1]\setminus\{j^*\}} \boldsymbol{V}_{[d+1]\setminus\{j^*\}})^{-1} \ . \tag{97}$$

Now, assume first that the C-vine is truncated at level $\tau = d + 1 - j^*$. This means that all pair copulas in tree levels $t \in \{d + 2 - j^*, \ldots, d\}$ are set to independence, and since they are all Gaussian, this is the same as setting the corresponding partial correlations to 0, i.e.:

$$\rho_{12\cdot3\ldots d+1} = \rho_{13\cdot4\ldots d+1} = \rho_{23\cdot4\ldots d+1} = \ldots = \rho_{1j^*\cdot j^*+1\ldots d+1} = \ldots = \rho_{j^*-1,j^*\cdot j^*+1\ldots d+1} = 0\,. \tag{98}$$

These partial correlations may be expressed in terms of the partial variance-covariance matrix (consult for instance Baba et al. (2004)). For this, let $k \in [j-1]$ for some $j \in \{2, ..., j^*\}$. Then partial variance-covariance matrix is given by:

$$\boldsymbol{\rho}_{kj\cdot j+1\ldots d+1} \tag{99}$$

$$= \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \tag{100}$$

$$= \begin{pmatrix} 1 & \rho_{kj} \\ \rho_{kj} & 1 \end{pmatrix} - \begin{pmatrix} \boldsymbol{\rho}^T_{j+1\ldots d+1,k} \\ \boldsymbol{\rho}^T_{j+1\ldots d+1,j} \end{pmatrix} \boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1} \begin{pmatrix} \boldsymbol{\rho}_{j+1\ldots d+1,k} & \boldsymbol{\rho}_{j+1\ldots d+1,j} \end{pmatrix} \tag{101}$$

$$= \begin{pmatrix} 1 - \boldsymbol{\rho}^T_{j+1\ldots d+1,k}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,k} & \rho_{kj} - \boldsymbol{\rho}^T_{j+1\ldots d+1,k}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,j} \\ \rho_{kj} - \boldsymbol{\rho}^T_{j+1\ldots d+1,j}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,k} & 1 - \boldsymbol{\rho}^T_{j+1\ldots d+1,j}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,j} \end{pmatrix}\,. \tag{102}$$

The partial correlation is then:

$$\rho_{kj\cdot j+1\ldots d+1} = \frac{a_{12}}{\sqrt{a_{11}a_{22}}} \tag{103}$$

$$= \frac{\rho_{kj} - \boldsymbol{\rho}^T_{j+1\ldots d+1,k}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,j}}{\sqrt{(1 - \boldsymbol{\rho}^T_{j+1\ldots d+1,k}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,k})(1 - \boldsymbol{\rho}^T_{j+1\ldots d+1,j}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,j})}} \tag{104}$$

$$= 0\,, \tag{105}$$

which is equivalent to the numerator being 0, i.e.:

$$\rho_{kj} = \boldsymbol{\rho}^T_{j+1\ldots d+1,k}\boldsymbol{\rho}^{-1}_{j+1\ldots d+1,j+1\ldots d+1}\boldsymbol{\rho}_{j+1\ldots d+1,j}\,. \tag{106}$$

This holds specifically for $j = j^*$ and any $k \in [j^* - 1]$, hence:

$$\boldsymbol{\rho}_{1\ldots j^*-1,j^*} - \boldsymbol{\rho}^T_{j^*+1\ldots d+1,1\ldots j^*-1}\boldsymbol{\rho}^{-1}_{j^*+1\ldots d+1,j^*+1\ldots d+1}\boldsymbol{\rho}_{j^*+1\ldots d+1,j^*} = \mathbf{0}\,. \tag{107}$$

Further, if we express:

$$\boldsymbol{\rho}_{[d+1]\backslash\{j^*\},[d+1]\backslash\{j^*\}} = \begin{pmatrix} \boldsymbol{\rho}_{1\ldots j^*-1,1\ldots j^*-1} & \boldsymbol{\rho}^T_{j^*+1\ldots d+1,1\ldots j^*-1} \\ \boldsymbol{\rho}_{j^*+1\ldots d+1,1\ldots j^*-1} & \boldsymbol{\rho}_{j^*+1\ldots d+1,j^*+1\ldots d+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{C}^T \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}\,, \tag{108}$$

then we have:

$$\boldsymbol{\rho}^{-1}_{[d+1]\backslash\{j^*\},[d+1]\backslash\{j^*\}} = \begin{pmatrix} \boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\boldsymbol{C}^T\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1}\boldsymbol{C}^T\boldsymbol{D}^{-1} \end{pmatrix}\,, \tag{109}$$

with $\boldsymbol{M} = \boldsymbol{B} - \boldsymbol{C}^T\boldsymbol{D}^{-1}\boldsymbol{C}$. We use this to obtain that the regression coefficient of V_{j^*} when the C-vine is truncated at level τ:

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)} = \begin{pmatrix} \boldsymbol{\beta}^{(j^*)}_{(\tau)\ 1\dots d-\tau} \\ \boldsymbol{\beta}^{(j^*)}_{(\tau)\ d+1-\tau\dots d} \end{pmatrix} \overset{(96)}{=} \boldsymbol{\rho}^{-1}_{[d+1]\backslash\{j^*\},[d+1]\backslash\{j^*\}} \boldsymbol{\rho}_{[d+1]\backslash\{j^*\},j^*} \tag{110}$$

$$\overset{(109)}{=} \begin{pmatrix} \boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\boldsymbol{C}^T\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1}\boldsymbol{C}^T\boldsymbol{D}^{-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{\rho}_{1\dots j^*-1,j^*} \\ \boldsymbol{\rho}_{j^*+1\dots d+1,j^*} \end{pmatrix} \tag{111}$$

$$= \begin{pmatrix} \boldsymbol{M}^{-1}(\boldsymbol{\rho}_{1\dots j^*-1,j^*} - \boldsymbol{\rho}^T_{j^*+1\dots d+1,1\dots j^*-1}\boldsymbol{\rho}^{-1}_{j^*+1\dots d+1,j^*+1\dots d+1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*}) \\ \boldsymbol{D}^{-1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*} - \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{M}^{-1}(\boldsymbol{\rho}_{1\dots j^*-1,j^*} - \boldsymbol{\rho}^T_{j^*+1\dots d+1,1\dots j^*-1}\boldsymbol{\rho}^{-1}_{j^*+1\dots d+1,j^*+1\dots d+1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*}) \end{pmatrix} \tag{112}$$

$$\overset{(107)}{=} \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\rho}^{-1}_{j^*+1\dots d+1,j^*+1\dots d+1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*} \end{pmatrix} , \tag{113}$$

where the subscript (τ) indicates the specific truncation level of the C-vine. Note that the regression coefficients $\boldsymbol{\beta}^{(j^*)}_{(\tau)} \in \mathbb{R}^d$ are indexed from 1 to d. This means that for any $k > j^*$ the coefficient for v_k is $\beta^{(j^*)}_{(\tau)\ k-1}$, i.e.:

$$V_{j^*} = \sum_{k=1}^{j^*-1} \beta^{(j^*)}_{(\tau)\ k} v_k + \sum_{k=j^*+1}^{d} \beta^{(j^*)}_{(\tau)\ k-1} v_k + \varepsilon \ . \tag{114}$$

Finally, we have:

$$(\sigma^{(j^*)}_{(\tau)})^2 \overset{(95)}{=} 1 - \boldsymbol{\rho}^T_{[d+1]\backslash\{j^*\},j^*}\boldsymbol{\rho}^{-1}_{[d+1]\backslash\{j^*\},[d+1]\backslash\{j^*\}}\boldsymbol{\rho}_{[d+1]\backslash\{j^*\},j^*} \tag{115}$$

$$\overset{(113)}{=} 1 - \begin{pmatrix} \boldsymbol{\rho}^T_{1\dots j^*-1,j^*} & \boldsymbol{\rho}^T_{j^*+1\dots d+1,j^*} \end{pmatrix} \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\rho}^{-1}_{j^*+1\dots d+1,j^*+1\dots d+1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*} \end{pmatrix} \tag{116}$$

$$= 1 - \boldsymbol{\rho}^T_{j^*+1\dots d+1,j^*}\boldsymbol{\rho}^{-1}_{j^*+1\dots d+1,j^*+1\dots d+1}\boldsymbol{\rho}_{j^*+1\dots d+1,j^*} \ . \tag{117}$$

If we assume that the C-vine is truncated at level $\tau < d + 1 - j^*$, we can reformulate the results of (113) and (117) in terms of a general truncation level τ and obtain:

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\ 1\dots d-\tau} = \boldsymbol{0} \ , \tag{118}$$

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\ d+1-\tau\dots d} = \boldsymbol{\rho}^{-1}_{d-\tau+2\dots d+1,d-\tau+2\dots d+1}\boldsymbol{\rho}_{d-\tau+2\dots d+1,j^*} \ , \tag{119}$$

$$(\sigma^{(j^*)}_{(\tau)})^2 = 1 - \boldsymbol{\rho}^T_{d-\tau+2\dots d+1,j^*}\boldsymbol{\rho}^{-1}_{d-\tau+2\dots d+1,d-\tau+2\dots d+1}\boldsymbol{\rho}_{d-\tau+2\dots d+1,j^*} \ . \tag{120}$$

Let us now assume that we truncate away one more tree, i.e. truncate the C-vine at level $\tau - 1$. Then among others the partial correlation:

$$\rho_{j^* d+2-\tau \cdot d+3-\tau\dots d+1} = 0 \ . \tag{121}$$

Proceeding in the same way as for $\rho_{kj\cdot j+1\dots d+1}$ in Equations (103) to (106), it is easily shown that:

$$\rho_{j^*,d+2-\tau} = \boldsymbol{\rho}^T_{d+3-\tau\dots d+1,j^*}\boldsymbol{\rho}^{-1}_{d+3-\tau\dots d+1,d+3-\tau\dots d+1}\boldsymbol{\rho}_{d+3-\tau\dots d+1,d+2-\tau} \ . \tag{122}$$

Further, we have:

$$\boldsymbol{\rho}_{d-\tau+2\dots d+1,d-\tau+2\dots d+1} = \begin{pmatrix} 1 & \boldsymbol{\rho}^T_{d+3-\tau\dots d+1,d+2-\tau} \\ \boldsymbol{\rho}_{d+3-\tau\dots d+1,d+2-\tau} & \boldsymbol{\rho}_{d+3-\tau\dots d+1,d+3-\tau\dots d+1} \end{pmatrix} = \begin{pmatrix} 1 & \boldsymbol{E}^T \\ \boldsymbol{E} & \boldsymbol{F} \end{pmatrix} , \tag{123}$$

so that:

$$\boldsymbol{\rho}^{-1}_{d-\tau+2\ldots d+1,d-\tau+2\ldots d+1} = \begin{pmatrix} \frac{1}{m} & -\frac{1}{m}\boldsymbol{E}^T\boldsymbol{E}^{-1} \\ -\frac{1}{m}\boldsymbol{F}^{-1}\boldsymbol{E} & \boldsymbol{F}^{-1} + \frac{1}{m}\boldsymbol{F}^{-1}\boldsymbol{E}\boldsymbol{E}^T\boldsymbol{F}^{-1} \end{pmatrix} \;, \tag{124}$$

with $m = 1 - \boldsymbol{E}^T\boldsymbol{F}^{-1}\boldsymbol{E}$. Now we know from (118) that is has to be:

$$\boldsymbol{\beta}^{(j^*)}_{(\tau-1)} = \begin{pmatrix} \boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ 1\ldots d-\tau} \\ \boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ d+1-\tau\ldots d} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ d+1-\tau\ldots d} \end{pmatrix} \;, \tag{125}$$

and we know that then the remaining sub-vector $\boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ d+1-\tau\ldots d}$ is:

$$\boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ d+1-\tau\ldots d} = \begin{pmatrix} \beta^{(j^*)}_{(\tau-1)\ d+1-\tau} \\ \boldsymbol{\beta}^{(j^*)}_{(\tau-1)\ d+2-\tau\ldots d} \end{pmatrix} \overset{(119)}{=} \boldsymbol{\rho}^{-1}_{d-\tau+2\ldots d+1,d-\tau+2\ldots d+1}\boldsymbol{\rho}_{d-\tau+2\ldots d+1,j^*} \tag{126}$$

$$\overset{(124)}{=} \begin{pmatrix} \frac{1}{m} & -\frac{1}{m}\boldsymbol{E}^T\boldsymbol{E}^{-1} \\ -\frac{1}{m}\boldsymbol{F}^{-1}\boldsymbol{E} & \boldsymbol{F}^{-1} + \frac{1}{m}\boldsymbol{F}^{-1}\boldsymbol{E}\boldsymbol{E}^T\boldsymbol{F}^{-1} \end{pmatrix} \begin{pmatrix} \rho_{j^*,d+2-\tau} \\ \boldsymbol{\rho}_{d+3-\tau\ldots d+1,j^*} \end{pmatrix} \tag{127}$$

$$= \begin{pmatrix} \frac{1}{m}(\rho_{j^*,d+2-\tau} - \boldsymbol{\rho}^T_{d+3-\tau\ldots d+1,j^*}\boldsymbol{\rho}^{-1}_{d+3-\tau\ldots d+1,d+3-\tau\ldots d+1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,d+2-\tau}) \\ \boldsymbol{F}^{-1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,j^*} + \frac{1}{m}\boldsymbol{F}^{-1}\boldsymbol{E}(\rho_{j^*,d+2-\tau} - \boldsymbol{\rho}^T_{d+3-\tau\ldots d+1,j^*}\boldsymbol{\rho}^{-1}_{d+3-\tau\ldots d+1,d+3-\tau\ldots d+1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,d+2-\tau}) \end{pmatrix} \tag{128}$$

$$\overset{(122)}{=} \begin{pmatrix} 0 \\ \boldsymbol{\rho}^{-1}_{d+3-\tau\ldots d+1,d+3-\tau\ldots d+1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,j^*} \end{pmatrix} \;, \tag{129}$$

where due to symmetry of the covariance matrix $\rho_{j^*,d+2-\tau} = \rho_{d+2-\tau,j^*}$. Finally, we have:

$$(\sigma^{(j^*)}_{(\tau-1)})^2 \overset{(120)}{=} 1 - \boldsymbol{\rho}^T_{d-\tau+2\ldots d+1,j^*}\boldsymbol{\rho}^{-1}_{d-\tau+2\ldots d+1,d-\tau+2\ldots d+1}\boldsymbol{\rho}_{d-\tau+2\ldots d+1,j^*} \tag{130}$$

$$\overset{(129)}{=} 1 - \begin{pmatrix} \rho_{d+2-\tau,j^*} & \boldsymbol{\rho}^T_{d+3-\tau\ldots d+1,j^*} \end{pmatrix} \begin{pmatrix} 0 \\ \boldsymbol{\rho}^{-1}_{d+3-\tau\ldots d+1,d+3-\tau\ldots d+1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,j^*} \end{pmatrix} \tag{131}$$

$$= 1 - \boldsymbol{\rho}^T_{d+3-\tau\ldots d+1,j^*}\boldsymbol{\rho}^{-1}_{d+3-\tau\ldots d+1,d+3-\tau\ldots d+1}\boldsymbol{\rho}_{d+3-\tau\ldots d+1,j^*} \;. \tag{132}$$

□

Proof of Theorem 2.5. We know from Theorem 2.4 that when the C-vine is truncated at level:

$$\tau \leq d + 1 - |K| - |S| \leq d + 1 - j^* \;, \tag{133}$$

we have:

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\ 1\ldots d-\tau} \overset{(118)}{=} \boldsymbol{0} \;, \tag{134}$$

$$\boldsymbol{\beta}^{(j^*)}_{(\tau)\ d+1-\tau\ldots d} \overset{(119)}{=} \boldsymbol{\rho}^{-1}_{d-\tau+2\ldots d,d-\tau+2\ldots d}\boldsymbol{\rho}_{d-\tau+2\ldots d+1,j^*} \;. \tag{135}$$

Further, since $\rho_{kl} = 0$, $\forall(k,l)$ with $k \in (K \cup S)$ and $l \in [d+1] \setminus (K \cup S)$, then $\boldsymbol{\rho}_{d-\tau+2\ldots d+1,j^*} = \boldsymbol{0}$, and it follows directly that $\boldsymbol{\beta}^{(j^*)}_{(\tau)} = \boldsymbol{0}$. □

I Statistical Discrepancy

Alaa et al. (2022) introduce α-precision, β-recall and authenticity (P_α, R_β, A), a three-dimensional, domain- and model-agnostic measure to evaluate fidelity, diversity and generalization of generative models on the sample level. Precision and recall for comparing two distributions were introduced in Sajjadi et al. (2018), and measure the degree of overlap of the supports of two distributions. On the contrary, α-precision and β-recall only give high scores if typical regions of the support of the two distributions (in our case: real and synthetic one), holding a certain probability mass, overlap. By this, α-precision and β-recall are able to diagnose different types of failures of the generative distribution, such as mode invention, mode drop or density shift. Hence, they give a more nuanced picture of the performance of a generative model.

For some $\alpha \in [0, 1]$ the α-support of the distribution P is defined as the minimum volume subset of $A \subset supp(P)$ that supports a probability mass of α (Alaa et al., 2022), i.e.:

$$\mathcal{S}^\alpha := \arg\min_{A \subset supp(P)} V(A) \quad s.th. \quad P(A) = \alpha \ , \tag{136}$$

where $V(A)$ is the volume (Lebesgue measure) of A. Thus, the α-precision and β-recall are given by:

$$P_\alpha := P(Z \in \mathcal{S}_R^\alpha) \ , \tag{137}$$

and:

$$R_\beta := P(X \in \mathcal{S}_S^\beta) \ , \tag{138}$$

respectively, with $\mathcal{S}_R^\alpha$ the α-support of the real distribution P_R and $\mathcal{S}_S^\beta$ the β-support of the generative distribution P_S and $\alpha, \beta \in [0, 1]$. For finding $\mathcal{S}_R^\alpha$ and $\mathcal{S}_S^\beta$ and evaluating P_α and R_β on data, Alaa et al. (2022) embed X and Z with an evaluation embedding. Letting α and β go from 0 to 1 we obtain curves for P_α and R_β. Alaa et al. (2022) show that $P_\alpha/\alpha = R_\beta/\beta = 1$ for all $\alpha, \beta \in [0, 1]$ if and only if $P_S = P_R$. Therefore it makes sense to define the integrated α-precision and integrated β-recall:

$$IP_\alpha := 1 - 2 \cdot \int_0^1 |P_\alpha - \alpha| d\alpha \ , \tag{139}$$

$$IR_\beta := 1 - 2 \cdot \int_0^1 |R_\beta - \beta| d\beta \ , \tag{140}$$

$$\tag{141}$$

both in $[0, 1]$, where values closer to 1 indicate a better generative model.

The authenticity score A measures to which percentage the generative model invents genuinely new samples rather than just copying real samples with some noise added. Consequently:

$$P_S = A \cdot P_S' + (1 - A) \cdot \delta_{S,\epsilon} \ , \tag{142}$$

where P_S' is the generative distribution conditioned on the synthetic samples not being copied. In the second summand $\delta_{S,\epsilon} = \delta_S * \mathcal{N}(0, \epsilon^2)$ is a convolution of the discrete distribution δ_S placing an unknown probability mass on each real sample in X and the noise distribution $\mathcal{N}(0, \epsilon^2)$ with arbitrarily small noise variance ϵ.

Alaa et al. (2022) estimate α-precision (β-recall) of a single synthetic (real) sample to be 1 if it resides within the estimate of $\mathcal{S}_R^\alpha$ ($\mathcal{S}_S^\beta$) and 0 otherwise. The mean of all sample-wise P_α (R_β) scores gives the α-precision (β-recall) of the synthetic (real) data set. The authenticity score of a synthetic sample is estimated through a likelihood ratio test and averaged to obtain the authenticity of the whole synthetic data set. In our analysis we estimate (P_α, R_β, A) in terms of the unlabeled real data X and unlabeled synthetic data Z.

J Model and Attack Parameters

Parameters of the AIA and MIA are given in Table 1, parameters of the generative models are given in the following.

Table 1: Parameters of AIAs and MIAs.

Attack	Parameter	Value
AIA	no. game iterations N (nIter):	10
AIA	size of reference data sizeRawT:	500
AIA	size of synthetic data sizeSynT:	500
AIA	no. bootstraped/synthetic data sets n_{synth} (nSynT):	50
AIA	size of bootstrap samples bootstrapSize:	500
MIA	no. game iterations N (nIter):	10
MIA	size of real reference data set for attacker's training sizeRawA:	500
MIA	no. of shadow models during attacker's training nShadows:	10
MIA	no. synthetic data sets sampled during attacker's training nSynA:	10
MIA	size of real reference data set for attacker's evaluation sizeRawT:	400
MIA	size of the synthetic data set generated during attacker's evaluation sizeSynT:	400
MIA	no. synthetic data sets evaluated nSynT:	50

Parameters of the generative models:

- **Vine Copula:** Parametric pair copula families and their rotations are estimated with maximum likelihood and selected with AIC as selection criterion.
- **PrivBayes:** Histogram bins 25 and degree 1. Privacy parameter $\epsilon \in \{0.1, 1, 5\}$.
- **CTGAN:** Number of epochs and batch size were tuned with random search to 1000 and 150 respectively for results on simulated real data in Section 3 and Appendix L. The remaining parameters are set to default values as provided in the CTGAN library implementing Xu et al. (2019). For results on SUPPORT2 data in Section 3.1 the random search resulted in 400 epochs and a batch size of 100.
- **TVAE:** Number of epochs, batch size and the dimension of the latent space were tuned with random search to 1500, 400 and 2 respectively for results on simulated real data in Section 3 and Appendix L. The remaining parameters are set to default values as provided in the CTGAN library implementing Xu et al. (2019). For results on SUPPORT2 data in Section 3.1 the random search resulted in 800 epochs, a batch size of 100 and latent space dimension of 4.
- **PrivPGD:** Parameters are kept to their default values, as Donhauser et al. (2024) state that PrivPGD does not require specific parameter tuning due to the data being represented as particles. We choose the authors' proposed DP parameter default values, i.e. $\epsilon = 2.5$ and $\delta = 10^{-5}$.

K Compute Resources

All experiments on the SUPPORT2 data with results in Section 3.1 and Appendix M and utility and statistical fidelity results on simulated real data of Section 3 and Appendix L were conducted on an Apple Macbook Pro with macOS Sonoma 14.4.1, Apple M2 Pro chip and 16 GB RAM using 10 cores. AIA and MIA experiments on simulated real data of Section 3 and Appendix L were conducted on an hpc cluster with the following specs:

- CPU: 256 threads (2 × AMD EPYC 7713 Milan: 64 cores, 128 threads per CPU)
- RAM: 4 TB (32 × 128 GB DDR4)
- OS: Linux (Red Hat Enterprise Linux 7)

Experiments were conducted in parallel on 20 cores.

Software used for experiments on Apple Macbook Pro:

- Python 3.10.13
- R version 4.3.1 (2023-06-16) – "Beagle Scouts"

- tmux 3.3a
- conda 23.10.0

Execution time measured with *time* command of AIA on SUPPORT2 data for C-vine per truncation level:

- truncation at level 1: 489.20s user 14.44s system 100% cpu 8:21.01 total
- truncation at level 5: 1168.23s user 14.72s system 99% cpu 19:44.55 total
- truncation at level 10: 1836.25s user 14.78s system 99% cpu 30:56.40 total
- truncation at level 15: 2427.97s user 15.03s system 99% cpu 40:49.45 total
- truncation at level 20: 2877.82s user 15.27s system 99% cpu 48:22.73 total
- no truncation: 3059.69s user 15.21s system 99% cpu 51:26.67 total

Execution time measured with *time* command of MIA on SUPPORT2 data for C-vine per truncation level:

- truncation at level 1: 1540.75s user 84.16s system 63% cpu 42:28.99 total
- truncation at level 5: 3180.88s user 89.65s system 103% cpu 52:51.39 total
- truncation at level 10: 4783.33s user 92.23s system 128% cpu 1:03:04.40 total
- truncation at level 15: 6077.78s user 93.04s system 143% cpu 1:11:34.51 total
- truncation at level 20: 6994.86s user 91.37s system 151% cpu 1:18:11.24 total
- no truncation: 7385.50s user 90.99s system 154% cpu 1:20:47.09 total

L Simulated Real Data

We simulate real data with $n = 1000$ and $n_{test} = 250$ realizations of the random vector $(X_1, X_2, \dots, X_{20}, Y) \in \mathbb{R}^{20} \times \{0, 1\}$ following a distribution F. The joint distribution F of $(\boldsymbol{X}^T, Y) := (X_1, X_2, \dots, X_{20}, Y)$ is composed the following way: $Y \sim \mathit{Bernoulli}(0.5)$, $\boldsymbol{X}|Y = 0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$ and $\boldsymbol{X}|Y = 1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$ with $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \Sigma_0$ and Σ_1 defined in Equations (143), (144), (145) and (146). As can be observed from parameters of F, the dependence structure of F is of block form where the three blocks $(X_1, \dots, X_5)$, $(X_6, \dots, X_{10})$ and $(X_{11}, \dots, X_{20}, Y)$ are independent. The distribution F was chosen deliberately such that, if we simulate data, we obtain three approximately uncorrelated blocks in the correlation matrix of the real data to investigate the effect of truncation according to Theorem 2.5. For this reason we do not apply Algorithm 1 additionally.

The estimated correlation matrix of the real data shown Figure 8 exhibits this block structure to imitate a scenario where some covariates (third block: $X_{11}, ..., X_{20}$) are important for classifying Y while others (first block: $X_1, ..., X_5$ and second block: $X_6, ..., X_{10}$) are less so. Let us assume that covariates X_1, X_6 and X_{11} are sensitive. In this experiment we know that the dependencies in the first and second block do not contribute to the classification of Y but provide information on the sensitive covariates X_1 and X_6 which may result in impaired privacy. Hence, in TVineSynth we choose the structure and truncation level of the vine copula such that these dependencies are not reflected in the synthetic data. Specifically, we use an ordering $\mathcal{O}^*$ of the covariates that corresponds to their indices, i.e. $X_j = X_{(j)}$, and Y as the center of T_1, see Appendix L.1. In Figure 9 we can observe how the correlation structure of the real data in Figure 8 is more and more reproduced in the synthetic data generated by a C-vine with increasing truncation level. Specifically, we note that dependencies in the first block containing sensitive covariate X_1 start to be represented in synthetic data from truncation level 17 and more closely from truncation level 19 onward. For X_6 in the second block this is the case from truncation level 12 onward. In the third block containing sensitive covariate X_{11} this happens already from truncation level 1 onward. This indicates the effect of truncation combined with the C-vine structure.

$$\boldsymbol{\mu}_0^{(I)} := (-2.42, 5.84, 20.10, 12.66, 0.35, 12.64, 12.29, 21.29, 1.11, 24.69, 25.27, -3.53, 6.10, -4.52, 3.37, 19.73, 5.78, 12.80, -3.19, 14.76)^T \;, \tag{143}$$

$$\boldsymbol{\mu}_1^{(I)} := (-2.42, 5.84, 20.10, 12.66, 0.35, 12.64, 12.29, 21.29, 1.11, 24.69, 24.44, -4.78, 6.51, -4.73, 2.08, 20.63, 5.26, 13.57, -2.94, 15.39)^T \;, \tag{144}$$

$$\Sigma_0^{(I)} = \left(\begin{array}{cccccccccccccccccccc}
2.57 & -2.14 & 1.33 & 0.04 & -0.76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2.14 & 6.12 & -2.99 & -0.36 & 1.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.33 & -2.99 & 6.36 & -1.85 & 2.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.04 & -0.36 & -1.85 & 3.29 & -0.97 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.76 & 1.25 & 2.05 & -0.97 & 6.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3.80 & -1.97 & 1.69 & -0.29 & -1.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1.97 & 7.77 & -1.69 & -2.07 & 2.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.69 & -1.69 & 3.86 & 1.67 & -1.46 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.29 & -2.07 & 1.67 & 4.12 & -1.47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1.01 & 2.00 & -1.46 & -1.47 & 1.92 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.82 & -1.29 & -2.52 & 1.80 & -1.93 & -2.13 & -2.74 & -1.84 & -0.09 & -2.72 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.29 & 8.60 & -0.98 & -3.48 & -1.80 & -1.33 & 2.56 & -2.21 & -1.09 & -0.62 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.52 & -0.98 & 5.44 & 0.48 & -1.02 & 0.63 & -1.18 & 1.90 & -1.13 & 2.84 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.80 & -3.48 & 0.48 & 5.67 & -1.13 & -1.80 & -2.98 & 0.89 & 0.28 & -0.37 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.93 & -1.80 & -1.02 & -1.13 & 7.80 & 2.92 & 3.83 & 3.01 & 1.11 & 2.81 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.13 & -1.33 & 0.63 & -1.80 & 2.92 & 4.44 & 3.05 & 2.73 & -0.03 & 1.93 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.74 & 2.56 & -1.18 & -2.98 & 3.83 & 3.05 & 7.16 & 2.72 & 2.21 & 2.05 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.84 & -2.21 & 1.90 & 0.89 & 3.01 & 2.73 & 2.72 & 5.21 & 1.16 & 3.65 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.09 & -1.09 & -1.13 & 0.28 & 1.11 & -0.03 & 2.21 & 1.16 & 4.99 & 0.23 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.72 & -0.62 & 2.84 & -0.37 & 2.81 & 1.93 & 2.05 & 3.65 & 0.23 & 5.88
\end{array}\right), \tag{145}$$

$$\Sigma_1^{(I)} = \left(\begin{array}{cccccccccccccccccccc}
2.57 & -2.14 & 1.33 & 0.04 & -0.76 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2.14 & 6.12 & -2.99 & -0.36 & 1.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.33 & -2.99 & 6.36 & -1.85 & 2.05 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.04 & -0.36 & -1.85 & 3.29 & -0.97 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.76 & 1.25 & 2.05 & -0.97 & 6.07 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3.80 & -1.97 & 1.69 & -0.29 & -1.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1.97 & 7.77 & -1.69 & -2.07 & 2.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.69 & -1.69 & 3.86 & 1.67 & -1.46 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.29 & -2.07 & 1.67 & 4.12 & -1.47 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1.01 & 2.00 & -1.46 & -1.47 & 1.92 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6.01 & -3.24 & 2.67 & -0.55 & 3.89 & 1.34 & 1.22 & -1.22 & 2.02 & -2.53 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.24 & 7.78 & -0.19 & 2.07 & -3.66 & 0.89 & -0.01 & 0.03 & -0.35 & -0.28 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.67 & -0.19 & 6.12 & 0.90 & 3.53 & 1.65 & -0 & -1.67 & 3.01 & -1.95 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.55 & 2.07 & 0.90 & 5.60 & 1.52 & 2.00 & 1.41 & 1.80 & 1.06 & -1.91 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.89 & -3.66 & 3.53 & 1.52 & 8.20 & 1.75 & 1.01 & -0.11 & 2.97 & -1.88 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.34 & 0.89 & 1.65 & 2.00 & 1.75 & 3.06 & 1.48 & -0.59 & 1.15 & -0.50 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.22 & -0.01 & -0 & 1.41 & 1.01 & 1.48 & 5.03 & -1.08 & 1.93 & -1.31 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.22 & 0.03 & -1.67 & 1.80 & -0.11 & -0.59 & -1.08 & 4.02 & -1.14 & 0.08 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.02 & -0.35 & 3.01 & 1.06 & 2.97 & 1.15 & 1.93 & -1.14 & 5.13 & -1.57 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.53 & -0.28 & -1.95 & -1.91 & -1.88 & -0.50 & -1.31 & 0.08 & -1.57 & 5.40
\end{array}\right). \tag{146}$$

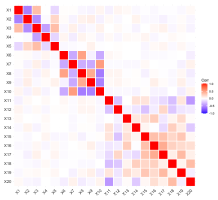
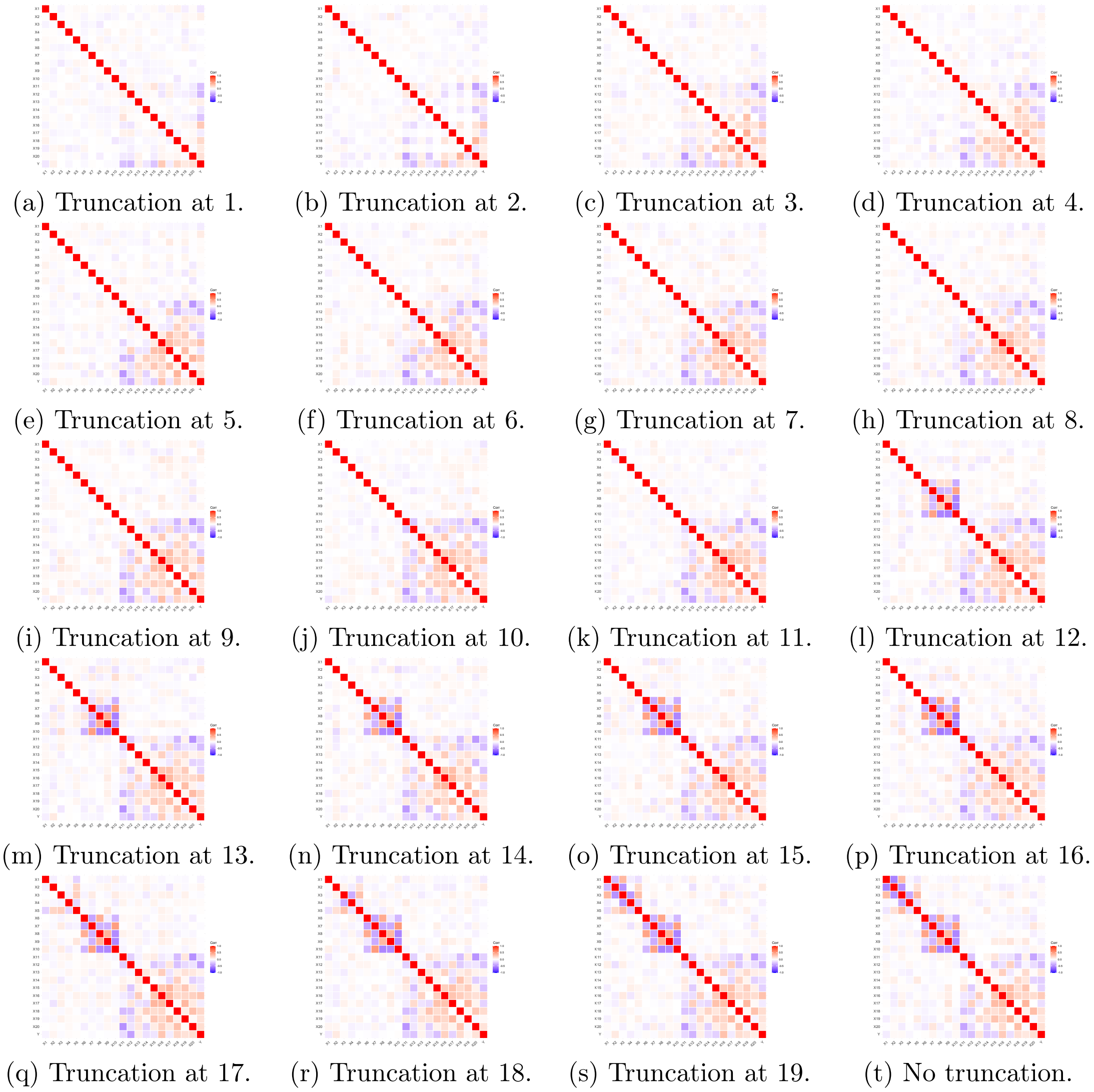


Figure 8: Pearson correlation matrix estimated on simulated real data. It exhibits a block structure to imitate a scenario where some covariates, i.e. the ones in the third block $X_{11}, ..., X_{20}$ are important for classification, while others in the first block, $X_1, ..., X_5$ and in the second block $X_6, ..., X_{10}$ are not.



(a) Truncation at 1. (b) Truncation at 2. (c) Truncation at 3. (d) Truncation at 4.

(e) Truncation at 5. (f) Truncation at 6. (g) Truncation at 7. (h) Truncation at 8.

(i) Truncation at 9. (j) Truncation at 10. (k) Truncation at 11. (l) Truncation at 12.

(m) Truncation at 13. (n) Truncation at 14. (o) Truncation at 15. (p) Truncation at 16.

(q) Truncation at 17. (r) Truncation at 18. (s) Truncation at 19. (t) No truncation.

Figure 9: The (Pearson) correlation matrices of the synthetic data generated with a C-vine for truncation levels from 1 to 19 and no truncation illustrate how the correlation structure of the real data is more and more reproduced with increasing truncation level. Note that correlations in the first and second block are only reflected in the synthetic data from truncation level 12 and 17 respectively. Details on the estimation of the C-vine can be found in Section 2.2 and Appendix J.

L.1 R-Vine Matrix of the C-Vine Used as a Generative Model on Simulated Real Data

R-vine matrices are a compact way to represent the vine tree structure $\mathcal{V}$. They indicate which pairwise conditional dependencies between covariates are modeled through an edge in a tree in $\mathcal{V}$. For a thorough introduction, the reader can consult for example Czado (2019). The R-vine matrix of the C-vine used as a generative model on simulated real data is as follows with Y on index 21 and index $j \in [20]$ corresponding to covariate X_j:

$$\begin{pmatrix} 21 & 21 & 21 & \cdots & 21 & 21 \\ & 20 & 20 & \cdots & 20 & 20 \\ & & 19 & \cdots & 19 & 19 \\ & & & \ddots & \vdots & \vdots \\ & & & & 2 & 2 \\ & & & & & 1 \end{pmatrix} . \tag{147}$$

L.2 Choice of Target Observations for Privacy Evaluation on Simulated Real Data

We conduct an AIA and MIA on simulated real data described in L. The parameter setup of the privacy attacks can be found in Table 1. For the attacks for each sensitive covariate four target observations are handpicked outside the 95%-quantile of the regarding sensitive covariate, see Table 2.

Additionally, five target observations, namely ID123, ID507, ID589, ID740 and ID922[13] are randomly sampled from the real data set. They correspond to the quantiles w.r.t. the respective sensitive covariate given in Table 3.

[13] NB: IDk corresponds to the $(k+1)$th observation in the real data set with $k \in \{0, ..., (n-1)\}$.

Table 2: Target observations of the simulated real data set, Section L that are handpicked to lie outside the 95%-quantile of the respective sensitive covariate X_1, X_6 and X_{11}.

	QUANTILES			
SENSITIVE COVARIATE	0.01	0.025	0.975	0.99
X_1	ID202	ID164	ID179	ID843
X_6	ID127	ID4	ID353	ID326
X_{11}	ID970	ID949	ID392	ID862

Table 3: Randomly sampled target observations from the simulated real data set of Section L and their corresponding quantiles w.r.t. covariates X_1, X_6 and X_{11}.

	TARGET IDs				
SENSITIVE COVARIATE	ID123	ID507	ID589	ID740	ID922
X_1	0.475	0.858	0.284	0.469	0.302
X_6	0.517	0.473	0.945	0.512	0.309
X_{11}	0.628	0.592	0.838	0.204	0.549

L.3 Simulated Real Data: Results

Parameters of the privacy attacks and of the generative models can be found in Appendix J. As outlined in Section 2.5, we pick four target observations outside the 95% quantile for each sensitive covariate X_1, X_6 and X_{11} and randomly sample five more target observations from the real data for the privacy analysis, see Appendix L.2.

L.3.1 Privacy: Attribute Inference Attack

The top row of Figure 10 corresponds to the case where the sensitive covariate X_1 is less important for classifying Y correctly. In this situation the star shaped C-vine combined with truncation at level 18 or lower is able to cut away sensitive dependencies that harm privacy but do not contribute to utility. This is in accordance with our observations from Figures 8 and 9. If the sensitive covariate is X_6, again playing a less important role for classifying Y correctly, we observe in the second row of Figure 10 that truncating a C-vine at level 11 or lower offers a high level of privacy, which again complies with our observations from Figures 8 and 9. Thus, the C-vine offers a high level privacy w.r.t. AIA, which is comparable to the one of the DP PrivBayes model at a very strict privacy budget of $\epsilon = 0.1$, and outperforms CTGAN, TVAE and PrivPGD in terms of AIA privacy with some margin for low truncation levels. Simultaneously, the C-vine achieves high utility for all truncation levels, outperforming PrivBayes by far, see Figure 14. Sensitive covariate X_{11} on the other hand shows pairwise association with Y, see Figure 8. In this case it is necessary to truncate the C-vine at level 1 to provide privacy w.r.t. AIA, see bottom row of Figure 10. The AIA results in terms of WCAB in Appendix L.3.6 and in terms of the MSE in Appendix L.3.7 confirm these findings.

As a proof of concept for why we base TVineSynth on a C-vine we generate synthetic data with an R-vine where the vine tree structure is not pre-specified, but selected as described in (Dissmann et al., 2013), and an R-vine star1 model and compare it to C-vine generated synthetic data. An R-vine star1 model is equal to an R-vine except that we exchange its first tree with T_1 of the C-vine. Even for truncation at a very low level, an R-vine struggles to offer effective protection against AIAs. The same holds for an R-vine star1 for truncation level 2 and higher. If it consists only of its star-shaped first tree, an R-vine star1 is equivalent to a C-vine and thus grants the same high level of privacy, see Figure 11.

L.3.2 Privacy: Membership Inference Attack

In Figure 12 we observe that the PG of C-vine generated synthetic data is around 1 with low variation for *all* truncation levels, indicating optimal privacy w.r.t MIA, independent of whether the target observation is randomly sampled (in blue) or an outlier (in orange).

The PG of the C-vine is seemingly independent of truncation level because the estimation of the un-truncated

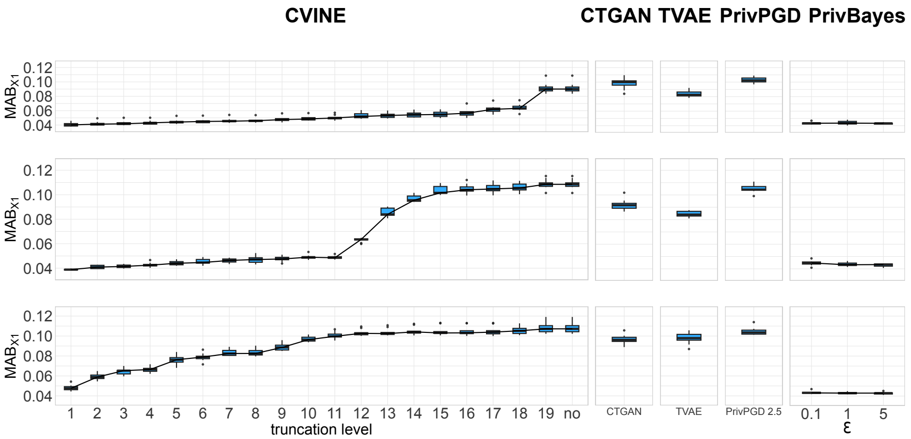


Figure 10: Simulated real data: Results of an AIA w.r.t. sensitive covariate X_1 (top row), X_6 (middle row) and X_{11} (bottom row) measured by MAB_j. Synthetic data are generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes (right) for privacy parameter $\epsilon \in \{0.1, 1, 5\}$. Results are reported as box plots over 10 AIA game iterations. Parameters of the generative models and privacy attacks can be found in Appendix J.

C-vine done with Maximum Likelihood is robust w.r.t. adding/removing a single observation to the real data. The robustness of Maximum Likelihood estimation (MLE) depends on the sample size of the (real) data. Thus, for lower sample sizes than the ones we use here, we would expect to see a MIA PG that varies more with truncation level. As a consequence also a C-vine truncated at level $t < d$ shows the same robustness, because it results from the un-truncated C-vine by setting pair copulas in tree levels $t + 1$ and higher to independence.

The results of the C-vine are similar to PrivBayes model for privacy parameter $\epsilon \in \{0.1, 1, 5\}$. CTGAN also gives average PG of about 1, but exhibits a high variation in the PG over different observations and repetitions of the MIA. Synthetic data generated with a TVAE provide very low protection against MIAs with a PG of around 0, hinting on that they include too much details of the real data which are harmful for privacy. The PrivPGD model performs poorly in terms of PG. Even though the covariate ranges are not directly inferred from the sensitive real data, this might have an impact on MIA privacy.

Similar to the C-vine, the R-vine and R-vine star1 score a PG of 1 at median with little variation over different observations and repetitions of the attack, see Figure 13.

L.3.3 Utility

For evaluating utility, 50 synthetic data sets of the same size as the simulated real data ($n = 1000$) are generated from each model, a random forest classifier is trained on each of them and tested on a hold-out test data set of size $n_{test} = 250$. From Figure 14 we observe that the C-vine generated synthetic data consistently outperform synthetic data generated from a CTGAN, PrivPGD and a PrivBayes model for all truncation levels. Only the TVAE scores a higher $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ and comes closest to the performance of the classifier trained on real data of $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{y}}^*) = 0.908$. Considering its high MAB under an AIA in Figure 10 and low PG under a MIA in Figure 12, the TVAE seems to model the real data too closely, thus violating privacy.

R-vine and R-vine star 1 generated synthetic data are as useful as C-vine generated synthetic data, see Figure 15. As they perform worse in terms of AIA privacy, we see ourselves confirmed in our choice of a C-vine as the core of TVineSynth.

L.3.4 Privacy-Utility Plots

Figures 1, 16 and 17 illustrate the privacy-utility balance per model. In the case of sensitive feature X_1 in Figure 16, the C-vine offers a well balanced privacy-utility trade-off for truncation between levels 11 and 18. If we truncate at level 11 for sensitive covariate X_6 and at level 1 for sensitive covariate X_{11}, the TVineSynth generated synthetic data offer a privacy-utility balance superior to the one of the competitor models, see Figure 16. Figure

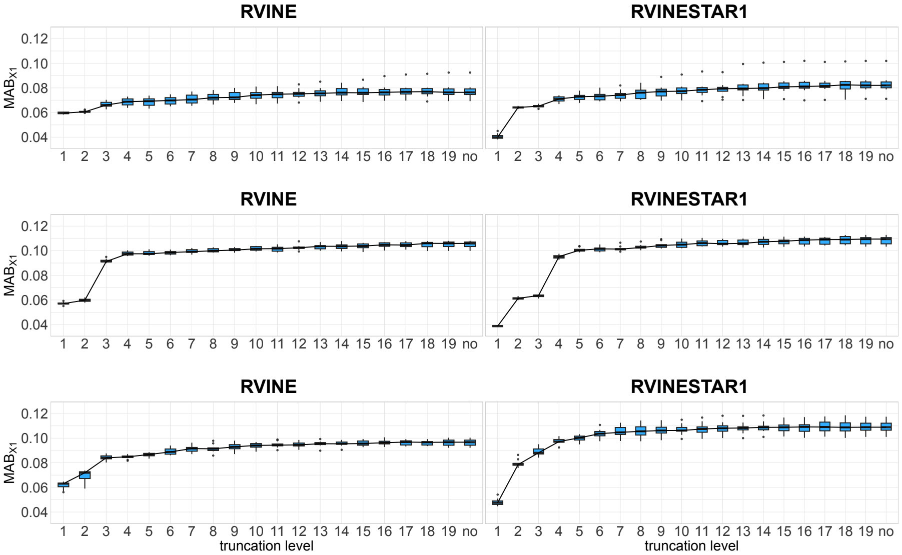


Figure 11: The lower the MAB_j of AIA w.r.t. sensitive covariate X_1, X_6 and X_{11}, the more private the synthetic data generated by a R-vine and R-vine star1 for different truncation levels. Results are reported as box plots over 10 AIA game iterations. Parameters of the generative models and privacy attacks can be found in Appendix J.

17 displays the privacy-utility plots w.r.t. a MIA and and sensitive features X_1, X_6 and X_{11}. As already observed in Sections L.3.2 and L.3.3, all models except for TVAE and PrivPGD score a PG of around 1. Compared to the competitor models that are able to protect sensitive covariates against a MIA, the C-vine scores the highest utility.

L.3.5 Statistical Discrepancy

We measure the statistical discrepancy between joint real and synthetic distribution with α-precision, β-recall and authenticity (P_α, R_β, A) introduced by Alaa et al. (2022). Their definition can be found in Appendix I.

From Figure 18 it can be observed that increasing truncation level of the C-vine improves fidelity and diversity of the synthetic data while it decreases their generalization. While PrivBayes generated synthetic data score very poorly in diversity (around 0) and moderately in fidelity (0.63 - 0.67), they achieve very high authenticity of around 1, indicating that the synthetic data do not reflect the real data sufficiently well. PrivPGD's diversity and authenticity (0.7 and 0.63) compare to the one of the C-vine truncated at level 1 (0.71 and 0.66), but achieves a fidelity of 0.76 that is considerably lower than the one of the C-vine truncated at level 1 (0.94). The TVAE performs very comparably to the C-vine truncated at level 10 in terms of statistical fidelity. The rather high generalization and rather low diversity of CTGAN generated data appear plausible w.r.t. the the model's results of Section L.3.3.

L.3.6 AIA Results in Terms of WCAB

Figure 19 displays the AIA results in terms of WCAB that give a worst-case assessment of how much information covariates in the synthetic data leak on the sensitive feature. They support and further strengthen the observations on the MAB of Figure 10. For sensitive feature X_1 truncating the C-vine at level 18 or lower providing a worst-case privacy superior to the differentially private PrivBayes. The same holds if the sensitive covariate is X_6 and we truncate the C-vine at level 11 or lower. Even for the sensitive feature X_{11}, which informs Y, the WCAB of the C-vine is comparable or lower than that of the competitors.

L.3.7 AIA Results in Terms of MSE

The top row of Figure 21 corresponds to the case where the sensitive covariate X_1 is less important for classifying Y correctly. In this situation the star shaped C-vine combined with truncation at level 18 or lower is able to cut away sensitive dependencies that harm privacy but do not contribute to utility. If the sensitive covariate is

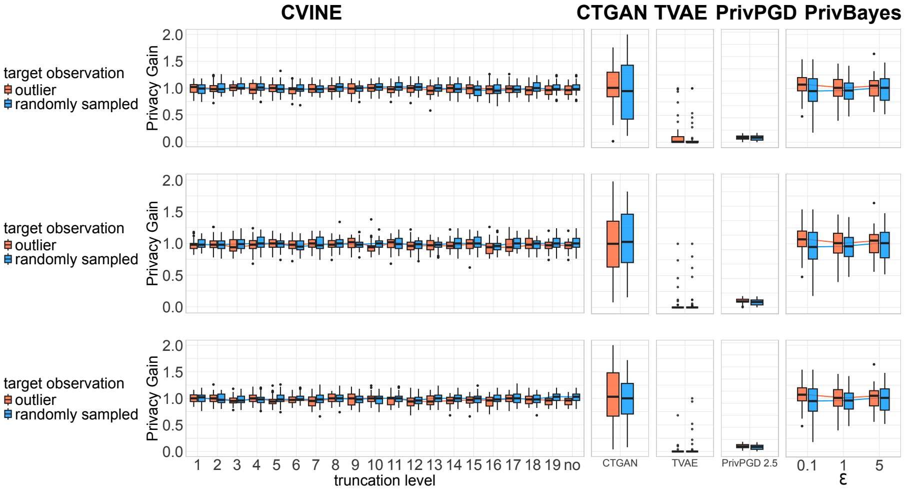


Figure 12: Simulated real data: PG under a MIA w.r.t randomly sampled target observations (in blue) and targets that are outliers (in orange) w.r.t. X_1 (top row), X_6 (middle row) and X_{11} (bottom row) of synthetic data are generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes for privacy parameter $\epsilon \in \{0.1, 1, 5\}$ (right). Results are reported as box plots over 10 MIA game iterations and 4 outlying (orange) and 5 randomly sampled (blue) target observations respectively. Parameters of the generative models and privacy attacks can be found in Appendix J.

X_6, again playing a less important role for classifying Y correctly, we observe in the second row of Figure 21 that truncating a C-vine at level 11 or lower offers a high level of privacy for outliers (in orange). These findings are consistent with our observations from the results in terms of the MAB in Figure 10 and the correlation structure in Figures 8 and 9. Thus, the C-vine offers a high level of privacy for outliers (in orange) w.r.t. AIA, which is comparable to the one of the DP PrivBayes model at a very strict privacy budget of $\epsilon = 0.1$, and better than the one of DP PrivPGD. Simultaneously, the C-vine achieves high utility for all truncation levels, outperforming PrivBayes and PrivPGD by far, see Figure 14. Sensitive covariate X_{11} on the other hand shows pairwise association with Y, see Figure 8. In this case it is necessary to truncate the C-vine at level 1 to provide privacy w.r.t. AIA, see bottom row of Figure 21.

From Figure 21 we observe that randomly sampled target observations (in blue) that are close to the median of the sensitive covariate show very low $MSE_S(x_{t,s}|s_t = 1)$. This raises the question of whether this actually presents a privacy breach. It does *not*, if it suffices for the attacker to merely guess the mean of the respective sensitive covariate without regarding the other non-sensitive covariates! In other words, if in the attacker's regression model, the coefficients of the respective non-sensitive covariates are (close to) 0, the synthetic data does not offer more information on the sensitive covariate value than what we really wish to learn from the synthetic data, i.e. aggregate information such as the mean of a covariate. For this reason, we assess the regression coefficients in the AIA model for C-vine generated synthetic data and sensitive covariates X_1, see Figure 22. There we indeed find that the results of Figure 21 do not present an impairment of privacy. For sensitive covariate X_1, the regression coefficients of non-sensitive covariates displayed in Figure 22 are at median 0 for all target observations up to truncation level 16, as we would expect from Figure 9. This means that even though $MSE_S(x_{t,s}|s_t = 1)$ is low in those cases, the attacker's guess is merely based on the mean of the respective sensitive covariate and guessing a covariate's mean correctly does not leak private information but confirms the synthetic data still allow to learn aggregate information about the real data as it is our goal.

These considerations build the basis for Definition 2.2 of the MAB.

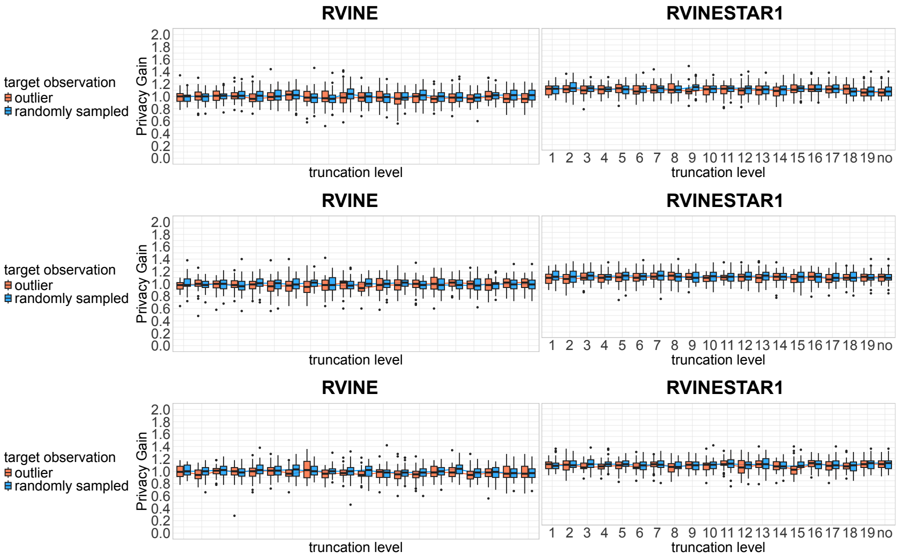


Figure 13: Simulated real data: Results of a MIA w.r.t randomly sampled target observations (in blue) and targets that are outliers (in orange) w.r.t. X_1 (top row), X_6 (middle row) and X_{11} (bottom row), measured by the PG. Synthetic data are generated with an R-vine (left) and an R-vine star1 (right) for different truncation levels. Results are reported as box plots over 10 MIA game iterations and 4 outlying (orange) and 5 randomly sampled (blue) target observations respectively Parameters of the generative models and privacy attacks can be found in Appendix J. Parameters of the generative models and privacy attacks can be found in Appendix J.

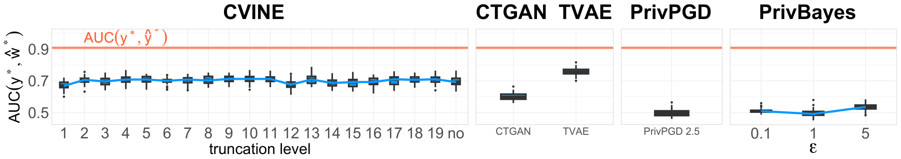


Figure 14: Simulated real data: Utility of synthetic data generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes for privacy parameter $\epsilon \in \{0.1, 1, 5\}$ (right) measured with $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ (blue) w.r.t. a random forest classifier and compared to $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{y}}^*)$ (orange). Results are reported as box plots over 50 AUC values obtained from 50 synthetic data sets per generative model. Parameters of the generative models can be found in Appendix J.

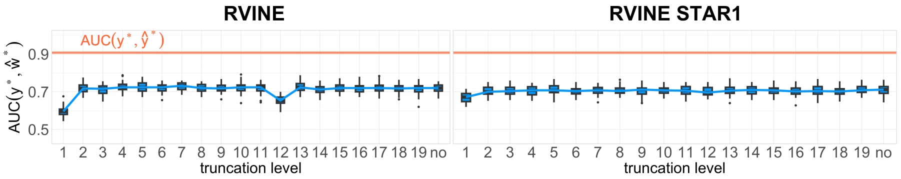


Figure 15: Simulated real data: Enforcing the first vine tree to be a star with Y as the center as for R-vine star1 yields higher utility at truncation level 1 than compared to R-vine. Utility is consistently high for all truncation levels. It is measured with $AUC(\boldsymbol{y}^*, \hat{\boldsymbol{w}}^*)$ w.r.t. a random forest classifier. Results are reported as box plots over 50 AUC values obtained from 50 synthetic data sets per generative model. Parameters of the generative models can be found in Appendix J.

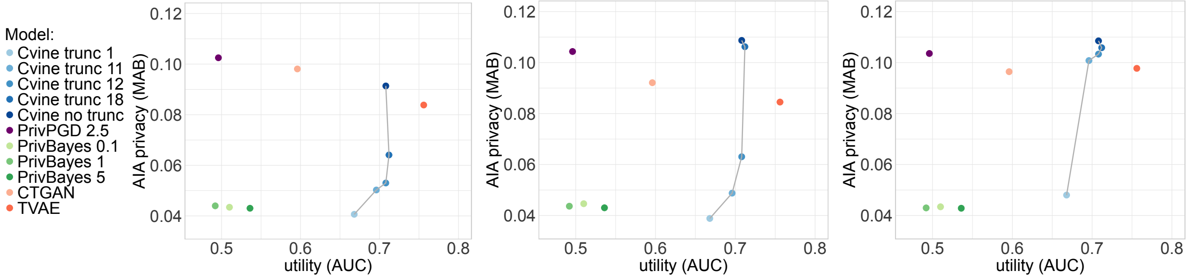


Figure 16: Simulated real data: Privacy-utility plot w.r.t. AIA and sensitive features X_1 (left), X_6 (middle) and X_{11} (right) of a C-vine with truncation levels $t \in \{1, 11, 12, 18\}$ and no truncation, PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$, PrivBayes model with $\epsilon \in \{0.1, 1, 5\}$, CTGAN and TVAE on simulated real data of Section 3. For AIA privacy the MAB_j is reported, for utility the median over 50 synthetic data sets is reported. Parameters of the generative models and privacy attacks can be found in Appendix J.

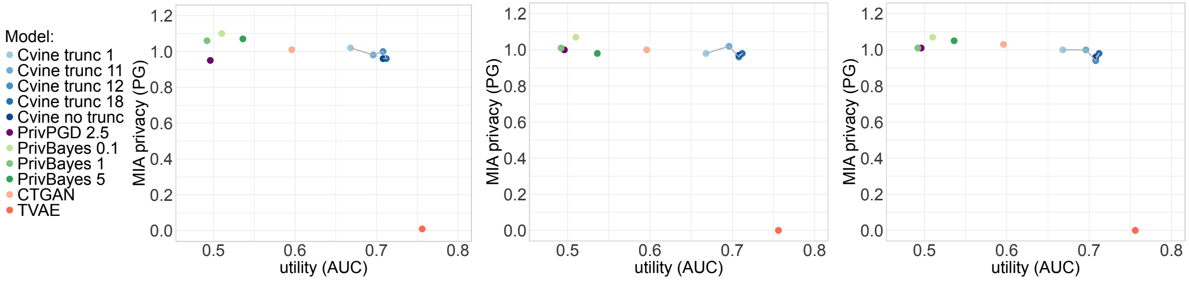


Figure 17: Simulated real data: Privacy-utility plot w.r.t. MIA and sensitive features X_1 (left), X_6 (middle) and X_{11} (right) of a C-vine with truncation levels $t \in \{1, 11, 12, 18\}$ and no truncation, PrivBayes model with $\epsilon \in \{0.1, 1, 5\}$, CTGAN and TVAE on simulated real data of Section L. The median MIA PG over all game iterations and utility for 50 synthetic data sets are reported. Parameters of the generative models and privacy attacks can be found in Appendix J.

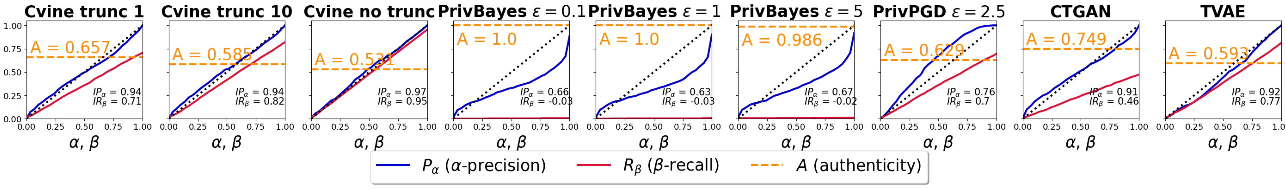


Figure 18: Simulated real data: Fidelity (α-precision), diversity (β-recall) and generalization (authenticity) of synthetic data generated in the order C-vine for truncation at levels 1 and 10 and no truncation, PrivBayes for $\epsilon \in \{0.1, 1, 5\}$, PrivPGD with $\epsilon = 2.5$, $\delta = 10^{-5}$, CTGAN and TVAE. Parameters of the generative models can be found in Appendix J.

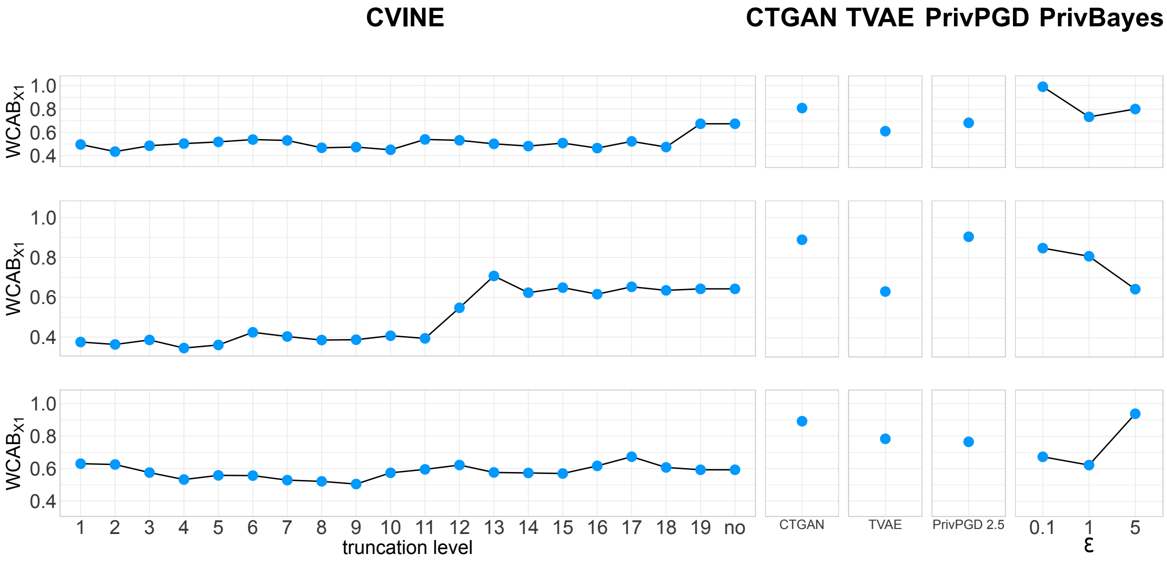


Figure 19: Simulated real data: Results of an AIA w.r.t. sensitive covariate X_1 (top row), X_6 (middle row) and X_{11} (bottom row) measured by $WCAB_{j^*}$. Synthetic data are generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes (right) for privacy parameter $\epsilon \in \{0.1, 1, 5\}$. Results are reported over 10 AIA game iterations. Parameters of the generative models and privacy attacks can be found in Appendix J.

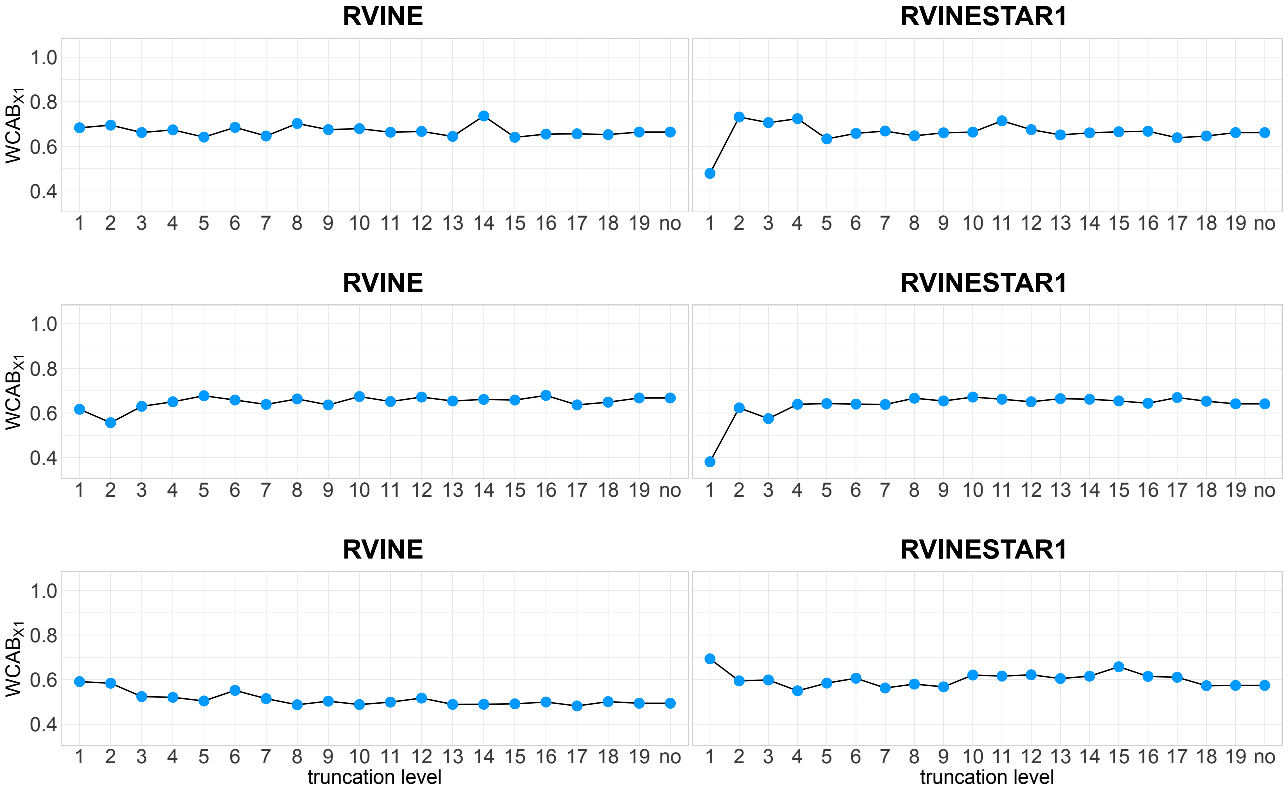


Figure 20: Simulated real data: The lower the $WCAB_{j^*}$ of AIA w.r.t. sensitive covariate X_1, X_6 and X_{11}, the more private the synthetic data generated by a R-vine and R-vine star1 for different truncation levels. Results are reported over 10 AIA game iterations. Parameters of the generative models and privacy attacks can be found in Appendix J.

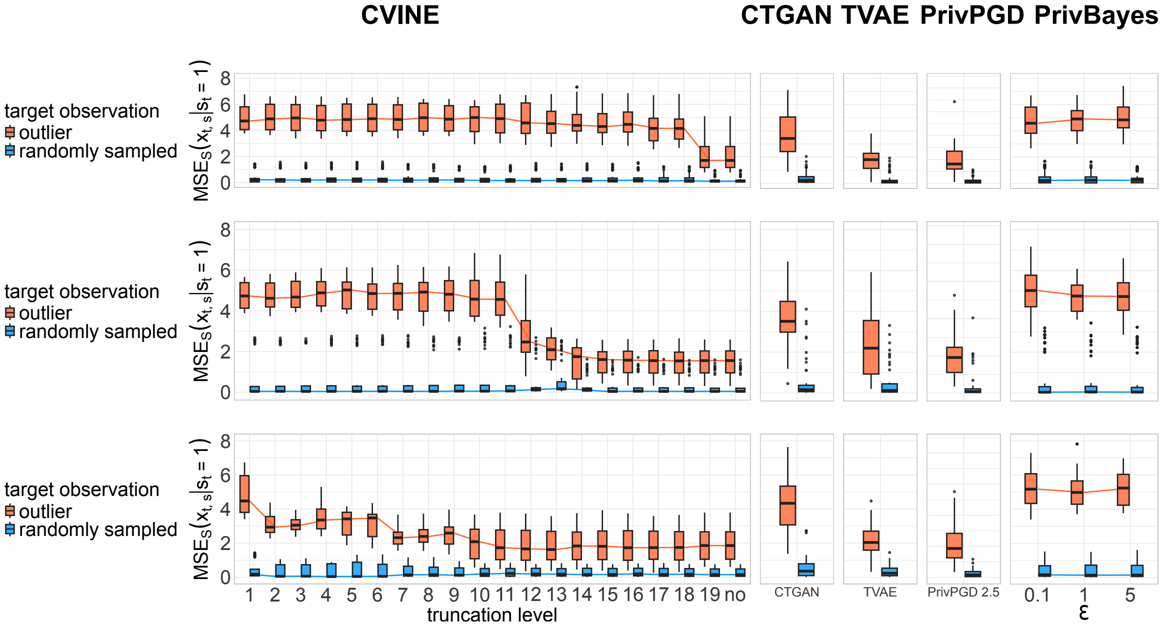
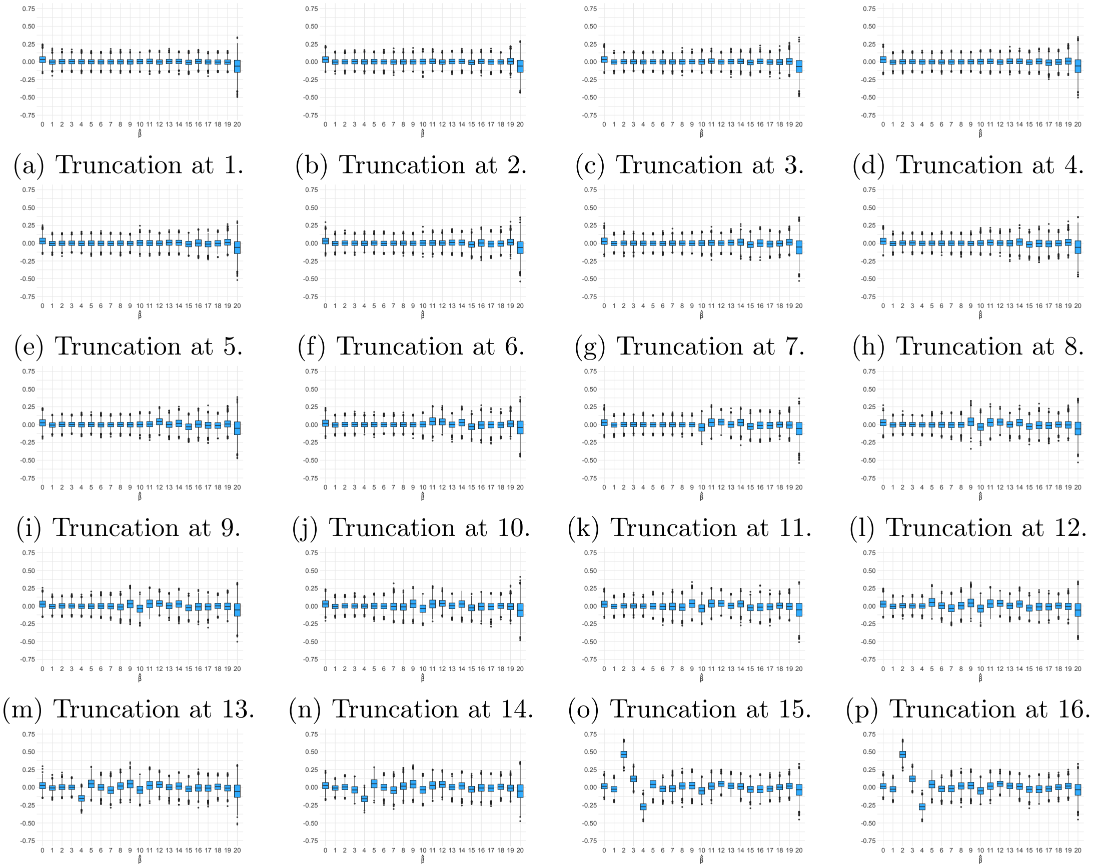


Figure 21: Simulated real data: Results of an AIA w.r.t. sensitive covariate X_1 (top row), X_6 (middle row) and X_{11} (bottom row) and randomly sampled (blue) and handpicked, outlying target observations (orange) measured by $MSE_S(x_{t,s}|s_t = 1)$. Synthetic data are generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes (right) for privacy parameter $\epsilon \in \{0.1, 1, 5\}$. Results are reported as box plots over 10 AIA game iterations and 4 outlying (orange) and 5 randomly sampled (blue) target observations respectively. Parameters of the generative models and privacy attacks can be found in Appendix J.



(a) Truncation at 1.

(b) Truncation at 2.

(c) Truncation at 3.

(d) Truncation at 4.

(e) Truncation at 5.

(f) Truncation at 6.

(g) Truncation at 7.

(h) Truncation at 8.

(i) Truncation at 9.

(j) Truncation at 10.

(k) Truncation at 11.

(l) Truncation at 12.

(m) Truncation at 13.

(n) Truncation at 14.

(o) Truncation at 15.

(p) Truncation at 16.

(q) Truncation at 17.

(r) Truncation at 18.

(s) Truncation at 19.

(t) No truncation.

Figure 22: Regression coefficients of an AIA on simulated real data from L with randomly sampled target observations w.r.t. sensitive covariate X_1.

M Real-World Data: SUPPORT2

The SUPPORT2 data set used in Section 3.1 is processed version of the raw SUPPORT2 data set by Harrell (2022b). The raw SUPPORT2 data 'comprises 9105 individual critically ill patients across 5 United States medical centers, accessioned throughout 1989-1991 and 1992-1994. Each row concerns hospitalized patient records who met the inclusion and exclusion criteria for nine disease categories: acute respiratory failure, chronic obstructive pulmonary disease, congestive heart failure, liver disease, coma, colon cancer, lung cancer, multiple organ system failure with malignancy, and multiple organ system failure with sepsis', (Harrell, 2022a).

In the processing step covariates *age*, *slos*, *num.co*, *scoma*, *charges*, *totcst*, *totmcst*, *sps*, *aps*, *surv2m*, *surv6m*, *hday*, *prg2m*, *dnrday*, *meanbp*, *wblc*, *hrt*, *resp*, *temp*, *pafi*, *alb*, *bili*, *crea*, *sod*, *ph*, *bun* and *death* from the raw data are kept where the bivariate covariate *death* is considered as response variable and renamed to Y. Additionally, all rows containing missing data are left out. The resulting SUPPORT2 data set contains $n = 1104$ observations and $d = 27$ covariates including response Y. Of these data, 220 randomly selected observations (equalling 20%) are stored away and only accessed later for assessing the utility of the synthetic data. For parameter tuning of the generative models, synthetic data generation, as well as privacy attacks, the remaining 884 observations are used.

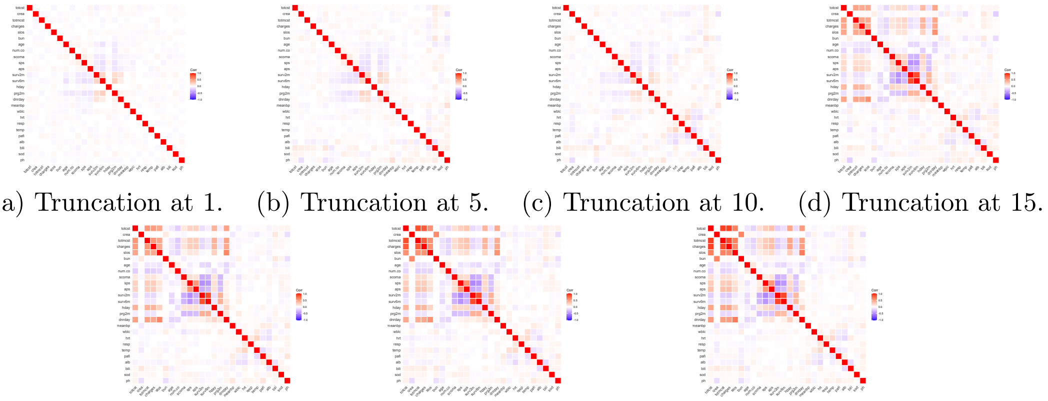
For the PrivPGD model to hold its DP guarantees, the ranges of all covariates need to be inferred from a source that is independent of the actual SUPPORT2 data. This is well possible for covariates describing features that inherently have limits outside which they lose their meaning (for example age or respiratory rate cannot be negative) and more difficult for other covariates. The following ranges have been inferred together with an MD:

- *totcst*: Total ratio of costs to charges (RCC) cost. Range $[0, 500000]$.
- *crea*: Serum creatinine levels measured at day 3. Range $[0, 13]$, assuming measurements in milligrams per liter (Finney et al., 2000).
- *totmcst*: Total micro cost. Range $[0, 500000]$.
- *charges*: Hospital charges (in \$). Range $[0, 500000]$.
- *slos*: Days from Study Entry to Discharge. Range $[0, 180]$ (Knaus et al., 1995).
- *bun*: Blood urea nitrogen levels measured at day 3. Range $[0, 120]$ (BUN).
- *age*: Age of the patients in years. Range $[18, 115]$ (Knaus et al., 1995).
- *num.co*: The number of simultaneous diseases (or comorbidities) exhibited by the patient. Range $[0, 9]$ Harrell (2022a).
- *scoma*: SUPPORT day 3 Coma Score based on Glasgow scale. Range $[0, 15]$ (Teasdale and Jennett, 1974).
- *sps*: SUPPORT physiology score on day 3. Range $[0, 163]$ (Le Gall et al., 1993; sap).
- *aps*: APACHE III day 3 physiology score. Range: $[0, 299]$ (Knaus et al., 1991).
- *surv2m*: SUPPORT model 2-month survival estimate at day 3. Range $[0, 1]$ (Knaus et al., 1995).
- *surv6m*: SUPPORT model 6-month survival estimate at day 3. Range $[0, 1]$ (Knaus et al., 1995).
- *hday*: Day in hospital at which patient entered study. Range $[0, 180]$ (Knaus et al., 1995).
- *prg2m*: Physician's 2-month survival estimate for patient. Range $[0, 1]$.
- *dnrday*: Day of DNR (Do Not Resuscitate) order (<0 if before study). Range $[-30, 180]$ (Knaus et al., 1995).
- *meanbp*: Mean arterial blood pressure of the patient, measured at day 3. Range $[0, 180]$ (McEvoy et al., 2024).
- *wblc*: Counts of white blood cells (in thousands) measured at day 3. Range $[0, 100]$ (Riley and Rupert, 2015).
- *hrt*: Heart rate of the patient measured at day 3. Range $[0, 200]$ (ACC).

- *resp*: Respiration rate of the patient measured at day 3. Range $[0, 40]$ (ATS).
- *temp*: Temperature in Celsius degrees measured at day 3. Range $[27, 42]$.
- *pafi*: PaO_2/FiO_2 ratio measured at day 3. The ratio of arterial oxygen partial pressure (PaO2 in mmHg) to fractional inspired oxygen (FiO2 expressed as a fraction). Range $[0, 500]$.
- *alb*: Serum albumin levels measured at day 3. Range $[0, 5]$ (Ref).
- *bili*: Bilirubin levels measured at day 3. Range $[0, 21]$ (Ref).
- *sod*: Serum sodium concentration measured at day 3. Range $[120, 160]$ (Ref).
- *ph*: Arterial blood pH. Range $[6.9, 7.8]$ (Ignatavicius et al., 2017).

M.1 Estimated Correlation Matrix of C-Vine Generated Synthetic Data Generated Per Truncation Level on Real-World SUPPORT2 Data

In Figure 23 we observe how the correlation structure of the real SUPPORT2 data presented is more and more reproduced in the synthetic data generated by a C-vine with increasing truncation level. Specifically, we note that dependencies in the first block containing sensitive covariates *crea* and *totcst* start to be represented in synthetic data generated by a C-vine from truncation level 15.



(a) Truncation at 1. (b) Truncation at 5. (c) Truncation at 10. (d) Truncation at 15.

(e) Truncation at 20. (f) No truncation. (g) Real SUPPORT2 data.

Figure 23: SUPPORT2 data: The matrices of pairwise Kendall's τ of continuous covariates in synthetic data generated with a C-vine for truncation at levels $t \in \{1, 5, 10, 15, 20\}$ and no truncation illustrate how the rank correlation structure of the SUPPORT2 data in (g) is more and more reproduced in the synthetic data with increasing truncation level.

M.2 R-Vine Matrix of the C-Vine Used as a Generative Model on Real-World SUPPORT2 Data

The R-vine matrix of the C-vine used as a generative model on real-world SUPPORT2 data is as follows with Y on index 27 and index $j \in [26]$ corresponding to covariate X_j:

$$\begin{pmatrix} 27 & 27 & 27 & \cdots & 27 & 27 \\ & 26 & 26 & \cdots & 26 & 26 \\ & & 25 & \cdots & 25 & 25 \\ & & & \ddots & \vdots & \vdots \\ & & & & 2 & 2 \\ & & & & & 1 \end{pmatrix}. \tag{148}$$

In the first vine tree the response Y is in the center. By this it is enforced that pairwise dependencies between Y and the covariates are modeled.

Table 4: Target observations of the SUPPORT2 data set, Section M that are handpicked to lie outside the 95%-quantile of the respective sensitive covariate *crea* and *totcst*.

	QUANTILES			
SENSITIVE COVARIATE	0.01	0.025	0.975	0.99
crea	ID820	ID45	ID403	ID447
totcst	ID806	ID823	ID31	ID41

Table 5: Randomly sampled target observations from the SUPPORT2 data set of Section M and their corresponding quantiles w.r.t. covariates *crea* and *totcst*.

	TARGET IDs			
SENSITIVE COVARIATE	ID123	ID507	ID589	ID740
crea	0.966	0.984	0.506	0.957
totcst	0.374	0.924	0.683	0.615

M.3 Choice of Target Observations for Privacy Evaluation on Real-World SUPPORT2 Data

We conduct an AIA and MIA on SUPPORT2 data described in M. The parameter setup of the privacy attacks can be found in Table 1. For the attacks, four target observations are handpicked outside the 95%-quantile of the regarding sensitive covariate for each sensitive covariate, see Table 4.

Additionally, four target observations, namely ID123, ID507, ID589 and ID740[14], are randomly sampled from the real data set. They correspond to the quantiles w.r.t. the respective sensitive covariate given in Table 5.

M.4 Attribute Inference Attack: Results in Terms of MSE

Figure 24 displays the MSE under an AIA w.r.t randomly sampled and outlying targets w.r.t. sensitive covariate *totcst*.

M.5 Attribute Inference Attack: Results in Terms of WCAB

Figure 25 displays the AIA results in terms of WCAB. We observe that the WCAB approximately replicates the trend of the MAB for the C-vine, CTGAN, TVAE and PrivPGD, see Figure 2a. The WCAB of PrivBayes for $\epsilon \in \{0.1, 5\}$ on the other hand lies above the one of the C-vine truncated at level 10 or lower and the one of CTGAN for sensitive attributes *crea* and *totcst*). This indicates that even though the PrivBayes provides formal guarantees on privacy leakage on a single individual that translate to theoretical bounds on the PG in an MIA, the PrivBayes might in the worst case leak dependencies that inform the sensitive covariate in an AIA from the real into the synthetic data, even for low ϵ.

M.6 Privacy-Utility Plots on Real-Wolrd SUPPORT2 Data: Additional Plots

Figure 26 displays the privacy-utility plots w.r.t. a MIA and and sensitive features *totcst* and *crea* based on the results of Sections 3.1 and 3.1. As already observed in Sections 3.1 and 3.1, all models except for TVAE score a PG of around 1. Compared to the competitor models that are able to protect sensitive covariates against a MIA, the C-vine scores the highest utility.

M.7 Statistical Fidelity

M.7.1 Statistical Discrepancy

We measure the statistical discrepancy between joint real and synthetic distribution with α-precision, β-recall and authenticity (P_α, R_β, A) introduced by Alaa et al. (2022). Their definition can be found in Appendix I.

[14] NB: IDk corresponds to the $(k+1)$th observation in the real data set with $k \in \{0, ..., (n-1)\}$.

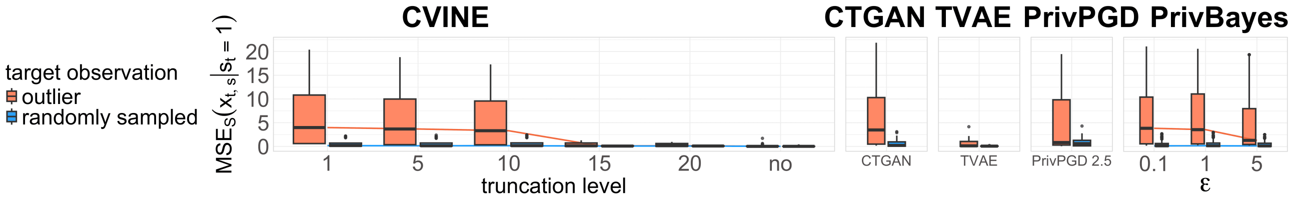


Figure 24: SUPPORT2 data: MSE under an AIA w.r.t randomly sampled (blue) and outlying targets (orange) w.r.t. sensitive covariate *totcst*.

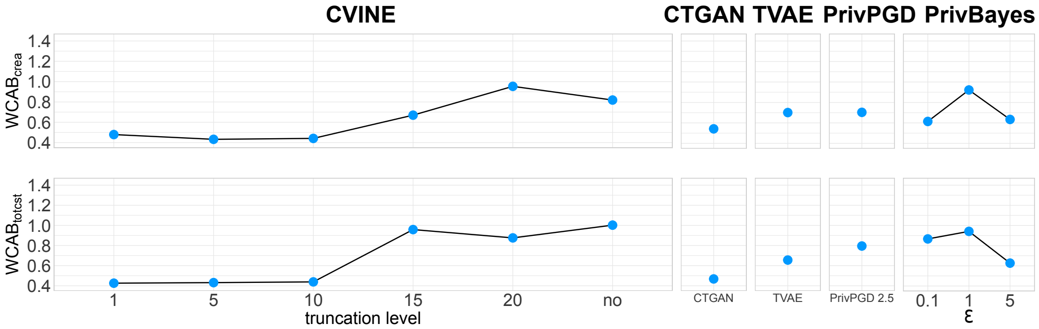


Figure 25: SUPPORT2 data: Results of an AIA w.r.t. sensitive covariate *crea* (top row) and *totcst* (bottom row) measured by $WCAB_{j^*}$. Synthetic data are generated with a C-vine for different truncation levels (left), CTGAN (2nd), TVAE (3rd), PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$ (4th) and PrivBayes (right) for privacy parameter $\epsilon \in \{0.1, 1, 5\}$. Results are reported over 10 AIA game iterations.

As for the simulated real data, an increasing truncation level of the C-vine leads to an increase in fidelity and diversity while it decreases the generalization, see Figure 27. Highest fidelity and diversity of an un-truncated C-vine results in lowest generalization compared to competitor models. The PrivBayes model reaches a generalization of up to 0.99 but its generated samples fail to resemble and cover the real data. The CTGAN generated synthetic data achieve a high generalization but struggle to be diverse enough to cover the real data. The TVAE generates synthetic data with very high fidelity, high generalization and moderate diversity.

Additionally, we evaluate the generative models by comparing empirical marginal histograms on real and synthetic data in Appendix M.7.2.

M.7.2 Comparing Marginal Histograms

On the SUPPORT2 data we generate synthetic data with the C-vine for different truncation levels and with the competitor models (CTGAN, TVAE, PrivBayes with $\epsilon \in \{0.1, 1, 5\}$). We select the 6 covariates *age*, *aps*, *surv2m*, *resp*, *alb*, and *ph* for which we want to show to empirical marginal histograms for real data (in blue) and for synthetic data (in red) superimposed, an overlap of the empirical marginal histograms results in a purple color. The 6 covariates are selected to represent the most challenging marginal distributions according to visual analysis of the empirical marginal histogram plots. In Figure 28 we compare the empirical marginal histograms of synthetic data generated by a C-vine with truncation levels 1 and 10 and with no truncation with the real data. It shows almost perfect overlap independent of the truncation level. The reason for this extraordinary good marginal overlap is the fact the vine copulas allow to model marginal distribution separately from the joint dependence structure. Thus, independent of how close the joint dependence structure present in the real data is modeled through the vine copula (e.g. if we truncate at an early tree level this will be worse than for the un-truncated vine), the marginal distributions are always captured well (if there is enough data to model them).

In Figures 29 and 30 the empirical marginal histograms of the competitor models are compared. Particularly, we observe that the PrivBayes model (for various ϵs) considerably struggles to reproduce marginal distributions of the real data confirming its low values of P_α and R_β and high authenticity in Figure 27 in Section M.7.1.

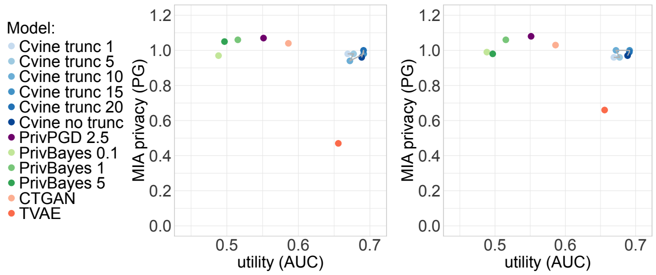


Figure 26: SUPPORT2 data: Privacy-utility plot w.r.t. MIA and sensitive features *totcst* (left) and *crea* (right) of a C-vine with truncation levels $t \in \{1, 5, 10, 15, 20\}$ and no truncation, PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$, PrivBayes model with $\epsilon \in \{0.1, 1, 5\}$, CTGAN and TVAE on SUPPORT2 data of Section M. The median MIA PG over all game iterations and utility for 50 synthetic data sets are reported.

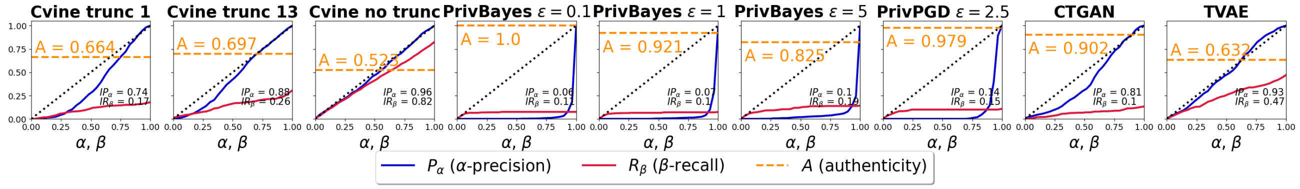
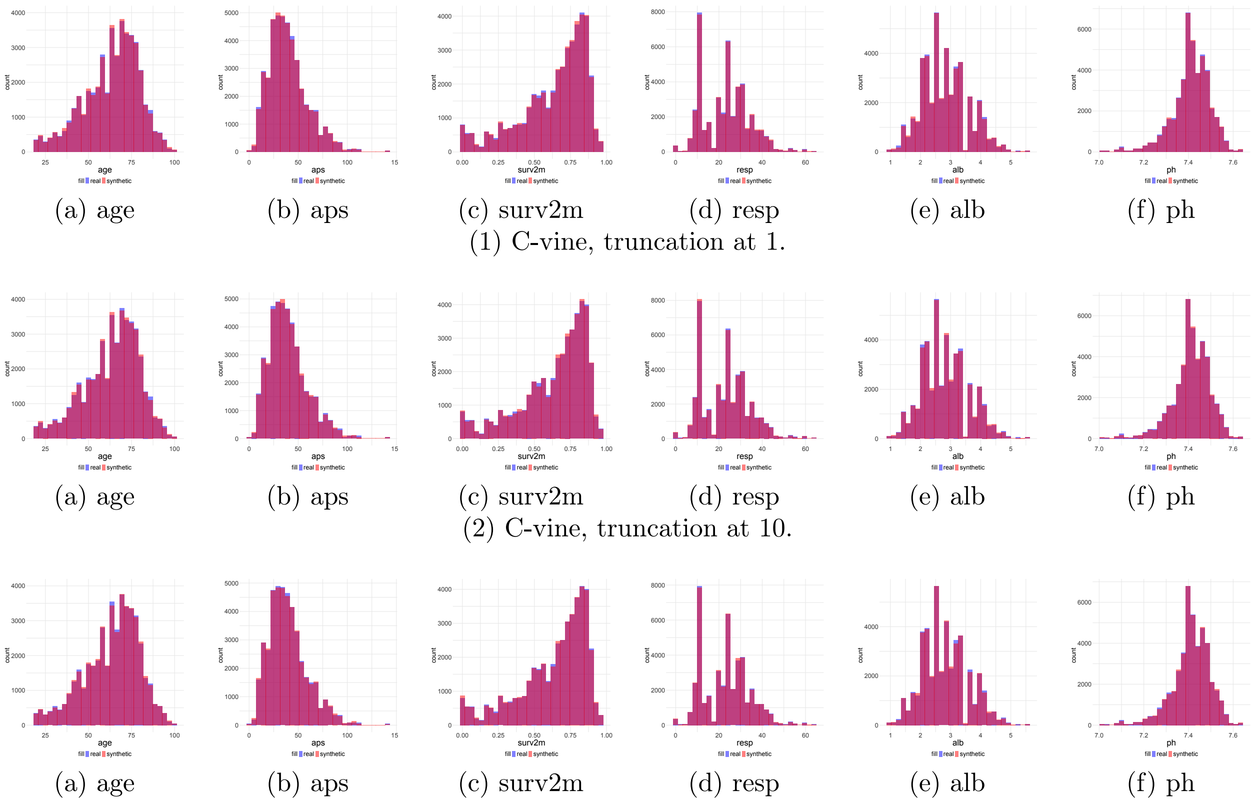


Figure 27: SUPPORT2 data: Fidelity (α-precision), diversity (β-recall) and generalization (authenticity) of synthetic data generated in the order C-vine for truncation at levels 1 and 10 and no truncation, PrivBayes for $\epsilon \in \{0.1, 1, 5\}$, PrivPGD with $\epsilon = 2.5$, $\delta = 10^{-5}$, CTGAN and TVAE from SUPPORT2 data.



(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(1) C-vine, truncation at 1.

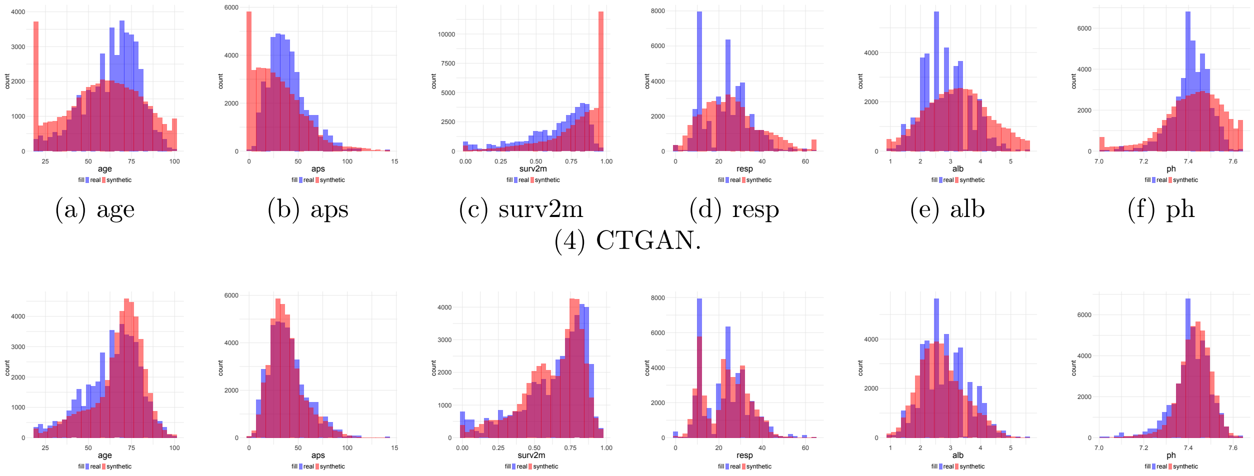
(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(2) C-vine, truncation at 10.

(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(3) C-vine, no truncation.

Figure 28: SUPPORT2 data: Overlapping empirical marginal histograms of covariates *age, aps, surv2m, resp, alb* and *ph* estimate on the real data (blue) and synthetic data (red) generated by a C-vine with truncation level 1 and 10 and no truncation.



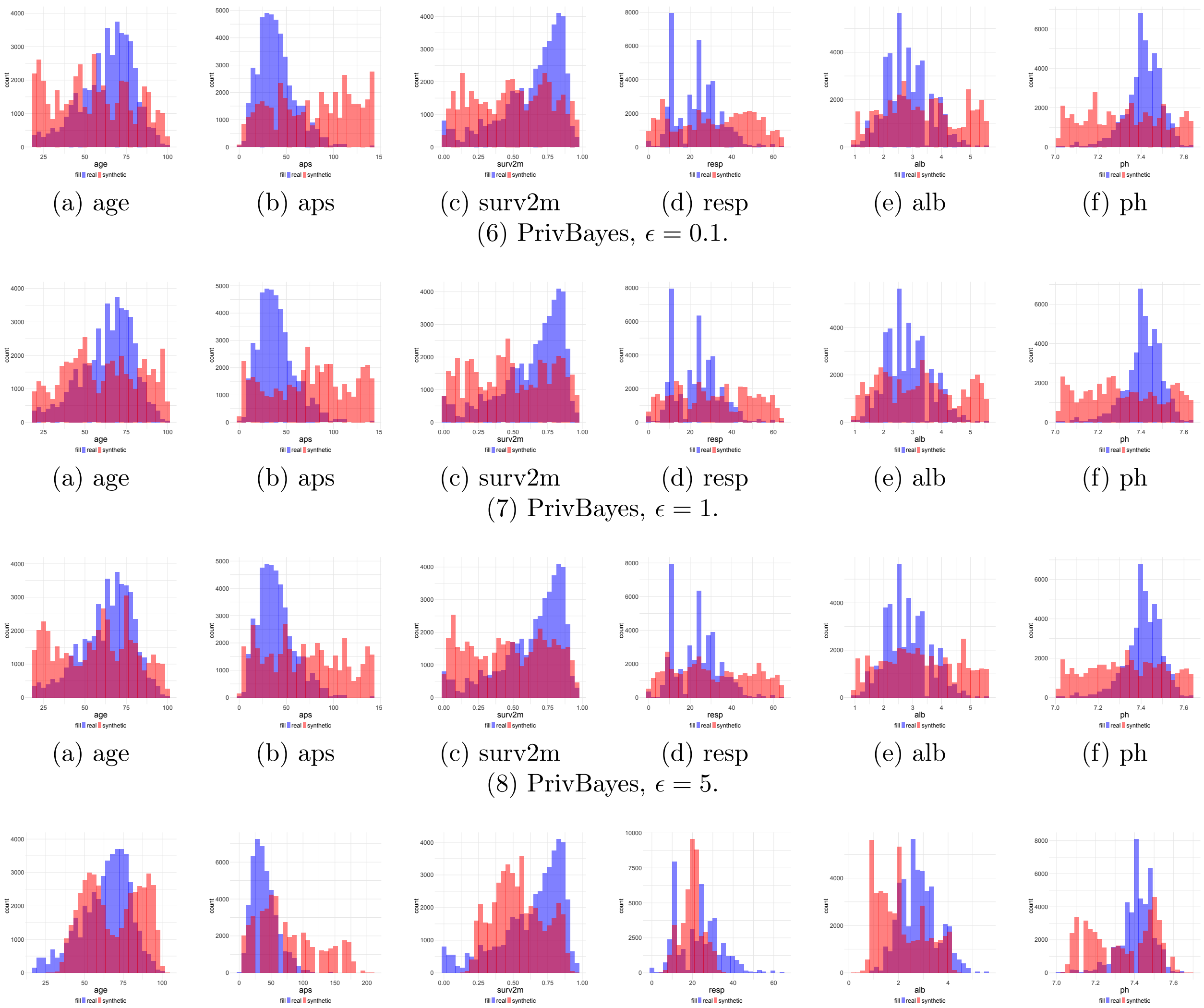
(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(4) CTGAN.

(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(5) TVAE.

Figure 29: SUPPORT2 data: Overlapping empirical marginal histograms of covariates *age, aps, surv2m, resp, alb* and *ph* estimate on the real data (blue) and synthetic data (red) generated by a CTGAN and TVAE.



(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(6) PrivBayes, $\epsilon = 0.1$.

(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(7) PrivBayes, $\epsilon = 1$.

(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(8) PrivBayes, $\epsilon = 5$.

(a) age (b) aps (c) surv2m (d) resp (e) alb (f) ph

(9) PrivPG, $\epsilon = 2.5$, $\delta = 10^{-5}$.

Figure 30: SUPPORT2 data: Overlapping empirical marginal histograms of covariates *age, aps, surv2m, resp, alb* and *ph* estimate on the real data (blue) and synthetic data (red) generated by PrivBayes with $\epsilon \in \{0.1, 1, 5\}$ and PrivPGD with $\epsilon = 2.5$ and $\delta = 10^{-5}$.