# Improved Techniques for Training Tabular GANs Using Cramer's V Statistics

Melle Mendikowski[†,*], Benjamin Schindler[‡], Thomas Schmid[◇,§,‡], Ralf Möller[†], Mattis Hartwig[†,○,*]
[†] German Research Center for Artificial Intelligence
[‡] Machine Learning Group, Universität Leipzig
[◇] Martin-Luther-Universität Halle-Wittenberg
[§] Lancaster University in Leipzig
[○] singularIT GmbH

**Abstract**

Considering the growing global demand for machine learning training data, synthetic data generation is a reasonable way to address the versatile challenges in data acquisition. Conditional Tabular Generative Adversarial Network (CTGAN), an extension of the widely used Generative Adversarial Network (GAN), is considered one of the most promising techniques in the field of tabular data generation. Despite numerous successes of CTGAN, a lack of preserving categorical dependencies within the data has been identified. In prior work, the Cramer's V (CV) as a natural metric for representing the correlation of categorical dependencies was proposed for hyperparameter tuning of CTGAN models. In this paper, we explore two novel strategies to directly integrate CV statistics of data batches within CTGAN training. The first approach is a generator loss term that penalizes differences between the CV statistics of the original and generated data. The second innovation is the extraction of the CV matrix as an additional feature for the critic. By applying our proposed methods to three benchmark datasets, we improve the averaged accuracy of supervised learning models trained on synthesized data by 11 % compared to the legacy CTGAN. We also outline the impact of CV statistics on preserving dependencies between categorical data columns in terms of integrity and contingency similarity, discuss existing challenges, and identify potential improvements.

**Keywords:** CTGAN, Cramer's V, tabular data synthesis, categorical data, Generative Adversarial Networks

## 1. Introduction

The advent of modern deep learning systems, the associated increase in model size, and the number of training parameters lead to a growing demand for more and more training data [1]. Often, data contains personal information (e.g. in e-commerce [2] or healthcare [3]), which is accompanied by the need to preserve the privacy of the individuals. Reconciling beneficial data sharing with adequate data protection is an ongoing challenge [4] that can be addressed through generative modeling. The central idea is to train a generative model that can subsequently synthesize highly realistic data samples that cannot be linked to individuals in the original dataset. This synthetic data should ideally provide the overall information present in the original data without containing personal information about individuals, and thus can be shared without breaching privacy [5].

The ability to maintain privacy when sharing data is a major ethical win when it comes to data use. Under recent personal data protection legislation (e.g., Canada's Personal Information Protection and Electronic Documents Act and the European General Data Protection Regulation), data synthesis is gaining an important position and opening up a wide range of applications. Often, only big corporations have the ability to collect sufficient data for complex machine learning use cases. Enabling smaller companies to share data

[*] melle.mendikowski@dfki.de, mattis.hartwig@dfki.de

with each other without violating privacy [6] would allow them to reap the competitive benefits of machine learning applications. Also generating more samples from a previously rare category (e.g., rare diseases in electronic health records) could have positive effects [7]. Whereas exploring new types of entries remains difficult because the models tend to prune outliers and shrink distributions [8, 9]. An unsolved issue identified for tabular data synthesis with GANs is the lack of correct representation of interactions between features [10]. In particular, discrete table columns can have very rigid dependencies (e.g., between categories and subcategories), and the ability to preserve such relationships has been investigated very recently [8].To support the preservation of rigid dependencies in tabular GAN, Mendikowski and Hartwig [8] introduced a metric for hyperparameter selection, that is based on Cramer's V (CV) [11] as a statistical measure for dependencies between categorical data.

In this paper, we present two approaches to incorporate CV directly into the GAN training and investigate the effects on the tabular data synthesis. Both approaches compare the original and synthesized data based on the statistical properties of data batches. The first approach translates the CV metric into a loss term that regulates the generating network in order to penalize statistical deviations from the original data. The second approach uses a CV matrix as an additional feature given to the critic to improve the ability to detect categorical dependencies. We evaluate our approach by modifying a state-of-the-art Conditional Tabular Generative Adversarial Network (CTGAN) model for tabular data synthesis [12], and are able to increase the averaged accuracy of supervised learning models by 11 %.

## 2. Preliminaries & Related Work

### 2.1. Generative Adversarial Networks

Unlike discriminative machine learning applications such as regression and classification, generative modeling aims to learn inherent distributions from data [13]. Once trained, an arbitrary number of samples can be derived from a generative model. In general, a distinction is made between explicit models with accessible distribution parameters and implicit models in which the the sampling distribution parameters remain hidden within the model. Explicit models include Probabilistic Graphical Models [14], Variational Autoencoders [15], and Normalizing Flows [16]. To date, Generative Adversarial Networks (GANs) are among the most commonly used implicit generative models [17]. As such, GANs exhibit competitive sample quality, but suffer from training difficulties (e.g., mode collapse and vanishing gradients) and remain more difficult to interpret due to their implicit nature. Therefore, many extensions for GANs have been proposed in recent years to increase the training stability, the quality of the synthetic data, and the model interpretability. While impressive capabilities have been demonstrated in the GAN-based generation of image data (e.g., for artwork [18] or text to image [19], photographs [20]), and audio signals [21], the application of this technique for the synthesis of tabular data is an ongoing area of research [10, 22, 23].

The GAN framework is based on two opposing neural networks, one of which learns to generate data samples and the other distinguishes the generated samples $\tilde{x} \sim P_g$ from real data $x \sim P_r$. An important milestone for the successful training of GANs is the introduction of the Wasserstein GAN (WGAN), which aims to minimize the Earth-Movers distance (also called Wasserstein-1 distance) between the data distribution $P_r$ and the model distribution $P_g$ [24]. The two-player game between the two neural networks, the generator $G$ and the critic $C$, can be formalized as a minimax objective with:

$$\min_G \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{x \sim P_r} [C(x)] - \mathop{\mathbb{E}}_{\tilde{x} \sim P_g} [C(\tilde{x})], \tag{2.1}$$

where $\mathcal{D}$ is the set of Lipschitz-1 functions and $\tilde{x} = G(z)$ a generated sample based on latent random noise $z \sim P_z$. As the critic improves, the generator is forced to gradually enhance its modeling to create more realistic samples, in theory eventually eliminating the difference between generated and real data. To increase the representational power of WGAN, Gradient Penalty was introduced in the WGAN-GP architecture [25]. Within the WGAN-GP architecture, the Lipschitz-1 continuity is softly enforced by a weighted penalty term, represented as ($\lambda_{GP}$), within the loss objective denoted as

$$\begin{aligned} L^{wgan} = & \mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[C(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[C(x)] \\ & + \lambda_{gp} \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} C(\hat{x})||_2 - 1)^2], \end{aligned} \tag{2.2}$$

where $\hat{x} \sim P_{\hat{x}}$ are samples resulting from a linear interpolation between synthetic and real samples and $||\nabla_{\hat{x}} C(\hat{x})||_2$ is the L2 norm of their backpropagated gradients within $C$.

Due to the adversarial nature of GAN training, instabilities in terms of poor convergence occur relatively frequently. Most striking is the phenomenon of mode collapse, where the generator does not exhibit large data variance, but focuses on a single example that is particularly confusing for the critic. Since the initial formulation of GANs, many improvements have been proposed to prevent mode collapse, with minibatch discrimination becoming a best practice [26]. Minibatch discrimination is done by computing the closeness of the independent and identically distributed samples of a minibatch and passing it on to the critic as additional side information. The computation of additional features in an intermediate layer just before the critic has also led to promising successes in other adversarially trained network architectures. For example, finite difference extraction has contributed to automatic error modeling in the interpretation of spectral measurement data [27]. Furthermore, modifying the loss functions for improving the GAN performance has been investigated in many works [28, 29]. E.g., Wang, Sun, and Halgamuge [30] have used a repulsive loss function and thus significantly improve Maximum Mean Discrepancy GAN training and Zhu, Park, Isola, and Efros [31] have developed the *cycle-consistency-loss* for better back and forth translations using CycleGANs.

### 2.2. **Conditional Tabular GAN**

In recent years, the application of GAN models for tabular data synthesis has gained popularity, leading to some state of the art models for tabular data synthesis like TableGan [32] and CTGAN [12]. CTGAN enables tabular data synthesis through a conditional generator, training-by-sampling and mode-specific normalization [12]. To confront data imbalance, a condition, represented as a conditional vector *cond*, is set for one of the categorical columns and the realization in synthetic data is forced by including its fulfillment in the generator loss term. The conditions vary during training which ensures a greater variety of the generated data. Including the deviation from the condition, $L^{cond}$, as an additional penalty term, the loss function of the generator is as follows:

$$L^{ctgan} = L^{wgan} + L^{cond}. \tag{2.3}$$

Inspired by [33], the critic receives multiple samples at once, combined into one *pac*, whereas the *pac* size denotes the number of rows processed simultaneously and is set to 10 by default.

Since its initial publication in 2019, CTGAN has been subject of repeated research and has often been extended or modified. For example, K-Means clustering has been integrated into CTGAN to improve the handling of imbalanced datasets [34]. To improve the differential privacy of synthesized data, CTGAN was combined by noise augmentation and federated training, outperforming other state-of-the-art models [23]. Other research evaluates the performance of CTGAN on a variety of datasets from different domains, identifying both strengths and limitations compared to other methods. For instance, CTGAN achieves solid

results in generating EEG data [35] or datasets for disk failure prediction [36]. Mendikowski and Hartwig improve the ability of CTGAN to autonomously detect non-modifiable relationships in the categorical variables by introducing the (*CV-d*) as performance metric for hyperparameter tuning. Despite some progress, CTGAN has still problems to detect all correct dependencies, especially for column pairs with a large number of categories [8].

### 2.3. **Cramer's V**

CV is a widely used measure that indicates the strength of the correlation between two categorical variables as values between 0 and 1. In a table consisting of $N_r$ rows, CV is derived from Pearson's chi-squared statistic $\chi^2$ [37] for a categorical column pair $(D_i, D_j)$ with number of categories $k = |D_i|$ and $r = |D_j|$ as follows [11]:

$$CV(D_i, D_j) = \sqrt{\frac{\chi^2(D_i, D_j)}{N_r \, min(k-1, r-1)}}. \tag{2.4}$$

In 2013, Bergsma and Wicher formulated a bias-corrected version of CV, particularly useful for larger tables, denoted $\tilde{CV}$ and calculated as follows [38]:

$$\begin{aligned} \tilde{CV}(D_i, D_j) &= \sqrt{\frac{\tilde{\Phi}^2}{min(\tilde{k}-1, \tilde{r}-1)}}, \text{where} \\ \tilde{\Phi}^2 = max(0, \Phi^2 - \frac{(k-1)(r-1)}{(N_r-1)}), \;\; \Phi^2 &= \frac{\chi^2(D_i, D_j)}{N_r}, \\ \tilde{k} = k - \frac{(k-1)^2}{N_r - 1}, \;\text{and}\; \tilde{r} &= r - \frac{(r-1)^2}{N_r - 1}. \end{aligned} \tag{2.5}$$

CV-deviation (*CV-d*) is a measure for synthetic tabular data that combines $\tilde{CV}$ and the root mean squared error. Given a categorical column pair $j \in \mathcal{P}_2(D_1, ..., D_{N_d})$ of a real tabular dataset $T$ and a synthetic replication $T_{syn}$, each with categorical columns $D = \{D_1, ..., D_{N_d}\}$, *CV-d* is determined as the following:

$$CV\text{-}d(T, T_{syn}) = \sqrt{\frac{1}{|\mathcal{P}_2(D)|} \sum_{j \in \mathcal{P}_2(D)} (\tilde{CV}_T(j) - \tilde{CV}_{T_{syn}}(j))^2} \tag{2.6}$$

*CV-d* measures the average deviation of categorical dependencies in the synthetic tabular data from those in the original data and scores the deviation between 0 and 1, with a higher value suggesting a greater difference between the two datasets [8]. Thus, with *CV-d*, we get a powerful similarity measure for categorical columns as there is for continuous columns, e.g. *Pearson* and *Spearman coefficient similarity*.

## 3. **Architecture**

### 3.1. **Cramer's V Integration**

For this work, we introduce two novel approaches of integrating CV within tabular GANs and explain their implementation in detail. Since CV is a measure of the statistical association between two categorical variables [38], we apply it to derive a better preservation of categorical relationships and to improve the overall quality of the synthesised data. While the integration of CV is generally applicable to GANs, we demonstrate it exemplarily on the CTGAN architecture, as illustrated in Figure 1. The two contrasting approaches modify different elements of the GAN framework: first, we add a penalty term to the generator's loss function, and second, we extend the critic's input by extracting a CV matrix for each batch to improve the ability of detecting categorical dependencies.

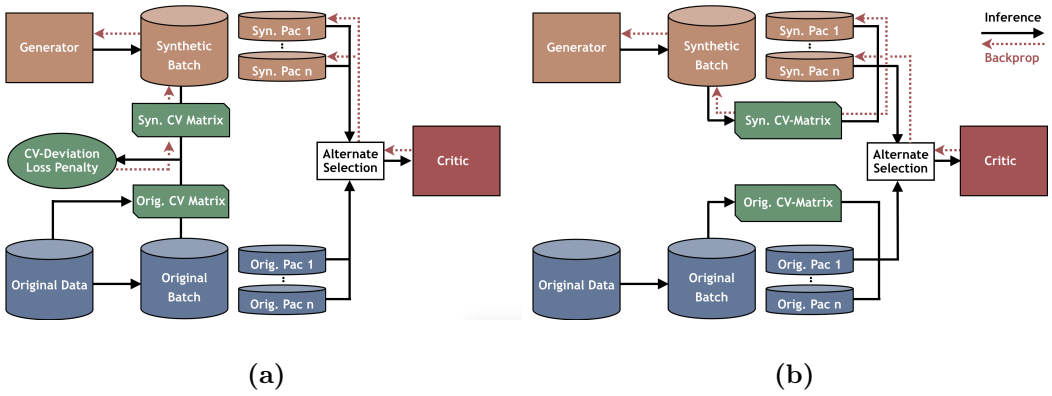


*Figure 1.* Integration of CV into the training regime of tabular GANs. The CTGAN architecture consisting of generator and synthesized data (■), critic (■), and original data (■) has been extended according to the two proposed approaches (■): (a) Regularization of the generator by *CV-d* loss penalty. (b) Backpropagation-aware extraction of the CV matrix as an additional feature for the critic.

#### 3.1.1. **Generator's Loss Function**

For the integration at the generator, we compute the Bergsma and Wicher corrected CV value of each categorical column pair in a synthetic batch and compare it to the corresponding CV values of the original dataset using root mean square error [39] according to *CV-d*[8]. *CV-d* thus provides the average squared deviation of the statistical associations in the categorical columns from the statistical associations in the original dataset. On this basis, we add a weighted *CV-d* penalty to the generator loss function, as described in Equation 3.1 below. By penalizing the deviation of the entire batch from the original dataset, we enforce a greater similarity of the categorical dependencies of synthetic and original data.

Our approach is inspired by the conditional loss [40] of CTGAN, that also penalizes a deviation of the synthetic data from a fixed condition as an addition to the generator loss function, as well as by minibatch discrimination, which adds information about dependencies of a whole batch of synthetic data to the training process [26].

Note that for comparison to the original data we use an entire batch rather than a *pac*, as rare relationships between categorical columns can be better accounted for. There is also a more realistic overall picture of relationships as the number of data rows gets closer to the number of original data, making it also beneficial to calculate *CV-d* on basis of an entire batch.

Since the training success of GAN systems is difficult to determine without the use of domain knowledge, the addition of *CV-d* provides a more humanly understandable condition for the loss function according to which the generator optimizes.

For a CTGAN model $(G, C)$ trained on the training table $T_{train}$, a sample batch vector $b = (\hat{x}_1, \ldots, \hat{x}_m)^T$ is obtained for each generator forward pass. We add $CV\text{-}d(T_{train}, b)$ to the generator's loss function for each batch $b$ generated during training, so that the generator loss objective resembles the following:

$$\begin{aligned} L^{ctgan\text{-}cv} &= L^{wgan} + L^{cond} + L^{cv} \\ \text{where } L^{cv} &= \lambda_{cv} \mathop{\mathbb{E}}_{b \sim P_g^m} [CV\text{-}d(T_{train}, b)] \end{aligned} \tag{3.1}$$

Analogous to gradient penalty, we introduce a *CV-d* penalty weight factor $\lambda_{cv}$.

#### 3.1.2. **Feature Extraction before Critic**

Opposed to the approach of a *CV-d* penalty for the generator, we implement a CV feature extraction approach to strengthen the critic's capability of evaluating categorical dependencies. Here, instead of automated feature extraction, as found e.g. in convolutional neural networks [41], we use an explicit computation for statistical feature extraction. By integrating the CV statistics of the current batch, we provide the critic with additional information that extends the information about the individual entities and provides more insight into the overall structure of the synthesized data, similar to a mini-batch discrimination [26]. We compute the CV matrix for each categorical column pair in each synthetic batch and each training batch. Since the critic works with *pacs* rather than individual samples, we include the flattened CV matrix of the corresponding batch as additional features at the end of each *pac* (original or synthetic) that enters the critic. By using auto-differentiation throughout feature extraction, we enable backpropagation of the associated losses during training. Therefore, this can be seen as a first attempt to improve the detection of categorical dependencies within CTGAN by intervening in the learning gradient provided by the critic.

#### 3.1.3. **Cramer's V Approximation for Backpropagation**

In accordance with the CTGAN implementation, categorical features are represented as one-hot-encoded probabilities during synthesis. To avoid loss of information by transforming these probabilities into discrete values, they are used directly for the approximation of Pearson's chi-squared statistic [37] needed for CV calculation. This is in contrast to discretizing the synthetic output during training but provides an effective backpropagation.

### 3.2. **Implementation**

For our implementation, we use Python 3.7 and the Synthetic Data Vault (SDV) library[1], which provides open-source software systems for synthetic data generation and was initiated in 2018 as a project of the Massachusetts Institute of Technology. Part of our evaluation structure and CTGAN models are based on version 0.9.0 of the SDV library, which provides a CTGAN implementation of the original paper [12] based on the *PyTorch* framework. We train multiple CTGAN models for each of the three benchmark datasets, using different variants of our methods as well as a baseline following the original CTGAN paper. Each model is used to create a synthetic dataset that matches the size of the original dataset.

## 4. **Experiments**

### 4.1. **Setup**

In this section, we present first experimental results from the evaluation of our proposed methods of integrating CV statistics into tabular data synthesis with GANs. For easy comparison, we adopt the model and training hyperparameters from the original CTGAN paper [12]. In addition to a baseline without modifications, we train models according to section 3.1.1 with different *CV-d* weights $\lambda_{cv} \in \{1, 10, 20\}$, as well as trial incorporating the feature extraction according to section 3.1.2. We run these experiments on three public benchmark datasets containing personal data from different domains in which data sharing and machine learning applications play a major role. The three datasets are specified in Table 1. Columns with duplicate information such as IDs and rows with missing entries were deleted during data preprocessing. All experiments were performed on a Quadro RTX 4000 graphics card.

[1] https://sdv.dev/

| Dataset | adult | superstore | cancer |
|---|---|---|---|
| # of rows | 32561 | 9994 | 858 |
| # of columns | 15 | 17 | 36 |
| # of categorical columns | 13 | 10 | 27 |
| domain | social | e-commerce | medical |
| source | UCI[2] | Kaggle[3] | UCI[4] |

*Table 1.* Basic overview of the datasets used for the experimental evaluation.

### 4.2. **Metrics**

We evaluate each of the synthetic datasets using two different metrics. We use machine learning performance as a key metric, as it corresponds to the most common use case of synthetic data. Furthermore, we evaluate the similarity of the synthetic to the original data in terms of associations between categorical columns, since our modifications pay particular attention to such relations:

(1) **Supervised Learning Performance:** To simulate a real-world use case, we compare the supervised learning performance of the synthesized datasets. We train various classifier models on each synthetic dataset and subsequently test these models on original data. In the case of the *cancer* and *adult* dataset we use a binary classifier to predict cervical cancer and income class, for the *superstore* dataset we predict one of three possible shipping modes. An identical model trained only on original data serves as a general quality baseline. Within our evaluation we use the following classifiers:
- *Decision Tree* with unlimited tree depths and splitting according to the gini impurity criterion.
- *Random Forest* ensemble consisting of 300 decision trees, each configured as specified above.
- *Logistic Regression* incorporating L2 regularization and y-axis intercept adjustment.
- *Gaussian Naive Bayes* with non-fixed prior distributions and variance smoothing term of $10^{-9}$ added to stabilize the computation.
- *Multilayer Perceptron* with one hidden layer consisting of 100 neurons and relu-activation, trained using adam optimizer with an initial learning rate of 0.001 and regularization parameters $\alpha = 0.0001$, $\beta_1 = 0.9$ and, $\beta_2 = 0.999$.

In line with Fang, Dhami, and Kersting [23], we aggregate the results of all different supervised learning classifiers, in our case by averaging all model accuracy-scores.

(2) **Contingency Similarity:** We compute the similarity of all pairs of categorical columns between the real and synthetic datasets in accordance to the SDV contingency similarity and calculate an overall score using the mean. The contingency similarity ranges from 0 to 1, with a higher value suggesting greater similarity. The calculation of contingency similarity $CS$ of categorical columns $D_i$ and $D_j$ with all possible categories $A$ and $B$ is shown in the following equation 4.1. We define $S_{a,b}$ and $R_{a,b}$ as the synthetic and real frequencies of categories $a$ and $b$.

$$CS(D_i, D_j) = 1 - \frac{1}{2} \sum_{a \in A} \sum_{b \in B} |S_{a,b} - R_{a,b}| \tag{4.1}$$

[2]https://archive.ics.uci.edu/ml/datasets/adult
[3]https://www.kaggle.com/datasets/vivek468/superstore-dataset-final
[4]https://archive.ics.uci.edu/ml/datasets/Cervical+cancer+%28Risk+Factors%29

### 4.3. **Evaluation**

| **Dataset** | $\lambda_{cv}$ | **Ft Extr** | **Avrg Acc** | **Contingency Similarity** |
|---|---|---|---|---|
| cancer original | 0 | False | 0.937 | 1 |
| cancer synthetic this paper | 10 | False | 0.912 | 0.862 |
| cancer synthetic this paper | 1 | False | 0.881 | 0.851 |
| cancer synthetic this paper | 20 | False | 0.880 | **0.865** |
| cancer synthetic Xu et al. | 0 | False | 0.872 | 0.862 |
| cancer synthetic this paper | 0 | True | 0.852 | 0.816 |
| adult original | 0 | False | 0.818 | 1 |
| adult synthetic this paper | 10 | False | 0.794 | 0.827 |
| adult synthetic this paper | 1 | False | 0.790 | 0.836 |
| adult synthetic Xu et al. | 0 | False | 0.786 | 0.828 |
| adult synthetic this paper | 20 | False | 0.783 | **0.844** |
| adult synthetic this paper | 0 | True | 0.777 | 0.813 |
| superstore original | 0 | False | 0.715 | 1 |
| superstore synthetic this paper | 10 | False | 0.524 | **0361** |
| superstore synthetic this paper | 20 | False | 0.451 | 0.358 |
| superstore synthetic this paper | 0 | True | 0.431 | 0.304 |
| superstore synthetic Xu et al. | 0 | False | 0.416 | 0.348 |
| superstore synthetic this paper | 1 | False | 0.413 | **0.361** |

*Table 2.* Experiments along with baselines ordered by average supervised learning accuracy, highest contingency similarity of each domain is highlighted.

Table 2 shows the supervised learning performance and contingency similarity of our methods compared to the original CTGAN. As expected, we achieve lower accuracy scores on all synthetic datasets compared to the model trained and tested on the original data.

For all three datasets, adding a 10 times *CV-d* to the generator loss results in the best supervised learning performance (see Table 2). Considering the *superstore* dataset, we are able to increase the supervised learning performance by about 26 % compared to the original CTGAN. Measured over all datasets, the average percentage increase by using *CV-d* with $\lambda_{cv} = 10$ is about 11 %. Across all three datasets, the original CTGAN baseline is outperformed by at least one additional trial with *CV-d* integrated into the generator loss. In contrast, the feature extraction approach leads to a better supervised learning result only for the *superstore* dataset. Table 3 shows the individual results of all machine learning classifiers. The improvements in supervised learning performance are most pronounced in Decision Trees (up to 36%), Multilayer Perceptrons (up to 37%), and Gaussian Naive Bayes (up to 39%), while no significant improvements are seen for Logistic Regression and Random Forests.

Looking at the contingency similarity in Table 2, we also notice that at least one of our approaches outperforms the original CTGAN for each dataset, although the differences between the individual trials are rather moderate. In most cases, approaches with a higher contingency similarity also show a higher supervised learning performance. Similar to the results in supervised learning performance, mostly approaches with integrated *CV-d* exceed the contingency similarity of the original CTGAN model.

In line with Mendikowski and Hartwig [8], we also examine the synthetic datasets in terms of categorical integrity, i.e., maintaining relationships between categorical columns that do not allow new combinations in synthetic data. While these relationships are already addressed in the contingency similarity, the categorical integrity provides a sharper focus on data rows that are clearly erroneous. The categorical integrity reflects the percentage of correct assignments and is thus assigned values from 0 to 1. By using our approaches, we are able to improve the categorical integrity from 0.63 to 0.80 (feature extraction) or 0.64 ($\lambda_{cv}$ of 10) for the cancer dataset and from 0.93 to 0.95 (feature extraction) for the adult

dataset compared to the original CTGAN. The impact of our approaches on categorical integrity during the training process of the cancer dataset is illustrated in Figure 2. The integration of feature extraction at the critic causes a nearly continuous improvement of the preservation of categorical integrity, whereas the addition of a *CV-d* penalty at the generator only results in a moderate enhancement. When looking at the representation of rare cases in the synthetic data we do not see significant differences to the original CTGAN. Rare cases are slightly reduced, but due to strategies of CTGAN, such as training by sampling and the conditional loss, these reductions are limited. We also investigate a combination of feature extraction and best *CV-d* approach, but this does not result in better performance than the single *CV-d*.

## 5. **Discussion**

In general, we can state that we outperform the original Xu et al. model [12] in every category with at least one of our modifications. Especially for supervised learning performance, which provides a good indicator for the overall quality of synthetic data, our approaches lead to significantly better results. In terms of contingency similarity, our extensions show the potential to address compliance with categorical dependencies in synthetic data, with many approaches achieving a moderately higher score than the original CTGAN. Note that all our extensions are implemented with the original CTGAN hyperparameters and thus still offer further improvement possibilities by, for example, targeted hyperparameter training. However, a first comparison of our modifications based on the original CTGAN is important to show the general potential of the proposed methods.

Our experiments show that the integration of *CV-d* seems very beneficial to the data synthesis process, especially by adding a 10 times *CV-d* penalty, which led to an average increase of 11 % in supervised learning performance. The integration of feature extraction seems to implement the more simple conditions of categorical integrity well, but generally leads to minor to no overall improvement in supervised learning performance in comparison to the original CTGAN. The smaller impact of feature extraction could be due to a degenerated learning gradient or to the simple structure of the critic network, which consists of only two hidden layers. It may be necessary to adapt the architecture to handle more complex inputs, like those generated by the feature extraction approach. Furthermore, in the

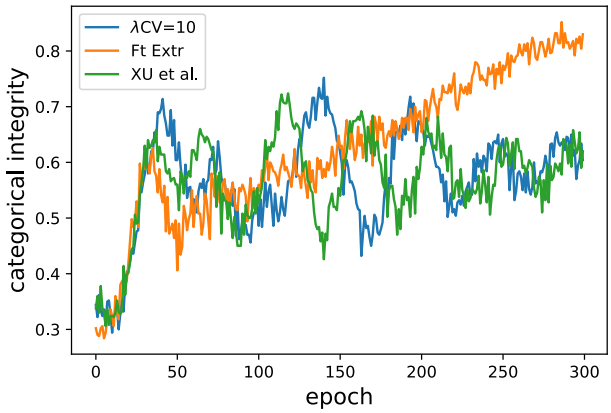


*Figure 2.* Categorical integrity of the *cancer* dataset synthesized by CTGAN during training with 10 times *CV-d* loss penalty (■), integrated CV feature extraction (■), and the Xu et al. baseline (■).

*CV-d* approach we penalize the direct deviation from the original data, which, in contrast to feature extraction, provides a clear optimization direction.

## 6. Conclusion & Outlook

In this paper, we have incorporated the CV statistics into the CTGAN architecture, providing additional information about the statistical dependencies between categorical columns to the direct training process. We have implemented two different approaches: First, we have introduced the *CV-d* to the original data as an additional loss term at the generator. Second, we have extracted the CV matrix as an additional feature vector for the critic's input. Both approaches combine single data units with the overall categorical dependencies in an entire batch of data. Our experiments show that especially the first adjustment is beneficial to the data synthesis process which led to an average increase of 11 % in supervised learning performance.

Overall this paper made an important contribution to improving the CTGAN architecture which could impact GAN networks for tabular synthesis in general, since our approach is transferable and can also be realized with statistical metrics other than CV. By providing additional information on entire batches of data, we adapt the idea of minibatch discrimination already widely used in GANs specific to tabular data synthesis. Even though our approach has led to an improvement in the quality of the synthetic data, contingency similarity as well as the supervised learning performance of all synthetic data remains in most cases significantly below that of the original data, research in the field of table synthesis with GAN networks remains promising.

Future work should apply both presented approaches to a larger variety of different GAN architectures and evaluate their results using similar metrics, with particular interest in their impact on GAN architectures incorporating differential privacy. Similar to our approach other statistical functions can also become part of the direct training process of CTGAN to increase the quality of the synthetic data and make the training process more comprehensible.

## Appendix A. Recorded Supervised Learning Model Accuracies

| | | | Accuracy | | | | |
|---|---|---|---|---|---|---|---|
| **Dataset** | $\lambda_{cv}$ | **Ft Extr** | **DTree** | **RF** | **LGR** | **NB** | **MLP** |
| cancer synthetic this paper | 10 | False | 0.952 | 0.978 | 0.974 | 0.686 | 0.969 |
| cancer synthetic this paper | 0 | True | 0.948 | 0.978 | 0.974 | 0.402 | 0.956 |
| cancer synthetic this paper | 1 | False | 0.948 | 0.978 | 0.978 | 0.537 | 0.965 |
| cancer synthetic this paper | 20 | False | 0.934 | 0.978 | 0.978 | 0.55 | 0.956 |
| cancer synthetic Xu et al. | 0 | False | 0.913 | 0.978 | 0.978 | 0.555 | 0.0934 |
| adult synthetic this paper | 10 | False | 0.774 | 0.815 | 0.787 | 0.806 | 0.79 |
| adult synthetic this paper | 1 | False | 0.772 | 0.83 | 0.791 | 0.795 | 0.763 |
| adult synthetic this paper | 0 | True | 0.749 | 0.802 | 0.776 | 0.798 | 0.761 |
| adult synthetic this paper | 20 | False | 0.743 | 0.803 | 0.793 | 0.789 | 0.788 |
| adult synthetic Xu et al. | 0 | False | 0.741 | 0.832 | 0.798 | 0.8 | 0.761 |
| superstore synthetic this paper | 10 | False | 0.497 | 0.564 | 0.597 | 0.466 | 0.497 |
| superstore synthetic this paper | 20 | False | 0.392 | 0.547 | 0.576 | 0.341 | 0.399 |
| superstore synthetic this paper | 1 | False | 0.384 | 0.479 | 0.506 | 0.325 | 0.369 |
| superstore synthetic Xu et al. | 0 | False | 0.363 | 0.491 | 0.526 | 0.336 | 0.362 |
| superstore synthetic this paper | 0 | True | 0.325 | 0.525 | 0.442 | 0.457 | 0.408 |

*Table 3.* Accuracies of the supervised learning models ordered by decision tree accuracy.

## References


[1] S. I. Nikolenko. *Synthetic data for deep learning*. Vol. 174. Springer, 2021.

[2] H. I. Maseeh, C. Jebarajakirthy, R. Pentecost, D. Arli, S. Weaven, and M. Ashaduzzaman. "Privacy concerns in e-commerce: A multilevel meta-analysis". In: *Psychology & Marketing* 38.10 (2021), pp. 1779–1798.

[3] C. Thapa and S. Camtepe. "Precision health data: Requirements, challenges and existing techniques for data security and privacy". In: *Computers in biology and medicine* 129 (2021), p. 104130.

[4] A. Figueira and B. Vaz. "Survey on synthetic data generation, evaluation methods and GANs". In: *Mathematics* 10.15 (2022), p. 2733.

[5] S. M. Bellovin, P. K. Dutta, and N. Reitinger. "Privacy and synthetic datasets". In: *Stan. Tech. L. Rev.* 22 (2019), p. 1.

[6] S. James, C. Harbron, J. Branson, and M. Sundler. "Synthetic data use: exploring use cases to optimise data utility". In: *Discover Artificial Intelligence* 1.1 (2021), pp. 1–13.

[7] R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. Williamson, and F. Mahmood. "Synthetic data in machine learning for medicine and healthcare". In: *Nature Biomedical Engineering* 5.6 (2021), pp. 493–497.

[8] M. Mendikowski and M. Hartwig. "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, 2022, pp. 91–105.

[9] H. Thanh-Tung and T. Tran. "Catastrophic forgetting and mode collapse in gans". In: *2020 international joint conference on neural networks (ijcnn)*. IEEE. 2020, pp. 1–10.

[10] Y. Zhang, N. A. Zaidi, J. Zhou, and G. Li. "GANBLR: a tabular data generation model". In: *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2021, pp. 181–190.

[11] A. C. Acock and G. R. Stavig. "A measure of association for nonparametric statistics". In: *Social Forces* 57.4 (1979), pp. 1381–1386.

[12] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni. "Modeling tabular data using conditional gan". In: *Advances in Neural Information Processing Systems* 32 (2019).

[13] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks. "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models". In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[14] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[15] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2014.

[16] D. Rezende and S. Mohamed. "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems* 3 (June 2014).

[18] L. A. Gatys, A. S. Ecker, and M. Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[19] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. "Generative adversarial text to image synthesis". In: *International conference on machine learning*. PMLR. 2016, pp. 1060–1069.

[20] T. Karras, S. Laine, and T. Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (Jan. 2020), pp. 1–1.

[21] J. Nistal, S. Lattner, and G. Richard. "Comparing representations for audio synthesis using generative adversarial networks". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE. 2021, pp. 161–165.

[22] S. Bourou, A. El Saer, T.-H. Velivassaki, A. Voulkidis, and T. Zahariadis. "A review of tabular data synthesis using gans on an ids dataset". In: *Information* 12.09 (2021), p. 375.

[23] M. L. Fang, D. S. Dhami, and K. Kersting. "DP-CTGAN: Differentially Private Medical Data Generation Using CTGANs". In: *Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, Proceedings*. Halifax, NS, Canada: Springer-Verlag, 2022, 178–188.

[24] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

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

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016).

[27] B. Schindler, D. Günzel, and T. Schmid. "Neural Noise Module: Automated Error Modeling using Adversarial Neural Networks". In: *International Conference on Bioelectromagnetism, Electrical Bioimpedance, and Electrical Impedance Tomography 2022 (ICBEM-ICEBI-EIT 2022), Seoul, South Korea.* 2022, pp. 191–194.

[28] S. Nowozin, B. Cseke, and R. Tomioka. "f-gan: Training generative neural samplers using variational divergence minimization". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, pp. 271–279.

[29] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. "Stabilizing training of generative adversarial networks through regularization". In: *Advances in neural information processing systems* 30 (2017).

[30] W. Wang, Y. Sun, and S. Halgamuge. "Improving MMD-GAN training with repulsive loss function". In: *arXiv preprint arXiv:1812.09916* (2018).

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

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

[33] Z. Lin, A. Khetan, G. Fanti, and S. Oh. "Pacgan: The power of two samples in generative adversarial networks". In: *Advances in neural information processing systems* 31 (2018).

[34] C. An, J. Sun, Y. Wang, and Q. Wei. "A K-means Improved CTGAN Oversampling Method for Data Imbalance Problem". In: *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*. IEEE. 2021, pp. 883–887.

[35] J. S. Lee and O. Lee. "CTGAN VS TGAN? Which one is more suitable for generating synthetic EEG data". In: *Journal of Theoretical and Applied Information Technology* 99.10 (2021).

[36] J. Jia, P. Wu, K. Zhang, and J. Zhong. "Imbalanced Disk Failure Data Processing Method Based on CTGAN". In: *International Conference on Intelligent Computing*. Springer. 2022, pp. 638–649.

[37] G. G. Koch and L. E. Wiener. "Chi-Squared Tests: Basics". In: *Wiley StatsRef: Statistics Reference Online* (2014), pp. 1–20.

[38] W. Bergsma. "A bias-correction for Cramér's V and Tschuprow's T". In: *Journal of the Korean Statistical Society* 42.3 (2013), pp. 323–328.

[39] T. Chai and R. R. Draxler. "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature". In: *Geoscientific model development* 7.3 (2014), pp. 1247–1250.

[40] M. Mirza and S. Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[41] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi. "A survey of the recent architectures of deep convolutional neural networks". In: *Artificial Intelligence Review* 53.8 (2020), pp. 5455–5516.