CAUSAL-TGAN: CAUSALLY-AWARE TABULAR DATA GENERATIVE ADVERSARIAL NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic tabular data generation has recently gained immense attention due to applications in medicine, finance, and other fields. Generative adversarial networks (GANs) designed initially for image generation have been demonstrated to be promising for generating certain types of tabular data. Tabular data may contain mixed data types such as continuous, ordered, binary, and categorical values. However, the causal relationships between the variables in tabular data have been largely ignored by the prior art. Causality encodes real-world relationships occurring naturally between variables measuring a phenomenon.

In this work, we propose Causal-TGAN, a data generation architecture that incorporates causal relationships at its core. The flexibility of this architecture is its capability to support different types of expert knowledge (e.g., complete or partial) about the causal nature of the underlying phenomenon. Extensive experimental results on both simulated and real-world datasets demonstrate that Causal-TGAN and its hybrid avatars consistently outperform other baseline GAN models. We also argue that the architecture's flexibility is promising for many practical applications.

1 INTRODUCTION

Synthetic data generation methods have drawn tremendous attention since the invention of generative adversarial networks (GAN). In addition to the original intent of generating realistic images that do not exist in the real world, synthetic data generation is now used to achieve a wide variety of objectives such as data augmentation (Odena et al., 2017; Antoniou et al., 2017; Mariani et al., 2018; Chatziagapi et al., 2019), missing values imputation (Wang et al., 2019; Shin et al., 2020), counterfactual example generation (Neal et al. (2018); Madaan et al. (2020); Sauer & Geiger (2021)), and interpretable artificial intelligence (Chang et al., 2018; Genovese et al., 2019; Kenny & Keane, 2020).

As generative models have demonstrated remarkable results on unstructured image and text data generation, (structured) tabular dataset generation is emerging as a prominent research problem (Chen et al., 2019; Yoon et al., 2018; Srinivasan et al., 2019). This is of great interest because tabular datasets frequently occur in areas such as medicine and finance. One key difference in tabular datasets, compared with image and text data, is that they contain mixed data types such as continuous, ordered, binary, and categorical values.

Most tabular datasets are structured where each column (variable) within the data table has a physical meaning, such as age or income. Though there are innumerable such variables representing different meanings or measurements, it is fair to assume that variables within the same tabular dataset are statistically related. This is because tabular data are usually collected to measure the attributes of an object or a phenomenon. Causal factors among variables produce statistical relationships between the collected data. For example, there could be three possible causal relations between variables A and B, i.e., A causes B, B causes A, or another variable C causes both A and B. In this work we show that exploiting these causal relations in deep generative models delivers synthesized data that more accurately (compared to the state-of-the-art) capture the target data distribution.

In this paper, we propose a causally-aware tabular data generative adversarial network (**Causal-TGAN**). Causal-TGAN employs a new framework for the generator that models inter-variable causal relations. The generator is constructed as a structural causal model (SCM) to capture multiple causal

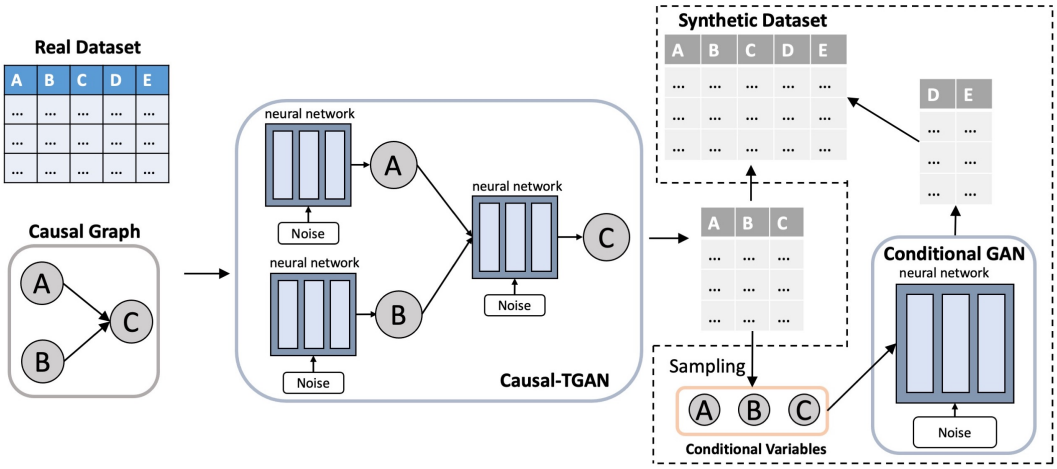


Figure 1: **An example the proposed Causal-TGAN with expert knowledge:** Given a dataset of five columns $A - E$ and the **partial knowledge** that A and B are the causes of C, the proposed Causal-TGAN constructs a generator following the causal structure of $A \rightarrow C \leftarrow B$. By doing this, the Causal-TGAN can generate a sub-table that only contains columns A, B, and C. We employ a conditional GAN to generate the rest of the columns D and E, conditioned on the samples from the Causal-TGAN. Finally, we concatenate the output of the conditional GAN (D and E) with the corresponding conditions (A, B, and C) to form the complete synthetic table. For Causal-TGAN operating with **full knowledge** (all causal relationships are known), the entire dataset can be generated without the use of another conditional GAN. The part enclosed in dashed lines will not be necessary as all variables can be generated with the proposed Causal-TGAN.

processes. Completeness of the causal relationship information provided to the Causal-TGAN is proportional to the extent of available expert knowledge. An example of Causal-TGAN working with expert knowledge is illustrated in Figure 1. The availability of accurate and complete knowledge (i.e., causal relations known for all variables) sometimes can be difficult. Therefore, we propose a hybrid generative mechanism for data generation using only partial knowledge (i.e., causal relations known only for some variables). In the partial knowledge setting, Causal-TGAN generates data for the variables with known causal information while an auxiliary conditional GAN is developed for generating data for the other variables. We also suitably modify the conventional conditional GANs for application to tabular datasets.

We summarize our contributions as follows:

- Causal-TGAN, a novel GAN framework, that can incorporate (complete and incomplete) inter-variable causal relationships from a domain expert.
- Detailed experimental results demonstrating that Causal-TGAN is better (compared to other methods) at capturing the target data distribution on both simulated and real-world datasets.
- A comprehensive evaluation of Causal-TGAN's performance for different degrees of knowledge about inter-variable causal relationships. Demonstration of the practicality of Causal-TGAN even in the absence of complete expert knowledge.

The paper is structured as follows. In Section 2, we discuss related synthetic tabular data generation methods. We outline preliminary information on SCM and generative adversarial networks in Section 3. In Section 4, we describe the proposed Causal-TGAN framework and the hybrid generative mechanism. We present the experimental setting and evaluation of Causal-TGAN on both simulated and real-world datasets in Section 5.

2 Related Work

Prior work has exploited GANs for tabular data generation. MedGAN (Choi et al., 2017) leverages a combination of GAN and autoencoder to generate high-dimensional discrete variables in elec-

tronic health record data. TableGAN (Park et al., 2018) guides the generator to generate synthetic samples with a reasonable label by adding an auxiliary classifier. PATE-GAN (Jordon et al., 2019) generates synthetic data with differential privacy guarantees. CorGAN (Torfi et al., 2020) uses a one-dimensional convolutional GAN architecture for both the generator and the discriminator to capture the correlation between variables in the feature space. CTGAN (Xu et al., 2019) addresses class imbalance by introducing a conditional generator that provides the model with the capacity to evenly re-sampling all categories from discrete columns during the training process. In addition, CTGAN employs a Gaussian mixture model for multimodal distributed data in the table's columns. OCT-GAN (Kim et al., 2021) deploys a discriminator with an ODE (ordinary differential equation) layer to perform a trajectory-based classification to improve the generator's performance. A conditional vector that efficiently encodes mixed-type data to solve the imbalanced data and long-tail distribution (e.g., a variable with a sparse distribution over a long-range) is considered in CTAB-GAN (Zhao et al., 2021).

Two papers closely related to this work are CGNN (Goudet et al., 2018) and CausalGAN (Kocaoglu et al., 2017). Both of these constructs SCM-based generators but with different objectives. CGNN discovers causal structures with the argument that a generator built with accurate causal graph knowledge can learn the data distribution better. Causal-GAN leverages a causality-based generator to sample interventional labels for generating facial images such as women with the mustache. The data label type considered in CausalGAN is only binary to indicate if an image has an attribute or not. Moreover, the main objective of CausalGAN is out-of-distribution image generation. The causality-based generator is only one component for sampling labels. In contrast, our Causal-TGAN design is proposed to generate datasets with different variable types.

3 BACKGROUND

3.1 STRUCTURAL CAUSAL MODELS

We briefly introduce the concept of the structural causal models (SCM) (Pearl, 2009). SCM constitutes a causal system with a set of variables, a causal graph encoding causal relations between variables, and a set of equations describing how each variable is generated based on its causes (represented by parent nodes in the causal graph).

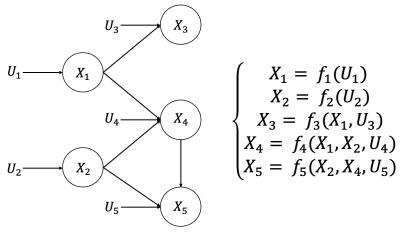


Figure 2: **An example of SCM:** The causal graph and the corresponding causal mechanisms are shown.

Mathematically, an SCM $\mathcal{M}_\mathcal{G}$ with a causal graph $\mathcal{G}$ can be represented by a triplet $\mathcal{M}_\mathcal{G} = \langle \mathcal{X}, \mathcal{F}, \mathcal{U} \rangle$ that contains a set of endogenous variables $\mathcal{X} = \{X_1, X_2, ..., X_d\}$, a set of causal equations (mechanisms) $\mathcal{F} = \{f_1, f_2, ..., f_d\}$ and a set of exogenous variables $U = \{U_1, U_2, ..., U_d\}$, where each U_i is independently from a distribution $\mathcal{U}$. The causal relationship "X_i causes X_j" is represented in the causal graph by a directed edge that orientates from X_i to X_j, i.e., $X_i \to X_j$. The value of X_j is determined by its causal equation $X_j = f_j(Pa_\mathcal{G}(X_j), U_j)$ where $Pa_\mathcal{G}(X_j)$ denotes all the parent nodes of X_j in $\mathcal{G}$. U_j is the exogenous variable of X_j and can be seen as the cumulative effect of all unobserved causes of X_j. Figure 2 illustrates an example of SCM with 5 variables.

3.2 GENERATIVE ADVERSARIAL NETWORKS (GAN)

GANs consist of two components: a generator G and a discriminator D. The purpose of G is to describe the data distribution by mapping a low-dimensional normal distribution to a high-dimension data distribution. D discriminates a sample from G as real or generated. G and D are trained alternatively following the two-player min-max game:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))] \tag{1}$$

where $z \in \mathbb{R}^{d_z}$ is a latent variable sampled from distribution $p(z)$ and $p_{\text{data}}(x)$ is the target data distribution. The generator aims to improve the generation quality by confusing the discriminator D. After the generator is improved the discriminator's ability is improved. The generator and the

discriminator repeatedly compete with each other until they reach a Nash Equilibrium (Goodfellow et al., 2014).

A variant of GAN is the conditional GAN (CGAN) (Mirza & Osindero, 2014). CGAN takes additional information as input and extends GANs to a conditional model. By conditioning on the "information", CGAN can generate data with the attributes described by the "information". The two-player minimax game of CGAN with a conditional variable y is defined as follow:

$$\min_G \max_D \mathbb{E}_{x,y\sim p_{\text{data}}(x,y)}[\log D(x,y)] + \mathbb{E}_{z\sim p(z),y\sim p_{\text{data}}(y)}[\log(1 - D(G(z,y),y))] \tag{2}$$

4 METHOD

4.1 SCM-BASED GENERATOR

In this section, we describe the steps to construct a Causal-TGAN for tabular data T consisting of a set of n variables $\mathcal{V} = \{V_1, V_2, ..., V_n\}$. Each variable corresponds to one column in T. We use the lower case v_i to denote a value of variable V_i. Hence a data sample is represented as a vector $(v_1, v_2, ..., v_n)$. We also have a causal graph $\mathcal{G}$ that encodes the causal relationships among V. The objective of Causal-TGAN is to estimate the empirical data distribution $P_T(\mathcal{V})$ of T by incorporating expert knowledge encoded in $\mathcal{G}$.

4.1.1 VARIABLE ENCODING

Tabular GAN training can benefit from the proper encoding of variables (Xu et al., 2019). Therefore, we consider discrete and continuous variable-specific encoding. It is natural to encode discrete variables as one-hot vectors. We use the mode-specific normalization proposed by (Xu et al., 2019) for encoding continuous variables. The mode-specific normalization first fits a variational Gaussian mixture model (VGM) for the column containing continuous variables. Let us assume that the VGM consists of n Gaussian components. Then, a single value from this column can be encoded as a vector of length $n + 1$. The first n elements denote a one-hot vector indicating the most likely Gaussian component that the value belongs to. The last (i.e., $n + 1$st) element is the mean and variance-normalized value of the corresponding Gaussian component.

4.1.2 GENERATOR CONSTRUCTION

Causal-TGAN is a generator that is built in an SCM fashion. Unlike conventional GANs that use only one generator to map low-dimensional noise vectors to the high dimensional data samples, Causal-TGAN employs multiple such generators, called sub-generators. Each sub-generator is responsible only for generating values for one variable. Therefore each sub-generator is associated with only one variable and can interpreted as the causal equation for that variable. These are the endogenous variables of an SCM. The noise vector inputs to the sub-generators are the exogenous variables. Connecting the sub-generators following the causal relationship described in the causal diagram $\mathcal{G}$ produces the Causal-TGAN as an SCM-based generator.

Formally, for the tabular dataset T, we construct the Causal-TGAN, $\mathcal{F}$, which is a set of sub-generators $\mathcal{F} = \{f_{V_1}, f_{V_2}, ..., f_{V_n}\}$, where the sub-generator f_{V_i} is a neural network corresponding to the causal equation for variable V_i. To causally connect two sub-generators f_{V_i} and f_{V_j} for two arbitrary variables V_i and V_j, if V_i is the parent node of V_j, we connect V_i and V_j by taking the output of f_{V_i} as the input to f_{V_j}, otherwise do nothing.

4.1.3 DATA SAMPLING

In Causal-TGAN, one complete data sample is generated following the topological order of the causal diagram. That is, the variables at the root position (i.e., variables without a parent node) are generated first, followed by generating their children, and then follow this rule until all leaf nodes (i.e., variables without child nodes) are generated. We summarize the steps to generate one sample by Causal-TGAN as follows:

1. Sample a set of noise vectors, $\mu = \{u_1, u_2, ..., u_n\}$, independently and identically from a distribution $\mathcal{U}$. In our case, $\mathcal{U}$ is the Gaussian distribution.

2. Following the topological order, generate each variable V_i by $v_i = f_i(u_i, Pa_{\mathcal{G}}(X_i))$, where $Pa_{\mathcal{G}}(X_i)$ are the values of parent nodes of V_i described in causal diagram $\mathcal{G}$.
3. If the topological order is not consistent with the order of the variables in the dataset then reorder the generated samples according to the order in the dataset.

Similar to regular GANs, Causal-TGAN can generates data samples by only taking noise as input. For simplicity, we represent the Causal-TGAN data generation process described above as a function $G_{\mathcal{F}}(\cdot)$ that maps noise μ to a data sample x_S:

$$x_S := G_{\mathcal{F}}(\mu) \tag{3}$$

4.2 Data Generation using Partial Knowledge

Causal-TGAN can consume only partial knowledge (i.e., causal diagram known only partially for a few variables) for data generation when complete knowledge is not accessible, or domain experts only have high confidence only in partial knowledge. There are several practical use cases (e.g., medicine) where this is applicable. To do this, we first fit Causal-TGAN on a subset of the target dataset containing only the variables with known causal relations. Then we leverage a conditional GAN to generate the rest of the variables (i.e., variables without known causal relations) conditioned on the variables used by Causal-TGAN.

Formally, continuing with the notations defined above, we have a tabular dataset T with variables $\mathcal{V}$. We know the causal diagram $\mathcal{G}_{\mathcal{K}_c}$ for a set of variables $\mathcal{K}_c$, and $\mathcal{K}_c \subset \mathcal{V}$. Then the rest variables can be represented as $\mathcal{V} \setminus \mathcal{K}_c$. We use $T(\mathcal{K}_c)$ and $T(\mathcal{V} \setminus \mathcal{K}_c)$ to denote the dataset with and without variables of known causal relations, respectively. To operate under partial knowledge, we first fit a Causal-TGAN on $T(\mathcal{K}_c)$ with a known causal diagram $\mathcal{G}_{\mathcal{K}_c}$. Then we use a conditional generator C_{cond} to generate one sample of the rest of the variables x_S^{cond} by conditioning on the sample x_S^{causal} generated by Causal-TGAN:

$$x_S^{cond} = C_{cond}(e, x_S^{causal}) \tag{4}$$

where e is the noise vector. Then, under the partial knowledge setting a complete data sample is generated as follow:

$$x_S = order_T(x_S^{cond} \oplus x_S^{causal}) \tag{5}$$

where $\oplus$ is the operation of vector concatenation and $order_T(\cdot)$ is the operation that reorders the values according to the order of the variables in T.

4.3 Training Causal-TGAN

Either the Causal-TGAN or the Conditional GAN is trained using WGAN loss with gradient penalty (Gulrajani et al., 2017):

$$\min_G \max_D V(G, D) = \mathbb{E}_{x_R \sim P_T(\mathcal{V})}[D(x_R)] - \mathbb{E}_{z \sim p_z}[D(G(z))] - \lambda \mathbb{E}_{\hat{x} \sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2] \tag{6}$$

where G and D are the generator and the discriminator. p_z is the distribution of the noise vector. $\hat{x}$ is a randomly weighted combination of the synthetic sample $G(z)$ and the true sample x_R. Mathematically, we randomly sample a weighting vector $\boldsymbol{\lambda} \in [0, 1]^n$, where n is the sample dimension. Then, $\hat{x}$ can be calculated by $\hat{x} = \boldsymbol{\lambda} \circ G(z) + (1 - \boldsymbol{\lambda}) \circ x_R$, where $\circ$ denotes the Hadamard product. Specifically, for training a Causal-TGAN, G represents $G_{\mathcal{F}}$ defined in 3. For training the conditional GAN, p_z is the joint distribution of μ and e. G describes the generating process of conditional GAN (defined in Eq.5) as:

$$G(z) = order_T(C_{cond}(e, G_{\mathcal{F}}(\mu)) \oplus G_{\mathcal{F}}(\mu)) \tag{7}$$

During the conditional GAN training, we only update the parameters of C_{cond}, and the parameters of $G_{\mathcal{F}}$ are kept unchanged as the Causal-TGAN is pre-trained on the sub-table with known causal relations.

5 Experimental Results

We introduce the datasets and baseline models in Section 5.1, and the evaluation metrics in Section 5.2. In Section 5.3, we present the performance of Causal-TGAN on simulated datasets under different expert knowledge conditions. Results on real-world datasets are presented in Section 5.4.

5.1 EXPERIMENTAL SETTING

We use Bayesian networks, Gaussian Bayesian networks, and conditional linear Gaussian Bayesian networks to simulate three types of simulated datasets that are discrete-only, continuous-only, and of mixed-type, respectively. We used the graph structures of the Bayesian networks to represent true causal relations. We utilized UCI (Dua & Graff, 2017) and Kaggle datasets representing the real-world. The Greedy Equivalence Search causal discovery (Glymour et al., 2019) technique was applied to real-world datasets to estimate the causal graphs. Details about these datasets are presented in Appendix A.

Several generative models proposed for tabular data—MedGAN (Choi et al., 2017), TableGAN (Park et al., 2018), CTGAN (Xu et al., 2019) and TVAE (Xu et al., 2019)–were used as baseline. We use Identity to denote the model for which the generated and the training data are the same. Therefore, Identity yields the best performance uniformly. This means that the model generating synthetic data with distribution most similar to the Identity model's distribution performs best. For all the baseline models, we use the recommended hyperparameters presented in the original papers or provided in their implementations. Each model is trained with a batch size of 500 for 300 epochs. For clarity of presentation, we highlight the best performance in bold font and underline the second-best performance.

5.2 EVALUATION METRICS

There is no uniform metric that can be measured to evaluate the quality of the synthesized tabular data accurately because of the combination of discrete, continuous, or mixed-type variables. Therefore, we use five different metrics to evaluate the performances on different types. The details are as follows.

The **Kullback–Leibler (KL) divergence** quantifies how much one probability distribution differs from another probability distribution. The KL divergence between synthetic ($P(x)$) and target ($Q(x)$) empirical distributions is:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \epsilon \mathcal{X}} P(x) \log(\frac{P(x)}{Q(x)}) \tag{8}$$

where $\mathcal{X}$ denotes the set of all possible samples. We use KL divergence for evaluating the mixed-type tabular datasets. For computations, continuous variables are binned to be treated as discrete variables.

Pairwise correlation difference (PCD) metric compares the difference of inter-variable correlation between the synthetic and real-world datasets. PCD measures how well a synthetic dataset captures the statistical properties of the corresponding real dataset. The PCD is defined as:

$$PCD(X_S, X_R) = \frac{1}{N} ||Corr(X_S) - Corr(X_R)||_1 \tag{9}$$

where N is the number of variables and $Corr(\cdot)$ denotes the inter-variables' correlation for a dataset. In our experiment, we use Pearson correlation (Freedman et al., 2007) for $Corr(\cdot)$. We use PCD on continuous-only datasets instead of KL divergence since the binning process in calculating KL divergence can reduce the precision.

Bayesian network (BN)-likelihood (Xu et al., 2019) is an evaluation metric specifically designed for Bayesian networks. BN-likelihood measures the likelihood that a synthetic dataset is derived from the Bayesian network (the oracle) generating the target dataset. This likelihood is denoted by L_{syn}. Note that L_{syn} can be high when over-fitting occurs. To solve this issue, L_{test} score that is robust to over-fitting is used. To calculate L_{test}, we first use same oracle graph structure to fit the synthetic dataset and then measure the likelihood of the target dataset being generated by the fitted BN.

Machine learning efficacy (Xu et al., 2019; Goncalves et al., 2020) evaluates how much the synthetic dataset represents the target dataset when used as the training data in a supervised machine learning task. The amount of representation can be quantified by the performance measurement of the machine learning task (e.g., accuracy in classification task). To measure the machine learning efficacy, we train machine learning models on the synthetic datasets and then evaluate the trained models on the test data consisting of target datasets. We use the Macro-F1 score for classification tasks and R^2 for regression task to measure the machine learning efficacy score. The used machine learning models for classification are the MLP classifier, AdaBoost, and Decision Tree. For regression, we use Linear

Method	Discrete		Continuous		Mixed	
	L_{syn}	L_{test}	PCD	Log-cluster	KL Div.	Log-cluster
Identity	-9.48	-9.51	0	$+\infty$	1	$+\infty$
MedGAN	-10.68	-12.00	-0.615	1.42	0.430	1.80
TableGAN	-13.68	-11.34	-0.032	6.23	0.802	2.39
TVAE	-10.41	-9.92	-0.068	3.88	0.838	2.84
CTGAN	-14.19	-11.30	-0.138	2.77	0.806	2.94
Causal-TGAN (Ours)	**-9.70**	**-9.62**	**-0.013**	**6.48**	**0.971**	**3.06**

Table 1: **Causal-TGAN in full expert knowledge setting**. Results on different types of simulated datasets. For all metrics, higher values indicate better performances. Scores for each dataset are provided in Appendix C.

regressor and MLP regressor. All the implementations of the machine learning algorithms mentioned above were from scikit-learn (Pedregosa et al., 2011). Full details of the model settings used in our experiments are provided in Appendix B.

The **Log-cluster** score (Goncalves et al., 2020) is intended to evaluate synthetic datasets in unsupervised machine learning tasks such as clustering. To measure log-cluster, we first concatenate the synthetic dataset and the real dataset into a single dataset. Secondly, a clustering method is applied to the concatenated dataset with a fixed number of clusters G. Then the log-cluster score can be calculated as:

$$C_{log}(X_S, X_R) = \log(\frac{1}{G}\sum_{i=1}^{G}[\frac{n_i^r}{n_i} - c]^2) \tag{10}$$

where n_i is the number of samples in the i^{th} cluster and n_i^r is the number of samples in n_i that form the target dataset. c is defined the as ratio of the number of samples in the target dataset to the number of samples in the concatenated dataset. A large value of C_{log} indicates a severe mismatch in cluster members, indicating a disparity between the distribution of target and synthetic datasets. We set G equal to 100 in our experiments.

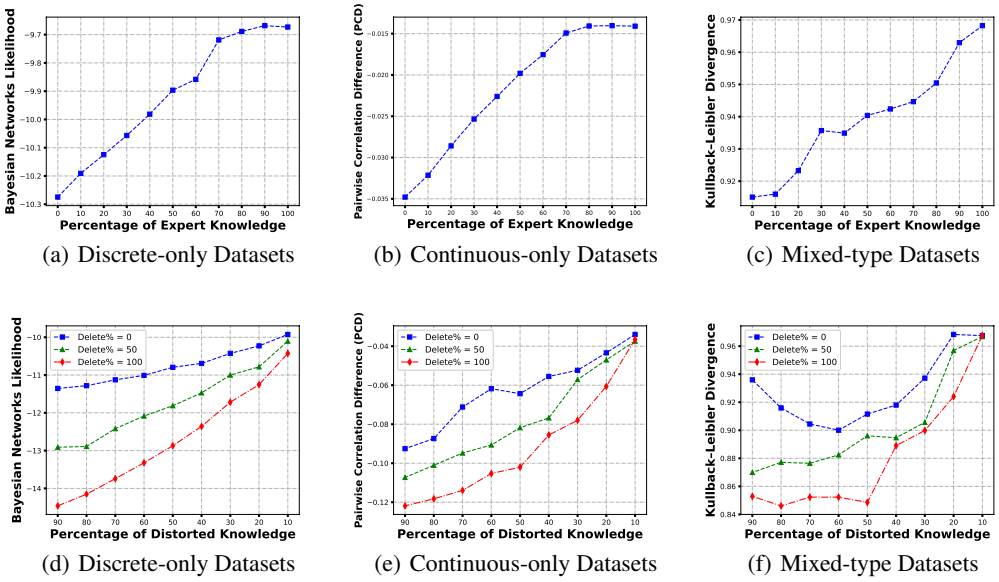
To make the value of these metrics proportional to the synthetic data quality, we transform the values. For PCD and Log-cluster, we report their negative values. The KL divergence is used to compute the metric, $D_{KL} = 1/(1 + D_{KL})$.

5.3 Comprehensive Experimental Analysis of Causal-TGAN

In this section, we evaluate Causal-TGAN on simulated datasets for which the causal relations for all the variables are known. We also thoroughly investigate the performance of Causal-TGAN with distorted causal information. The performance of Causal-TGAN with full expert knowledge is first presented followed by the details of how we manipulate the true causal graph for examining Causal-TGAN's performance, and the corresponding results.

Full Causal Knowledge Performance comparison of baseline models with Causal-TGAN is reported in Table 1. Notice from the table that Causal-TGAN outperforms all the baseline models on all types of datasets. MedGAN fails to produce comparable results on continuous-only and mixed-type datasets since it is designed to deal with multi-label discrete patient data. Overall, TVAE ranks second in performance. However, TVAE must have access to the true datasets for data generation that limits its practical use. Surprisingly, TableGAN outperforms CTGAN on continuous datasets by a large margin. One possible reason for this is that the continuous variables in the simulated datasets do not conform to a mixture Gaussian distribution. CTGAN's mixture Gaussian model could be introducing a mismatch. However, note that, even though Causal-TGAN employs the same encoding strategy as CTGAN, it still outperforms TableGAN by incorporating causal knowledge.

Partial Causal Knowledge To create partial knowledge, we prune the true causal graph by different percentages. The percentage of knowledge is calculated as the percentage of remaining or unpruned



(a) Discrete-only Datasets (b) Continuous-only Datasets (c) Mixed-type Datasets

(d) Discrete-only Datasets (e) Continuous-only Datasets (f) Mixed-type Datasets

Figure 3: Performance of Causal-TGAN for different qualities of expert knowledge. (a)-(c) true graphs are given in different proportion. (d)-(f) different degrees of graph distortion. Plots for each dataset are provided in Appendix D

variables. 100% denotes knowing causal relations for all variables and 0% denotes generating using purely conditional GAN [1]. For each percentage of knowledge, we randomly generate 10 corresponding causal graphs and report the average scores.

The average of results for each type of dataset is shown in Figure3 (a)-(c). Firstly, an ablation study can compare the results of 0% knowledge with other fractional knowledge. It indicates that including inter-variable causal relations can improve the data generation performance. Secondly, Causal-TGAN can consistently and stably gain from incorporating more expert knowledge on all types of datasets. However, the gain starts to get saturated when over 80% knowledge is incorporated on discrete-only and continuous-only datasets.

Wrong Knowledge To create a wrong knowledge graph, we distort the graph. This includes either deleting an edge or changing the direction of an edge. The percentage of distortion is measured by the proportion of manipulated edges in the graph. We consider three settings for distortions: deleting entirely (Delete$\%$=100), half deletion and half reversing (Delete$\%$=50), and fully reversal (Delete$\%$=0). Similarly, we randomly generate 10 distorted graphs for each level and each setting of distortion and report the average scores.

The results of our experiments are shown in Figure3 (d)-(f). Except for Delete$\%$=0 on the mixed-type datasets, all other results show that the performance of Causal-TGAN degrades as the graph distortion becomes worse. A possible cause of this behavior is that the causal graphs of the mixed-type datasets are simple (i.e., they contain fewer nodes and edges), and hence the reversal of the causal directions to a large degree is more likely to create a Markovian equivalent graph with the same statistical properties of the dataset from the undistorted graph. We find that, when compared to deletion distortion, Causal-TGAN is more robust to distortions due to reversal in the causal directions.

[1] we keep using the term conditional GAN here. However, in our implementation, 0% means there is no need for Causal-TGAN for generation, and the conditional GAN is hence degraded to be a regular GAN without taking conditional vectors.

Method	adult	census	intrusion	news
	F1	F1	Macro	R^2
Identity	0.677	0.648	0.756	0.038
MedGAN	0.045	0.000	0.000	-1.182
TableGAN	0.496	0.367	0.308	-0.521
CTGAN	<u>0.628</u>	0.459	<u>0.576</u>	<u>0.006</u>
TVAE	0.611	<u>0.464</u>	0.380	-0.055
Causal-TGAN-NK	0.658	0.401	0.468	0.008
Causal-TGAN-PK	**0.662**	**0.509**	**0.577**	**0.025**

Table 2: Machine learning efficacy on real-world datasets. Note that, Causal-TGAN-NK is excluded from comparison with baseline models since it is included for ablation study purposes.

5.4 Results on Real-world Dataset

We trained Causal-TGAN with partial knowledge setting to evaluate it on real-world datasets. The reason for this is causal discovery methods for mixed-type datasets (Tsagris et al., 2018) have a much lower recall score than those designed for single-type datasets (Ramsey, 2015). A lower recall score of a causal discovery method indicates that it identifies insufficient causal relationships. Section 5.3 showed that Causal-TGAN is more vulnerable to the absence of causal relations. Therefore, we designed an unbiased strategy for training Causal-TGAN on real-world datasets, i.e., we estimated the causal relations for variables of the majority data type for each dataset. This strategy delivers as much accurate partial knowledge as possible. For example, we estimate causal relations for continuous variables in the news dataset, in which 45 out of 59 variables are of the continuous type. We also trained Causal-TGAN in the no-knowledge setting to differentiate the performance difference when expert knowledge is available. We denote Causal-TGAN with partial knowledge as **Causal-TGAN-PK** and Causal-TGAN with no knowledge as **Causal-TGAN-NK**.

The results on real-world datasets are reported in Table 2. It illustrates that, even when full knowledge is absent, Causal-TGAN outperforms all the baseline models. On the census and intrusion datasets, by incorporating expert knowledge, Causal-TGAN surpasses CTGAN in its performance, whereas originally, Causal-TGAN was not as good as CTGAN. The ablation results derived from comparing Causal-TGAN-PK and Causal-TGAN-NK further validate the intuition that incorporation of expert knowledge helps to improve the quality of synthesized data.

6 Conclusion

We propose Causal-TGAN, a tabular data generative model that leverages inter-variable causal relationships. This method can handle discrete, continuous, and mixed data types. When combined with an auxiliary conditional GAN, the proposed approach can flexibly consume different types or qualities (complete or partial) of expert knowledge about the underlying causal structures. Extensive experimental evaluation on simulated and real-life datasets indicates superior performance and practicality of Causal-TGAN when compared to several other baseline generative models available in the literature.

References

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. *arXiv preprint arXiv:1807.08024*, 2018.

Aggelina Chatziagapi, Georgios Paraskevopoulos, Dimitris Sgouropoulos, Georgios Pantazopoulos, Malvina Nikandrou, Theodoros Giannakopoulos, Athanasios Katsamanis, Alexandros Potamianos, and Shrikanth Narayanan. Data augmentation using gans for speech emotion recognition. In *Interspeech*, pp. 171–175, 2019.

Haipeng Chen, Sushil Jajodia, Jing Liu, Noseong Park, Vadim Sokolov, and VS Subrahmanian. Faketables: Using gans to generate functional dependency preserving tables with bounded real data. In *IJCAI*, pp. 2074–2080, 2019.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pp. 286–305. PMLR, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

Angelo Genovese, Vincenzo Piuri, and Fabio Scotti. Towards explainable face aging with generative adversarial networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3806–3810. IEEE, 2019.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC medical research methodology*, 20(1): 1–40, 2020.

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.

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. In *Explainable and interpretable models in computer vision and machine learning*, pp. 39–80. Springer, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

Eoin M Kenny and Mark T Keane. On generating plausible counterfactual and semi-factual explanations for deep learning. *arXiv preprint arXiv:2009.06399*, 2020.

Jayoung Kim, Jinsung Jeon, Jaehoon Lee, Jihyeon Hyeong, and Noseong Park. Oct-gan: Neural ode-based conditional tabular gans. In *Proceedings of the Web Conference 2021*, pp. 1506–1515, 2021.

Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. Generate your counterfactuals: Towards controlled counterfactual generation for text. *arXiv preprint arXiv:2012.04698*, 2020.

Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 613–628, 2018.

Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pp. 2642–2651. PMLR, 2017.

Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. In *The 44th International Conference on Very Large Data Bases*, volume 11, 2018.

Judea Pearl. *Causality*. Cambridge university press, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Joseph D Ramsey. Scaling up greedy causal search for continuous variables. *arXiv preprint arXiv:1507.07749*, 2015.

Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

Yong-Goo Shin, Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Wook Kim, and Sung-Jea Ko. Pepsi++: Fast and lightweight network for image inpainting. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):252–265, 2020.

Ramya Srinivasan, Ajay Chander, and Pouya Pezeshkpour. Generating user-friendly explanations for loan denials using gans. *arXiv preprint arXiv:1906.10244*, 2019.

Amirsina Torfi, , and Edward A. Fox. Corgan: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. In *The 33rd International FLAIRS Conference, AI in Healthcare Informatics*, pp. 335–340, 2020.

Michail Tsagris, Giorgos Borboudakis, Vincenzo Lagani, and Ioannis Tsamardinos. Constraint-based causal discovery with mixed data. *International journal of data science and analytics*, 6(1):19–30, 2018.

Haodi Wang, Libin Jiao, Hao Wu, and Rongfang Bie. New inpainting algorithm based on simplified context encoders and multi-scale adversarial network. *Procedia computer science*, 147:254–263, 2019.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *33rd Conference on Neural Information Processing Systems*, 2019.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, pp. 5689–5698. PMLR, 2018.

Zilong Zhao, Aditya Kunar, Hiek Van der Scheer, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. *arXiv preprint arXiv:2102.08369*, 2021.

A DATASETS DESCRIPTION

The statistics of datasets are summarized in Table 3 and the websites for all the datasets are listed as follows:

- Simulated Datasets: http://www.bnlearn.com/bnrepository/
- Adult: http://archive.ics.uci.edu/ml/datasets/adult
- Census: https://archive.ics.uci.edu/ml/datasets/census+income
- Intrusion: http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data
- News: https://archive.ics.uci.edu/ml/datasets/online+news+popularity

Dataset Name	# train/test	# Cols.	# Binary Cols.	# Categorical Cols.	# Continuous Cols.	Task
Simulated Dataset						
asia	10k/10k	8	0	8	0	-
child	10k/10k	20	0	20	0	-
insurance	10k/10k	27	7	20	0	-
alarm	10k/10k	37	10	27	0	-
ecoli70	15k/15k	46	0	0	46	-
arth150	25k/25k	107	0	0	107	-
healthcare	15k/15k	7	0	3	4	-
mehra	25k/25k	24	0	4	20	-
Real Dataset						
adult	23k/10k	15	2	7	6	**C**
census	200k/100k	41	3	31	7	**C**
intrusion	394k/100k	41	5	10	26	**C**
news	31k/8k	59	14	0	45	**R**

Table 3: Summary of datasets. Note that, # denotes the number of and Col. is the abbreviation of columns. The machine learning tasks for real-world datasets are listed in the last column, where C stands for classification and R stands for regression. We separate the train and test datasets by randomly sampling from the whole dataset.

B MODEL SETTINGS IN MACHINE LEARNING EFFICACY

Dataset	Model	Description
adult	Adaboost	**n_estimators=50**, and others=defaulted values.
	Decision Tree	**max_depth=20**, **max_leaf_nodes=50**, and others=defaulted values.
	MLP	**hidden_layer_sizes=50**, **early_stopping=True**, and others=defaulted values.
census	Adaboost	**n_estimators=50**, and others=defaulted values.
	Decision Tree	**max_depth=20** and others=defaulted values.
	MLP	**hidden_layer_sizes=50, early_stopping=True**, and others=defaulted values.
intrusion	Adaboost	**n_estimators=50**, and others=defaulted values.
	Decision Tree	**max_depth=20** and others=defaulted values.
	MLP	**hidden_layer_sizes=50, early_stopping=True**, and others=defaulted values.
news	Linear Regression	All settings with defaulted values.
	MLP	**hidden_layer_sizes=100**, **early_stopping=True**, and others=defaulted values.

Table 4: Classifier and Regressor used in the evaluation of Machine Learning Efficacy. The names of all parameters that used in the description are consistent with those defined in scikit-learn.

C BENCHMARK RESULTS ON SIMULATED DATASETS

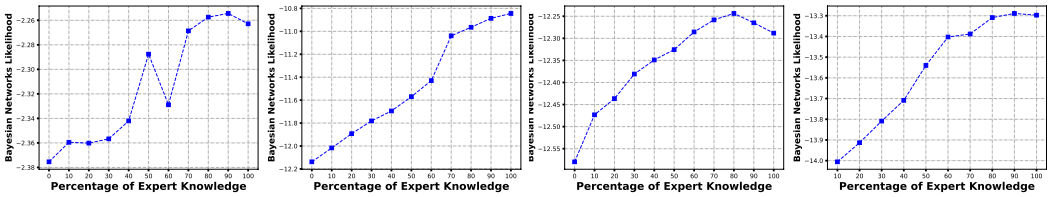
Method	asia		alarm		child		insurance	
	L_{syn}	L_{test}	L_{syn}	L_{test}	L_{syn}	L_{test}	L_{syn}	L_{test}
Identity	-2.23	-2.25	-10.4	-10.5	-12.2	-12.2	-13.1	-13.1
MedGAN	-2.81	-2.59	-11.1	-14.2	-14.2	-15.4	-14.6	-15.8
TableGAN	-3.52	-2.75	-16.3	-12.5	-17.6	-14.1	-17.3	-16.0
TVAE	-2.35	-2.28	-12.2	-11.0	**-12.3**	-12.5	-14.8	-13.9
CTGAN	-4.34	-2.49	-20.3	-14.1	-14.9	-13.2	-17.2	-15.4
Causal-TGAN (Ours)	**-2.32**	**-2.26**	**-10.7**	**-10.8**	**-12.3**	**-12.2**	**-13.5**	**-13.2**

Table 5: Benchmark results on discrete-only simulated datasets.

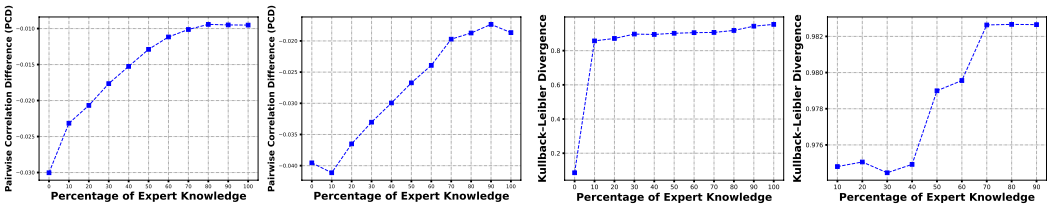
Method	ecoli70		arth150		healthcare		mehra	
	PCD	Log-cluster	PCD	Log-cluster	KL div.	Log-cluster	KL div.	Log-cluster
Identity	0	$+\infty$	0	$+\infty$	1	$+\infty$	1	$+\infty$
MedGAN	-0.477	1.41	-0.753	1.43	0.432	1.57	0.428	2.02
TableGAN	**-0.010**	6.68	-0.053	5.78	0.737	2.85	0.866	1.92
TVAE	-0.044	5.63	-0.092	2.14	0.901	3.62	0.775	2.06
CTGAN	-0.227	2.71	-0.049	2.82	0.733	3.84	0.878	2.04
Causal-TGAN (Ours)	-0.018	**6.69**	**-0.008**	**6.27**	**0.956**	**4.04**	**0.985**	**2.08**

Table 6: Benchmark results on continuous-only and mixed-type simulated datasets. Specifically, ecoli70 and arth150 are continuous-only datasets and healthcare and mehra are mixed-type datasets.

D EXPERT KNOWLEDGE OF DIFFERENT QUALITIES

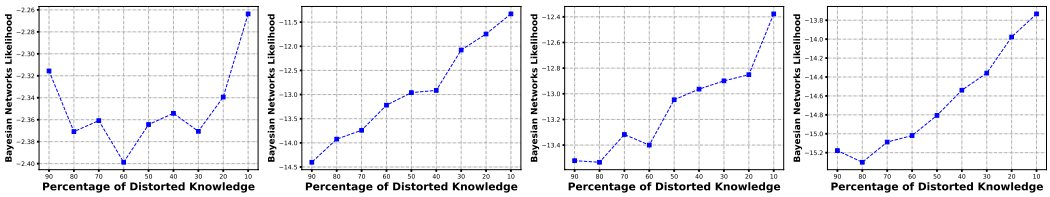


(a) asia (b) alarm (c) child (d) insurance

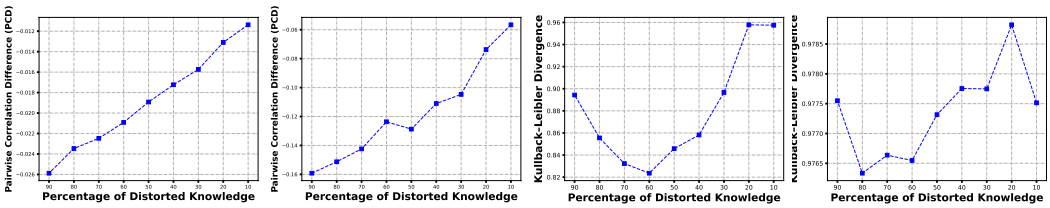


(e) arth150 (f) ecoli70 (g) healthcare (h) mehra-complete

Figure 4: **Partial Knowledge.** (a)-(d) are discrete-only datasets; (e)-(f) are continuous-only datasets; (g)-(h) are mixed datasets.

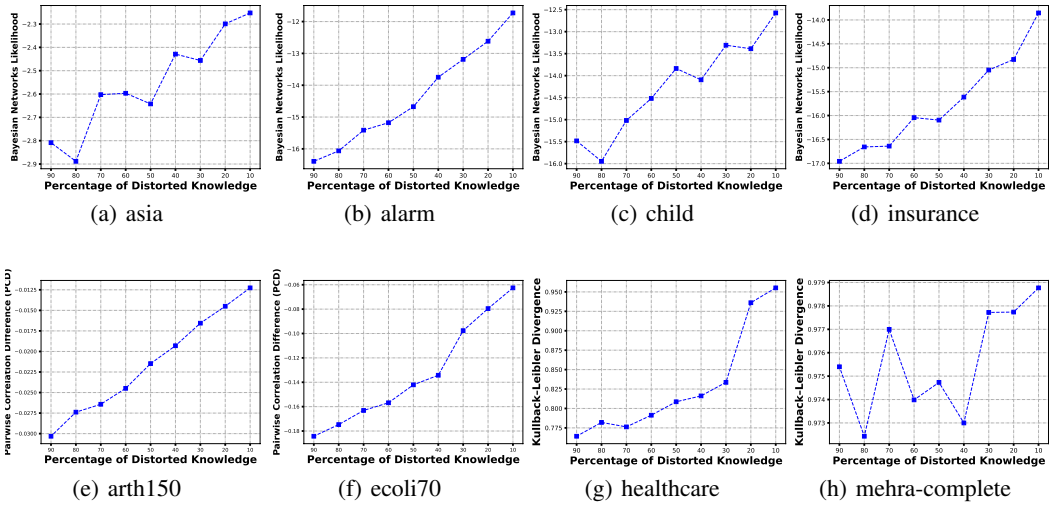


(a) asia (b) alarm (c) child (d) insurance



(e) arth150 (f) ecoli70 (g) healthcare (h) mehra-complete

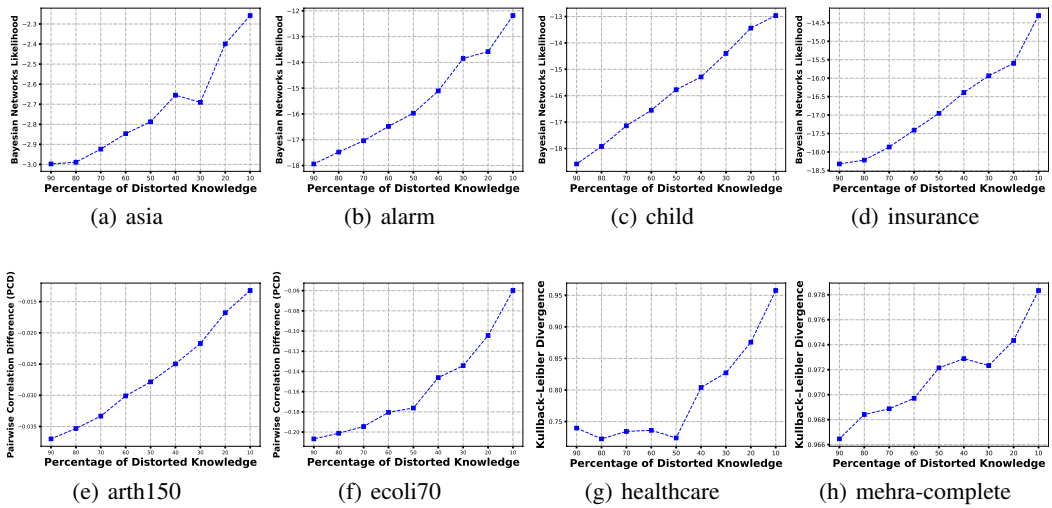
Figure 5: **Wrong Knowledge: fully reversing (Delete=0%).** (a)-(d) are discrete-only datasets; (e)-(f) are continuous-only datasets; (g)-(h) are mixed datasets.



(a) asia (b) alarm (c) child (d) insurance

(e) arth150 (f) ecoli70 (g) healthcare (h) mehra-complete

Figure 6: **Wrong Knowledge: half deleting and half reversing (Delete=50%).** (a)-(d) are discrete-only datasets; (e)-(f) are continuous-only datasets; (g)-(h) are mixed datasets.



(a) asia (b) alarm (c) child (d) insurance

(e) arth150 (f) ecoli70 (g) healthcare (h) mehra-complete

Figure 7: **Wrong Knowledge: fully deleting (Delete=100%).** (a)-(d) are discrete-only datasets; (e)-(f) are continuous-only datasets; (g)-(h) are mixed datasets.