Multi-Dependence Conditional Vector - Boosting Categorical Fidelity in Tabular-Data GANs

Melle Mendikowski[1], Benjamin Schindler[2,3], Thomas Schmid[2,4], Ralf Möller[1], and Mattis Hartwig[5,6]

[1] University of Hamburg, Mittelweg 177, 20148 Hamburg, Germany
melle.mendikowski@uni-hamburg.de
[2] MLU Halle-Wittenberg, Magdeburger Str 8, 06120 Halle, Germany
[3] Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany
[4] Lancaster University in Leipzig, Nikolaistraße 10, 04109 Leipzig, Germany
[5] German Research Center for Artificial Intelligence, Ratzeburger Allee 160, 23562 Lübeck, Germany
[6] singularIT GmbH, Inselstraße 27, 04103 Leipzig, Germany

Abstract. Modern deep learning's demand for huge training datasets collides with data privacy concerns, creating a critical data availability gap. Synthesis of tabular training data using generative models such as conditional GANs offers a promising solution, but the utility of current models is still limited as they show problems in capturing complex relationships of categorical features. Therefore, we introduce the concept of multi-dependence conditional vectors (*M-DCV*), which allow for setting multiple conditions on the creation of categorical data directly during the training and sampling process of conditional GANs. We propose three different strategies to create *M-DCV*: first, based on Bayesian Networks (*M-DCV* Bayes), second on the association measurement Cramer's V (*M-DCV* Cramer), and third one involving a cloning process (*M-DCV* RPC). We evaluate all approaches on two state-of-the-art GAN architectures and three diverse datasets using a holistic evaluation pipeline in terms of data utility, statistical fidelity and privacy risks. Across all experiments, there is an overall average improvement in the quality of categorical fidelity by 33.2% with the RPC approach, 13.2% with the Bayes method, and 8.8% using the Cramer modification. Additionally, we can further enhance the machine learning utility with all *M-DCV* variants, although this comes at the cost of reduction in privacy scores.

Keywords: Synthetic Tabular Data · Conditional GANs · Inter-Column Dependencies

1 Introduction

Access to sufficient high-quality training data remains the critical bottleneck in many machine learning (ML) applications [1]. This problem is especially pronounced in jurisdictions with stringent data protection laws, such as the European Union's General Data Protection Regulation (GDPR) or Canada's Personal Information Protection and Electronic Documents Act (PIPEDA). While

such regulatory frameworks introduce practical constraints for data-driven development, they also represent a significant achievement in protecting individual rights and promoting responsible data governance [22].

Synthetic data generation has emerged as a promising strategy to address this tension, aiming to replicate the statistical properties of real datasets without disclosing information about actual individuals [3]. A major breakthrough in synthetic data generation has been the introduction of Generative Adversarial Networks (GANs) [12], which demonstrate strong generative performance, especially in the creation of synthetic image datasets [2]. The adaptation of GAN architectures to the domain of tabular data has led to the development of several state-of-the-art models, most notably the widely recognized *Conditional Tabular GAN* (CTGAN) [23]. CTGAN addresses class imbalance in categorical columns by introducing a conditional vector that encodes a single category from a randomly selected feature, serving as a conditioning signal for both the *generator* and the *discriminator*. Despite this progress, preserving inter-column dependencies in synthetic tabular data is still largely unsolved. Empirical studies, including evaluations of conditional GANs, show that all recent state-of-the-art models frequently violate logical constraints (e.g., implausible category combinations) and rarely maintain stricter functional dependencies, with corresponding drops in downstream ML utility [25, 16, 17, 21].

To address this limitation in existing conditioning strategies, we introduce the *Multi-Dependence Conditional Vector* (*M-DCV*), an extension of the CTGAN's conditioning mechanism that enables multiple categorical constraints to be applied during training and sampling. The *M-DCV* is designed to capture complex interdependencies between categorical columns, thereby enhancing the structural fidelity and realism of the synthesized tabular data. We introduce three distinct *M-DCV* strategies: (i) *M-DCV* Bayes propagates coherent variable assignments based on Bayesian Networks learned from the training data; (ii) *M-DCV* Cramer identifies strongly associated column pairs using Cramér's V [10]; and (iii) *M-DCV Randomised Partial Cloning* (RPC) transfers categorical fragments from real data rows directly into the synthesis process. To evaluate all *M-DCV* variants, we present a modular benchmarking pipeline that jointly assesses synthetic data quality across three key dimensions: ML utility, statistical fidelity, and privacy risk. Using our integrated evaluation framework we present a holistic analysis of the trade-offs involved in multi-dependence conditioning, neccesary for the development of generative tabular models that are both high-quality and privacy-aware.

2 Preliminaries & Related Work

GANs consist of two neural networks: a *generator* that produces synthetic data and a *discriminator* that learns to distinguish real from fake samples, with both networks improving iteratively through adversarial feedback to approximate the real data distribution [12]. Mirza and Osindero extended the original GAN framework by concatenating a one–hot condition vector to the generation process [18].

Their augmentation directly guides the generator to produce specified class label outputs and has become a standard technique for class-aware generation across domains. Building on this idea, Xu et al. introduced CTGAN, a conditional GAN architecture specifically designed for tabular data [23]. CTGAN uniformly selects a single categorical column, samples one of its categories according to a logarithmic *probability mass function* (PMF) that up-weights rare classes, and encodes this choice as a sparse one-hot *conditional vector* concatenated to the generator input. During the same step the *discriminator* receives only real data rows matching the chosen category, and the generator is assigned an auxiliary cross-entropy penalty if the synthesized row violates the *conditional vector*, allowing the model to mitigate categorical class imbalance. Zhao et al. extended CTGAN to CTAB-GAN by introducing mixed-variable encodings and additional loss components to better align first- and second-order statistical moments of the generated data [26].

Bayesian Networks (BNs) are designed to model the dependency structure among categorical variables by factorising their joint probability distribution into a set of local conditional distributions. Concretely, a BN represents the variables $X_1, \ldots, X_d$ as nodes of a directed acyclic graph G; for each node X_i the incoming edges encode its parent set $\mathrm{pa}(X_i)$. The joint distribution then factorises as

$$P(X_1, \ldots, X_d) \;=\; \prod_{i=1}^{d} P\big(X_i \,\big|\, \mathrm{pa}(X_i)\big),$$

where each factor $P(X_i \mid \mathrm{pa}(X_i))$ is represented by a conditional-probability table. By decomposing the joint distribution into local conditionals, a BN models complex categorical dependencies with tractable Bayes-based inference, while its graph transparently encodes the underlying conditional independencies and thus remains human interpretable [15]. BNs main limitation for usage in tabular synthesis is that they handle continuous columns only via ad-hoc discretisation or specialized hybrid extensions. To leverage the strengths of BNs for modeling categorical tabular data, Zhang et al. proposed GAN-BLR [25], a variant of the GAN framework in which both *generator* and *discriminator* are replaced with Bayesian Networks. While this enforces strong discrete fidelity, the resulting model lacks the capacity to generate continuous features. Although the extension, GAN-BLR++ [24] is able to automatically convert every continuous column entry into a Dirichlet-process Gaussian-mixture code and can therefore synthesize tabular data with continuous columns,the underlying BN still learns purely over discretized continuous states.

3 Methodology

3.1 Multi-Dependence Conditional Vector (*M-DCV*)

The *M-DCV* extends the one-hot *conditional vector* of CTGAN by injecting several categorical assignments rather than only a single choice. Following the

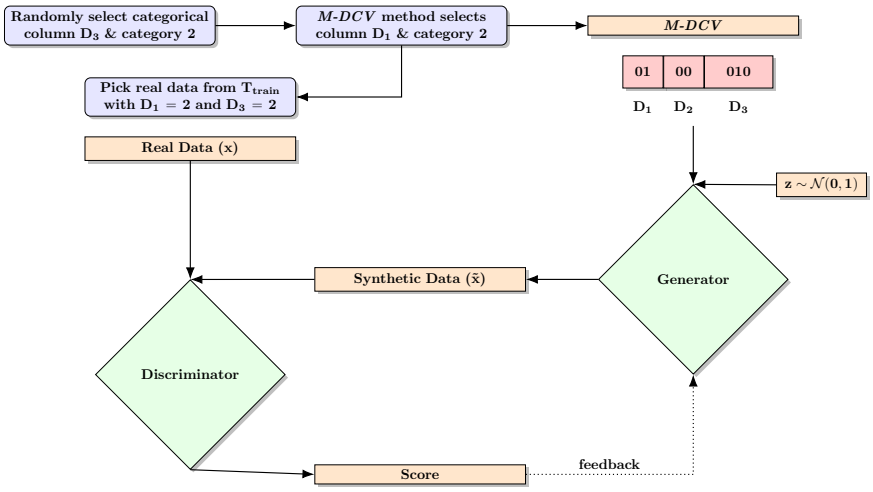


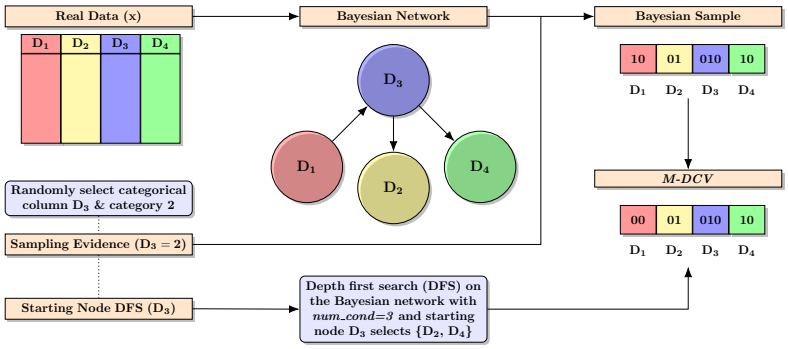
Fig. 1: Integration of the *M-DCV* into the conditional GAN architecture, here starting from the *primary* condition ($D_3 = 2$).

Train on Synthetic, Test on Real (TSTR) approach [8] the real data is split into T_{train} and T_{test}, let $\{D_1, \ldots, D_K\}$ be the categorical columns of that data. During each training step we sample the *primary* condition ($D_{i^\star} = k^\star$) as in Xu et al. [23]: (i) select a column $D_{i^\star}$ uniformly from $\{1, \ldots, K\}$; (ii) draw a category $k^\star \in D_{i^\star}$ from that column's log-frequency PMF. Starting from this anchor, an *M-DCV* method can select up to $K-1$ *secondary* conditions that reflect empirically observed inter column dependencies. Then, an empty one-hot mask m_i is assigned for each categorical column D_i in T_{train} (m_1 is therefore the empty mask for the column D_1), in accordance to the *conditional vector* method. Finally, the complete *M-DCV* is obtained by concatenating all mask vectors and then activating the entries specified by the set of active conditions $\mathcal{C}$, resulting in the sparse binary vector:

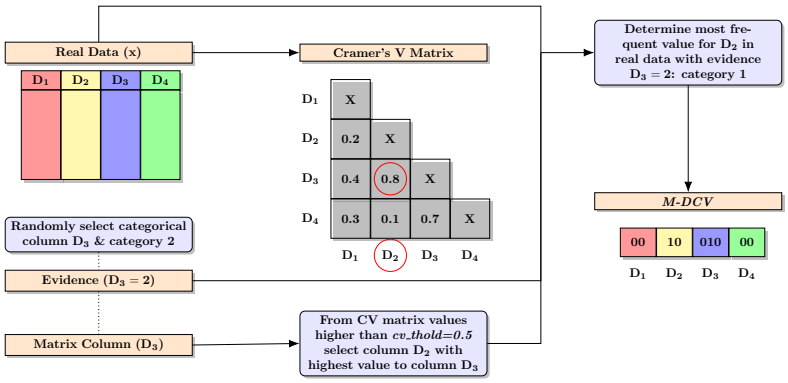
$$\textit{M-DCV} = m_1 \oplus \cdots \oplus m_K, \qquad m_i^{(k_i)} = \begin{cases} 1 & \text{if } (i, k_i) \in \mathcal{C}, \\ 0 & \text{otherwise.} \end{cases}$$

The *generator* receives *M-DCV* together with noise z, while the *discriminator* is forced to compare against real data rows that satisfy the same allocations (cf. Fig. 1). To propagate the constraints, we add a multi-label cross-entropy term $\mathcal{L}_{\text{cond}} = \sum_{(i,k_i)\in\mathcal{C}} \text{CE}\big(\hat{D}_i = k_i,\ D_i = k_i\big)$ to the GAN objective, forcing each synthesized column that is included in $\mathcal{C}$ to match its target category. We investigate three distinct strategies for selecting *secondary* conditions in the *M-DCV*:

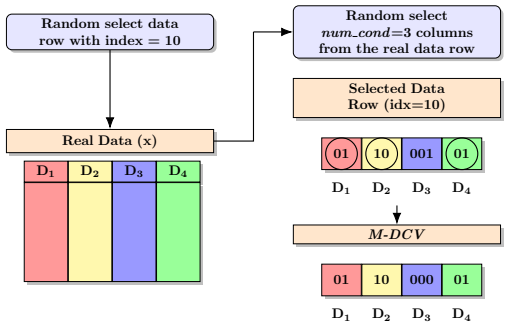
***M-DCV* Bayes:** From the categorical subset of T_{train} we learn a BN with hill-climb search and MAP parameter estimation [15, 19]. Given the *primary* condition ($D_{i^\star} = k^\star$), likelihood-weighted sampling produces a new synthetic



(a) Creating the *M-DCV* using a Bayesian Network, demonstrated using primary condition ($D_3 = 2$) and `num_cond` $= 3$.



(b) Creating the *M-DCV* using the Cramer's V Matrix, demonstrated using primary condition ($D_3 = 2$) and `cv_thold` $= 0.5$.



(c) Creating the *M-DCV* using Randomized Partial Cloning, demonstrated on a single training-row fragment (index 10) and `num_cond` $= 3$.

Fig. 2: Overview of the three proposed strategies for constructing *M-DCV*:
(a) M-DCV Bayes **(b)** M-DCV Cramer **(c)** M-DCV RPC

row using the trained BN. A depth-first traversal of the BN, bounded by our new hyper-parameter `num_cond` $\in [1, K]$, extracts the `num_cond` -1 most proximate variables; their sampled categories together with the *primary* condition are incorporated into $\mathcal{C}$ (Fig. 2a). Because a BN sampler can generate novel category combinations, there are iterations in which no real data row matches every element of $\mathcal{C}$. In that case we uniformly draw the *discriminator* input from the real data rows whose attributes satisfy a maximal, equal-sized subset of $\mathcal{C}$.

***M-DCV* Cramer:** We first compute the Cramér's V association matrix $V \in \mathbb{R}^{K \times K}$ on T_{train}, a symmetric measure that ranges from 0 (no association) to 1 (perfect association) for any two categorical variables [10]. With the new hyperparamter `cv_thold` $\in [0, 1]$ as a treshhold we keep all pairs whose association exceeds `cv_thold`; among them the column D_j with $\max V_{i^\star j}$ to the the *primary* condition is chosen and the most co-occurring category of D_j with $(D_i^\star, k^\star)$; this pair forms $\mathcal{C}$ (Fig. 2b). At most one *secondary* condition is introduced, to steer the *generator* towards the strongest empirical dependency.

***M-DCV Randomized Partial Cloning* (RPC):** RPC selects a random row $r \in [1, |T_{\text{train}}|]$ and clones a subset of categorical values from r, where the number of cloned entries is determined by the hyperparameter `num_cond`. These values are incorporated into the condition set $\mathcal{C}$ (Fig. 2c). Because all assignments originate from a real training record, the resulting vector always reflects empirically valid dependencies.

3.2 Evaluation Pipeline

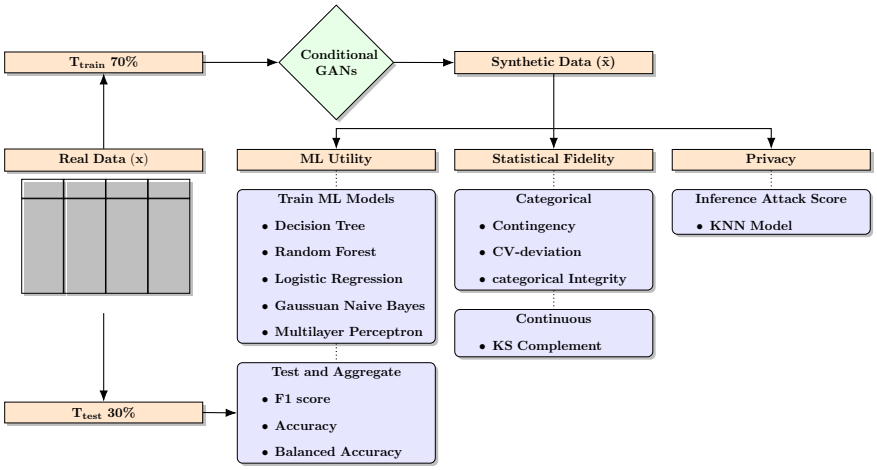


Fig. 3: A concise overview of our evaluation pipeline for synthetic tabular data, detailing three main evaluation points: ML *utility*, statistical *fidelity*, and *privacy*.

Table 1: Basic overview of the three datasets used for the experimental evaluation of our proposed *M-DCV* strategies.

Dataset	**adult**	**aids**	**cancer**
# of rows	32 561	2 139	668
# of columns	14	24	33
# of categorical columns	10	15	25
Domain	social	medical	medical
CDP	1.219×10^9	4.915×10^4	1.049×10^7
ML target column	income	target	Dx:Cancer
Source	UCI[7]	UCI[8]	UCI[9]

Our evaluation protocol (Fig. 3) unifies *utility*, *fidelity*, and *privacy* into a single pass, thereby providing a concise yet multi-faceted judgement of every synthesized dataset.

(1) Machine–learning utility. Following the TSTR protocol [8], we evaluate the utility of each synthetic data set on five widely used classifiers: Decision Tree, Random Forest, Logistic Regression, Gaussian Naive Bayes, and Multilayer Perceptron. We report the arithmetic mean of F_1, accuracy, and balanced accuracy achieved across these five models, as recommended in [9, 17].

(2) Statistical fidelity. Four complementary metrics quantify how well synthetic tables reproduce the original data characteristics:

- *Contingency Similarity* (CS) from SDMetrics [4] – average pairwise agreement of categorical joint distributions;
- *CV-Deviation* (CV-d) [16] – the RMSE between Cramér's V of every categorical pair in real and synthetic data;
- *Categorical Integrity* [16] – synthetic data's share of correct categorical pairs that admit only specific valid combinations (e.g., postal codes must match the corresponding city);
- *KSComplement* [5, 20] – SDV's Kolmogorov–Smirnov complement averaged over all continuous columns.

Together these measures capture distributional shape, pairwise dependence, and row-level plausibility.

(3) Privacy. To estimate the risk of attribute-inference attacks triggered by our condition-enriched generators, we employ SDMetrics' *CategoricalKNN Privacy Against Inference* [6]. The attacker knows five public (key) attributes drawn from T_{train}, merges them with the information in synthesized data, and predicts two sensitive attributes using a k-NN regressor [13]. The resulting privacy score ranges from 0 (perfect disclosure) to 1 (perfect protection).

[7] `https://archive.ics.uci.edu/dataset/2/adult`

[8] `https://archive.ics.uci.edu/dataset/890/aids+clinical+trials+group+study+175`

[9] `https://archive.ics.uci.edu/dataset/383/cervical+cancer+risk+factors`

4 Experimental Setup

We benchmark *M-DCV* on two leading conditional tabular GANs—CTGAN [23] and CTAB-GAN [26]—across three public datasets spanning social and medical domains (Tab. 1). We train the two models using the authors' default hyper-parameters (*epochs* = 300, *batch_ size* = 500) and add a small grid for our new controls: `num_cond` $\in \{0.25, 0.5, 1.0\} \cdot K$ for Bayes/RPC and `cv_thold` $\in \{0.1, 0.3, 0.5\}$ for Cramer. We also train each model in its original, unmodified form to serve as a baseline. After training, each model emits a synthetic table that matches $|T_{\text{train}}|$.
All code runs in Docker containers on an NVIDIA RTX 3060 under Python 3.10; CTGAN is instantiated via `sdv 0.8.1`, CTAB-GAN via the authors' repository, and Bayesian Networks via `pgmpy 0.1.24`. The entire workflow executes GPU-accelerated CUDA kernels and is fully reproducible from a public repository provided with this paper.

5 Results

Table 2: Best run per dataset, architecture and *M-DCV* variant, selected by maximising the average F_1 score (ties broken by higher CT-Similarity). Each cell reports *Avg* $F_1 \uparrow$ / *CT-Sim* $\uparrow$ / *Privacy* $\uparrow$; the highest value for each metric within each dataset–architecture block is underlined; '—' indicates that the variant did not converge.

Arch.	Variant	adult	aids	cancer
CTGAN	Base	0.650/0.826/0.623	0.483/0.874/<u>0.426</u>	0.435/0.931/1.000
	Bayes	—	0.515/0.899/0.409	0.645/0.932/1.000
	Cramer	0.672/0.850/<u>0.635</u>	0.495/0.909/0.354	0.661/0.927/1.000
	RPC	<u>0.682</u>/<u>0.988</u>/0.543	<u>0.556</u>/<u>0.942</u>/0.349	<u>0.684</u>/<u>0.988</u>/1.000
CTAB-GAN	Base	0.641/0.836/<u>0.662</u>	0.593/0.908/0.400	0.500/0.945/0.972
	Bayes	—	0.692/0.927/0.343	0.520/0.957/0.979
	Cramer	<u>0.672</u>/0.878/0.658	0.626/0.903/<u>0.464</u>	0.489/0.959/0.946
	RPC	0.670/<u>0.968</u>/0.591	<u>0.716</u>/<u>0.983</u>/0.285	<u>0.595</u>/<u>0.987</u>/<u>0.983</u>

Table 2 reports, for each dataset–architecture pair, the run that maximised average F_1 score alongside its unmodified baseline. Augmenting the conditional mask with multiple categorical constraints yields consistent benefits across many evaluation axes: predictive utility rises, statistical fidelity improves, and,except for the RPC variant, privacy remains largely intact. RPC delivers the most pronounced gains, lifting mean F_1 by up to 57 %, and simultaneously achieves the highest CT-Similarity and Categorical Integrity, yet it incurs the steepest reduction in privacy score. The Bayes method could not be applied to the *adult*

Table 3: Average percentual improvement and decline of experiments shown in Table 2 of evaluation metrics across all architectures (CTGAN, CTAB-GAN) on the *cancer* and *aids* datasets relative to their respective unmodified baselines.

Variant	**Avg F_1**	**CV-Dev**	**CT-Sim**	**Cat Int**	**KSC**	**Priv**
M-DCV RPC	28.1 %	62.7 %	6.6 %	30.4 %	–1.1 %	–11.4 %
M-DCV Bayes	18.9 %	26.2 %	1.6 %	11.9 %	–1.2 %	–4.4 %
M-DCV Cramer	14.4 %	11.0 %	1.1 %	14.2 %	–3.5 %	–0.9 %

dataset because its categorical dimensionality product of 1.219×10^9 renders Bayesian-network structure learning and likelihood-weighted sampling computationally infeasible, underscoring a key limitation of BN-based conditioning in high-dimensional discrete spaces.

Table 3 aggregates the percentual improvements on the two medical datasets (*aids* and *cancer*) for all *M-DCV* variants and both architectures, thus providing a concise overview that is unaffected by the aforementioned feasibility constraint on *adult*. The pattern observed in the per-run results is confirmed: RPC raises mean F_1 by 28.1 %, lowers CV-Deviation by 62.7 %, and increases Categorical Integrity by 30.4 %; Bayes and Cramer follow the same monotonic ordering with progressively smaller yet still substantial improvements, demonstrating that the degree to which categorical dependencies are reproduced is correlated with downstream ML utility.

6 Discussion

Table 2 together with the aggregates in Table 3 point to three main insights. **(i) Enhanced categorical fidelity:** adding several categorical assignments per sample lifts all category-sensitive metrics (CV-Deviation, Contingency Similarity, and Categorical Integrity) with the largest gains delivered by the RPC variant. **(ii) Stability of the continuous part:** metrics that depend only on continuous columns like KSComplement, remain mostly unchanged. The *M-DCV* intervenes solely in categorical sampling; preserving the baseline quality of continuous attributes is therefore a positive outcome that explains the concurrent rise in overall predictive utility, where both categorical and continous column quality is of importance. **(iii) A systematic utility–privacy tension:** the same ordering that maximises categorical fidelity (RPC > Bayes > Cramer) minimises the privacy score. RPC injects the most additional conditions by copying fragments from real rows; Bayes contributes fewer assignments because its depth-first traversal of a pruned Bayesian Network often stops before reaching `num_cond`; Cramer appends at most one extra assignment, controlled by `cv_thold`. Consequently, samples become progressively closer to the source distribution and therefore more informative for downstream learning, but this same proximity makes it easier for an attacker to infer sensitive attributes, an effect widely observed in the privacy–utility literature [23, 4].

Our experiments show that the underlying generative architecture type also influences the effect size of *M-DCV* conditioning: In the best-run comparisons, CTGAN shows stronger improvements than CTAB-GAN. This is particularly evident on the *cancer* dataset, where its mean F_1 increases from 0.435 to 0.684, while CTAB-GAN already starts from a higher baseline and therefore shows a smaller gain.

Our findings further demonstrate that Bayesian-network conditioning is infeasible for datasets with higher categorical dimensionality, such as *adult*. However, on the medically oriented *aids* and *cancer* datasets, the Bayes variant performs favourably: it surpasses Cramer in nearly every quality metric while incurring roughly one-third of the privacy loss recorded for RPC. This balance suggests that BN-based *M-DCV* conditioning is well suited to domains with moderate cardinalities such as medical diagnosis codes, where data protection is a critical aspect. Furthermore, the successful integration of BN-derived samples demonstrates that our *M-DCV* can also act as a generic interface: using our approach any external model that generates categorical tuples can be plugged directly into the training process of a given conditional GAN. Such hybrid architectures—fusing human interpretable models like Bayesian Networks with neural network architectures—enhance the transparency and controllability of synthetic data generation. These directions resonate strongly with the goals of the emerging field of neurosymbolic AI [11], which aims to unify the strengths of statistical learning and symbolic reasoning.

Across all experiments, there is an overall average improvement in the quality of categorical data by 33.23% with the RPC approach, 13.23% with the Bayes method, and 8.77% using the Cramer's modification. Although synthetic utility has not yet caught up with real-data performance, multi-conditional vectors demonstrably narrow the gap and offer a controllable trade-off between categorical fidelity and privacy.

7 Conclusion and Outlook

In this paper, we introduce the Multi-Dependence Conditional Vector (*M-DCV*), a novel extension of the conditioning mechanism in tabular GANs that enables multiple categorical constraints to be imposed during both training and sampling. By explicitly encoding inter-column dependencies among categorical features, *M-DCV* addresses a core limitation of existing architectures in capturing discrete structural relationships and substantially enhances categorical fidelity in the generated data. We proposed three strategies for constructing the *M-DCV*: (i) *M-DCV* RPC, which clones partial row fragments from the training data; (ii) *M-DCV* Bayes, which infers high-probability categorical tuples from a Bayesian network; and (iii) *M-DCV* Cramer, which derives secondary conditions from high-association pairs identified via Cramér's V. All variants are evaluated using a holistic benchmarking pipeline that jointly assesses statistical fidelity, machine learning utility, and privacy risks.

While our methods yield significant gains in categorical fidelity, they do not yet

fully close the gap to real data distributions. Moreover, the privacy degradation observed in some variants—most notably *M-DCV RPC*—raises concerns regarding their use in privacy-sensitive domains. Future work should explore the integration of formal privacy-preserving mechanisms such as differential privacy [7], which has been successfully applied in various generative settings [9, 14], to enable more favorable privacy–utility trade-offs.

Beyond improving fidelity, our results highlight the potential of *M-DCV* as an modular interface for integrating complementary generative models directly into GAN-based tabular synthesis. Future work could extend this hybrid design by incorporating interpretable models such as decision trees or other rule-based systems to constrain conditional vectors. This line of research promises improved control over synthetic data properties and a step towards more transparent, human-aligned generation processes.

References

1. Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A.S., Al-dabbagh, B.S.N., Fadhel, M.A., Manoufali, M., Zhang, J., Al-Timemy, A.H., et al.: A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. Journal of Big Data **10**(1), 46 (2023)
2. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Cvae-gan: fine-grained image generation through asymmetric training. In: Proceedings of the IEEE international conference on computer vision. pp. 2745–2754 (2017)
3. Bellovin, S.M., Dutta, P.K., Reitinger, N.: Privacy and synthetic datasets. Stan. Tech. L. Rev. **22**, 1 (2019)
4. docs.sdv.dev: Contingencysimilarity metric (2023), `https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/contingencysimilarity`, accessed: 2024-05-18
5. docs.sdv.dev: Kscomplement metric (2023), `https://docs.sdv.dev/sdmetrics/metrics/metrics-glossary/kscomplement`, accessed: 2024-05-18
6. docs.sdv.dev: Privacy against inference metric (2023), `https://docs.sdv.dev/sdmetrics/metrics/metrics-in-beta/privacy-against-inference`, accessed: 2024-05-18
7. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
8. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017)
9. Fang, M.L., Dhami, D.S., Kersting, K.: Dp-ctgan: Differentially private medical data generation using ctgans. In: International Conference on Artificial Intelligence in Medicine. pp. 178–188. Springer (2022)
10. Feller, W.: Harald cramer, mathematical methods of statistics. The Annals of Mathematical Statistics **18**(1), 136–139 (1947)
11. Garcez, A.d., Lamb, L.C.: Neurosymbolic ai: The 3 rd wave. Artificial Intelligence Review **56**(11), 12387–12406 (2023)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)

13. Jayaraman, B., Evans, D.: Are attribute inference attacks just imputation? In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. pp. 1569–1582 (2022)
14. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations (2018)
15. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
16. Mendikowski, M., Hartwig, M.: Creating customers that never existed: Synthesis of e-commerce data using ctgan. In: 18th International Conference on Machine Learning and Data Mining (MLDM-22). New York, US: IBAI Publishing. pp. 91–105 (2022)
17. Mendikowski, M., Schindler, B., Schmid, T., Möller, R., Hartwig, M.: Improved techniques for training tabular gans using cramer's v statistics. In: Canadian AI (2023)
18. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
19. Robert, C.P., et al.: The Bayesian choice: from decision-theoretic foundations to computational implementation, vol. 2. Springer (2007)
20. Smirnov, N.V.: Table for estimating the goodness of fit of empirical distributions. Annals of Mathematical Statistics **19**(2), 279–281 (1948). https://doi.org/10.1214/aoms/1177730256
21. Umesh, C., Schultz, K., Mahendra, M., Bej, S., Wolkenhauer, O.: Preserving logical and functional dependencies in synthetic tabular data. Pattern Recognition **163**, 111459 (2025)
22. Voigt, P., Von dem Bussche, A.: The eu general data protection regulation (gdpr). A practical guide, 1st ed., Cham: Springer International Publishing **10**(3152676), 10–5555 (2017)
23. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. Advances in neural information processing systems **32** (2019)
24. Zhang, Y., Zaidi, N., Zhou, J., Li, G.: Ganblr++: incorporating capacity to generate numeric attributes and leveraging unrestricted bayesian networks. In: Proceedings of the 2022 SIAM International Conference on Data Mining (SDM). pp. 298–306. SIAM (2022)
25. Zhang, Y., Zaidi, N.A., Zhou, J., Li, G.: Ganblr: a tabular data generation model. In: 2021 IEEE International Conference on Data Mining (ICDM). pp. 181–190. IEEE (2021)
26. Zhao, Z., Kunar, A., Birke, R., Chen, L.Y.: Ctab-gan: Effective table data synthesizing. In: Balasubramanian, V.N., Tsang, I. (eds.) Proceedings of The 13th Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 157, pp. 97–112. PMLR (17–19 Nov 2021), https://proceedings.mlr.press/v157/zhao21a.html