Journal of Computational and Applied Mathematics 457 (2025) 116312

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

On the causality-preservation capabilities of generative modelling

Yves-Cédric Bauwelinckx [a], Jan Dhaene [a,*], Milan van den Heuvel [d], Tim Verdonck [c,b]

[a] Faculty of Economics and Business, KU Leuven, Naamsestraat 69, Leuven, 3000, Belgium
[b] Department of Mathematics, KU Leuven, Celestijnenlaan 200B, Leuven, 3001, Belgium
[c] Department of Mathematics, University of Antwerp, Middelheimlaan 1, Antwerp, 2020, Belgium
[d] Department of Economics, Ghent University, Sint-Pietersplein 6, Gent, 9000, Belgium

ARTICLE INFO

Keywords:
Synthetic data
Generative adversarial networks
Generative modelling
Causality
Data science
Shortcut

ABSTRACT

Modelling is essential in both the financial and insurance industries. The emergence of machine learning and deep learning models offers new tools for this, but they often require large datasets that are typically unavailable in business fields due to privacy and ethical concerns. This lack of data is currently one of the main hurdles in developing better models. Generative modelling, such as Generative Adversarial Networks (GANs), can address this issue by creating synthetic data that can be freely shared. While GANs are widely studied in fields like computer vision, their use in business is limited, primarily because business questions often focus on identifying causal effects, whereas GANs and neural networks typically emphasise high-dimensional correlations. This paper explores whether GANs can produce synthetic data that reliably answers causal questions by performing causal analyses on GAN-generated data under varying assumptions. The study includes cross-sectional, time series, and complete structural model scenarios. Findings show that while basic GANs replicate causal relationships in simple cross-sectional data, they struggle with more complex structural models. In contrast, CausalGAN effectively replicates the original causal model, and TimeGAN modifies the causal representation in time series data.

Contents

* Corresponding author.
E-mail addresses: Yves-Cedric.Bauwelinckx@kuleuven.be (Y.-C. Bauwelinckx), Jan.Dhaene@kuleuven.be (J. Dhaene), Milan.vandenHeuvel@UGent.be (M. van den Heuvel), Tim.Verdonck@uantwerpen.be (T. Verdonck).

https://doi.org/10.1016/j.cam.2024.116312
Received 9 January 2024; Received in revised form 28 September 2024
Available online 9 October 2024

1. Introduction

To make sense of the complexities of reality, and make optimal decisions accordingly, organisations and researchers have always striven to come up with models that can accurately represent observed phenomena (e.g. consumption behaviour, loan defaults). In the past, these models were defined by the analyst and calibrated to (small) data. Recently, however, during the so-called machine learning revolution, the focus shifted to a more data-driven, algorithmic approach. Machine learning algorithms now search for the optimal model by finding support for it in the data instead of being chosen by the analyst. This approach led to the increased collection of, investment in, demand for, reliance on, and value of data for organisations and research significantly [1]. It has also brought the tension between utility of data and privacy of its subjects to the forefront of public discussion [2]. Recently developed generative modelling methods, which create data with a distribution similar to the original while excluding any real data, have been proposed as a potential solution [3]. Decision-making is, however, almost always a causal question and little is known about the replication capabilities of these methods beyond correlations. For this reason, this paper seeks to fill the gap by performing an investigation of the causal replication capabilities of data replication methods as well as defining a path forward to making them a viable option for decision-making.

There are a lot of advantages to the algorithmic approach to modelling, the most important being increased performance and the opportunity for analysts to be systematic and transparent about the process by which the model was selected [4]. The power of this approach has been apparent in several fields that have had incredible advances in replicating reality due to the availability of large amounts of data. One of the most famous examples is ImageNet, a database with millions of hand-labelled pictures, enabling revolutionary progress in image recognition [5]. More recently GPT-3, a multi-purpose natural language model, similarly achieved impressive results after learning from a data set containing 45 TB of plain text [6]. However, such large amounts of data are not always readily available. In many fields centring around individuals, such as the social and health sciences (e.g. finance, insurance, medical fields), the collecting or sharing of such datasets is far from trivial due to ethical and privacy concerns [7]. One recently emerging option to alleviating such concerns is generative modelling.

Generative models aim to learn the (high-dimensional) distribution of a dataset, but traditional neural networks, which underpin models like GANs, often focus on correlations rather than causality. This limitation hampers their ability to generalise across different contexts. Finding the causal structure from observational data is a big challenge however. This problem, finding the cause from the effect is called an inverse problem. Consider this example in The Black Swan from Taleb [8]:

"*Operation 1 (the melting ice cube): Imagine an ice cube, and consider how it may melt over the next two hours while you play a few rounds of poker with your friends. Try to envision the shape of the resulting puddle.*

Operation 2 (where did the water come from?): Consider a puddle of water on the floor. Now try to reconstruct in your mind's eye the shape of the ice cube it may once have been. Note that the puddle may not have necessarily originated from an ice cube".

Operation 1 is an example of the forward way of thinking, where the effect (the water) is to be predicted from the cause (ice cube). With the right models it is possible to accurately come up with the resulting pool of water. In contrast, operation 2 asks the inverse, finding the shape of the cube (cause) from the pool of water (effect). There are however an almost infinite amount of possible ice cubes that could have led to that pool of water. This example also translates to joint probability distributions and underlying causal models. For a given joint distribution there are a multitude of possible underlying causal models. This non-uniqueness leads to uncertainty in determining the causal model from a joint probability distribution [9].

In this paper, we survey the literature on generative adversarial networks, and evaluate their capacity to preserve certain causal structures (i.e. cross-sectional, time series, and full structural) in the synthetic datasets they generate. We do so by first generating a dataset where the data-generating function, and thus the structural causal model, is known. Secondly, we make a synthetic copy of this known dataset with a specific GAN method and perform different causal analyses with an increasingly lenient set of assumptions, from cross-sectional to time-series to structural. The considered GAN models for these experiments are the original GAN model [10], TimeGAN [11] and CausalGAN [12], each with their own focus of preserving the structure of the data. Lastly, we check if the results in the real data align with those in the synthetic data to evaluate the causality preserving capabilities.

We find that for relationships in data where the assumptions hold such that correlation equals causation, inference on the real and synthetic data yield the same results only in the case where the actual causal structure aligns with the most simple model that can

replicate the correlations in the data. In more complex cases, for instance when a variable has time-dependence and both influences cross-sectional features as well as itself, we find that the generative model converges on a model with the same general distribution, but that it does so with a simpler underlying causal structure. Our results point at the reason being the often-used regularisation in machine learning that builds in a preference for smaller models (as posited in Occam's razor) which is not necessarily a valid principle in causality. Finally, when the whole causal structure is considered, it becomes apparent that currently the applicability is still limited due to the stringent assumptions that need to be met in order to overcome the challenges of the inverse problem.

The remainder of this paper is structured as follows. In Section 2, we overview the field of generative machine learning models and the relation to causality. In Section 3, we lay-out the problem setup and discuss the structural approach we take to evaluate the causal replication capacity of GAN-based models. In Section 4, we give a general introduction to the inner workings of GAN-models and detail three different GAN variations that we take as representative for the different streams in the GAN literature that aim to capture increasingly complex correlations (i.e. cross-sectional correlations, time-series correlations, full causal structure). In Section 5, we present the results of our evaluation. In Section 6, we discuss some of the additional real-world challenges that we abstracted away from but that need to be considered where these methods to be used in real-world cases. Lastly, in Section 7, we summarise and conclude our findings.

2. Literature review

Generative models aim to learn a representation of the high-dimensional distribution of a dataset. Once this representation is learned, it can then be used to generate new samples that maintain the original dataset's distribution of features but that did not appear in the original dataset.[1] Generative methods are thus capable of simulating non-existent but realistic-looking data, also referred to as *synthetic data*, that can be shared more freely. A well-known use-case is pictures of human faces for computer vision applications. Even in the possession of a large dataset of pictures of human faces, sharing this freely could present issues concerning privacy. However, generative models are capable of constructing fake but human-looking faces that can, due to their non-existence, be shared more freely to further the quality of applications.

While generative modelling has been around for decades, a major breakthrough in the ability to efficiently training such models was achieved in 2014 with Generative Adversarial Networks (GANs) [10]. This method increased our capacity to fit high-dimensional distributions of data, like images and video data. The GAN framework has found widespread applications throughout computer vision, like image generation [15,16], text to image translation [17], the blending of images [18], enhancing quality of pictures [19,20], filling in blanks in pictures [21], removing rain droplets from pictures [22] and a more infamous example of deepfakes [23]. While these are noteworthy variations and applications of the GAN framework, the common factor here is the focus on computer vision.

A limited amount of applications of the GAN framework have been found in fields where data is numerical or tabular in form. Some examples are real-time risk warning of process industries [24], traffic event detection [25], risk forecasting in financial markets [26] as well as synthetic data generation of credit card [27], housing [28] and insurance [29] data. Similarly, there are also GAN applications adapted for time series data. Some examples are financial time series generation [30], electricity price and consumption data [31,32], or just time series generation in general [11,33,34]. A more complete overview of GANs for tabular data and time series can be found in the work of Fonseca et al. [35] and Brophy et al. [36] respectively.

The adoption of GANs in these fields, especially in human sciences like economics, is still quite limited. The main reason for this is that in these fields, most questions are inherently about identification of causal effects. Neural networks, which are at the centre of the GAN framework, in contrast, still focus mostly on high-dimensional correlations. An example of this is shown in the paper by [37], where they analyse a neural network trained to classify images. The neural network appears to be able to accurately identify whether or not there is a cow in a picture, until you ask the network to classify a picture of a cow in an uncommon environment. The model is, for instance, not able to recognise a cow on a beach, because of the spurious correlation between cows and grasslands. Learning to label images with grass in it are shortcuts that expose the lack of generalisation of the neural network, unable to adapt to a new domain [38]. Recently, a field has emerged called *Causal Machine Learning* where researchers try to make steps towards making machine learning models more causal [39]. While this field is promising, due to the inverse problem nature of finding causality in observational data, it is currently still in its infancy in regards to applicability.

The most prevalent used loss-functions for GANs are some form of binary cross-entropy [10,11,16,40] or Wasserstein distance [41–43]. These losses indicate in some form or another the difference between two joint probability functions. Replicating the joint probability distribution, however, does not guarantee replication of the underlying causal process.

Other work has been done combining the idea of causality and generative modelling. DAG-GAN [44] uses the generative adversarial network framework for causal discovery. CGN [45] is able to generate counterfactual images and can use these to improve out-of-distribution robustness. GANs incorporating causal models have also been explored. For CausalGAN [12] the focus is on intervening on images through the causal model and generating more creative images. Causal-TGAN [46] structures the generator according to the causal model of the data. To judge the quality of the generated data they look at the how well the synthetic data performs in prediction tasks. This differs from the causal preservation we want to evaluate.

Recently, diffusion models showed impressive results in image generation as well [47]. With adaptations of this method still mainly focusing on image generation, the possible models that focus on generating data at the cross-sectional and time series level

[1] Note that to have such privacy guarantees, one needs to explicitly include an optimisation for it in the model fitting step such as in [13]. Else there are cases when replication could occur [14].

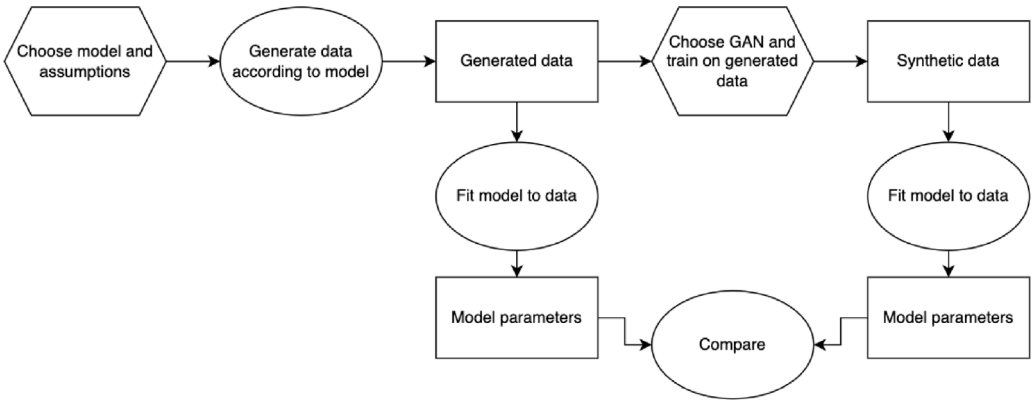


Fig. 1. Experiment setup for each choice of assumptions and GAN method.

are still limited [48,49], while at the full structural level no adaptation has been proposed to our knowledge. As the field of diffusion models develops more our research could also be extended to this model. For now however, the focus is on the more established generative adversarial networks.

3. Problem setup

In this section we explain how we will evaluate how well causal relations are replicated by the generative models. The evaluation setup is shown in Fig. 1. First, a data generation model with a known causal structure is made according to the assumptions listed in Sections 3.1–3.3. The resulting model is discussed in Section 3.4. This will be used to generate a dataset (*generated dataset*) that will be used to train the generative machine learning models discussed in Section 4. Experiments will then be done to see if the causal structure imposed in the *generated data* is still present in the *synthetic data* made by the trained generative models. The results of these experiments are shown in Section 5.

3.1. Cross-sectional

The first type of causal relationships are those on a cross-sectional level. Ordinary least squares (OLS) is a popular regression model to find causal effects in cross-sectional data. In this case we assume that a variable can be represented by an OLS model. The OLS model produces valid causal inference under the following assumptions:

Assumption 1 (*Linear in Parameters*)**.** The model can be written in the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon. \tag{1}$$

Assumption 2 (*Random Sampling*)**.** The sample of n observations $(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i) : i = 1, 2, \ldots, n$ is drawn randomly from the model.

Assumption 3 (*No Perfect Collinearity*)**.** An independent variable in (1) cannot be an exact linear combination of the other independent variables.

Assumption 4 (*Zero Conditional Mean*)**.** The expected value of the error ϵ should be zero, given any values of the independent variables:

$$E(\epsilon | x_1, x_2, \ldots, x_k) = 0.$$

Assumption 5 (*Homoskedasticity*)**.** The error ϵ has the same variance, given any values of the independent variables. This can be noted as:

$$Var(\epsilon | x_1, x_2, \ldots, x_k) = \sigma^2.$$

Assumption 6 (*Normality*)**.** The error ϵ is normally distributed with a mean of zero and variance σ^2. The error is independent of the explanatory variables $x_1, x_2, \ldots, x_k$. More simply:

$$\epsilon \sim Normal(0, \sigma^2).$$

3.2. Time-series

In cross-sectional modelling observations have no time aspect, this changes when considering time-series models. Here we consider the popular class of linear autoregressive models. The assumptions to perform valid causal inference with these models are as follows:

Assumption 1 (*Linear in Parameters*)**.** The stochastic process $(x_{t1}, x_{t2}, \dots, x_{tk}, y_t) : t = 1, 2, \dots, n$ can be written in the form:

$$y_t = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \cdots + \beta_k x_{tk} + \epsilon_t. \tag{2}$$

Assumption 2 (*No Perfect Collinearity*)**.** An independent variable in (2) cannot be an exact linear combination of the other independent variables.

Assumption 3 (*Zero Conditional Mean*)**.** For each t, the expected value of the error ϵ_t should be zero, given any values of the independent variables:

$$E(\epsilon_t | x_{t1}, x_{t2}, \dots, x_{tk}) = 0, t = 1, 2, \dots, n.$$

Assumption 4 (*Homoskedasticity*)**.** The error ϵ_t has the same variance, given any values of the independent variables:

$$Var(\epsilon_t | x_{t1}, x_{t2}, \dots, x_{tk}) = \sigma^2, t = 1, 2, \dots, n.$$

Assumption 5 (*No Serial Correlation*)**.** Given the independent variables $x_{t1}, x_{t2}, \dots, x_{tk}$, errors in two different time steps are not correlated:

$$Corr(\epsilon_s, \epsilon_t | x_{t1}, x_{t2}, \dots, x_{tk}) = 0, \forall t \neq s.$$

Assumption 6 (*Normality*)**.** The error ϵ_t is normally distributed a zero mean and variance σ^2 and independent of the explanatory variables $x_{t1}, x_{t2}, \dots, x_{tk}$.

$$\epsilon_t \sim Normal(0, \sigma^2).$$

Most assumptions are very similar to the previous OLS assumptions. There are two main differences. First is the absence of OLS Assumption 2 specifying observations to be randomly sampled. Under time-series assumptions observations have an order determined by the time step t. Second, time-series Assumption 5 is added, requiring the error term to have no serial correlation.

In the time-series we will consider, autoregressive terms are included as well. We make an additional assumption for this autoregressive time-series to be weakly dependent, meaning the correlation between y_t and y_{t+s} is almost 0 for s large enough. In other words, as the variables get farther away from each other in time, the correlation decreases. In the following case:

$$y_t = \alpha y_{t-1} + \epsilon,$$

the autoregressive model is lagged for one period and the assumption is satisfied if $|\alpha| < 1$.

3.3. Structural model

Lastly, the case remains where the whole causal structure is considered. Here, the goal is to attempt to reconstruct the full structural causal model from the data. As far as we know, no such methods exist in econometrics.[2] For this reason, we adopt a method from the emerging field of causal discovery, primarily within the computer science literature, to accomplish this task.

Recovering the causal model from observational data is far from trivial. Recall the example above of trying to figure out the shape of the ice cube from a pool of water. As many forms of ice cubes can result in the same pool of water, many structural causal models can result in the same observational data. Therefore, picking one of all possible models is dependent on further assumptions made by each causal discovery algorithm. The general approach is to embed known features of causality, such as environment independence [50] or acyclicity [51], into the loss function that a machine learning algorithm optimises for. Even then, it is sometimes only possible to provide a set of possible structural causal models that are all equally able to generate the observational data, also called Markov equivalent. A recent trend is to extend the data to also include interventions and their outcomes [52]. This extra information can be used to exclude certain Markov equivalent models and decrease the set of potential underlying causal models.

One of the more frequently used causal discovery algorithms is LiNGAM [53], which assumes that the causal effects are linear, the generating causal graph is acyclic, that the distribution of the noise is non-gaussian and no unobserved confounders. The LiNGAM model can be expressed in matrix form as follows:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{e},$$

[2] In economics, and many other fields that model complex phenomena, a structural model is defined from theory and then calibrated to data instead of trying to infer the complete model itself from the data.

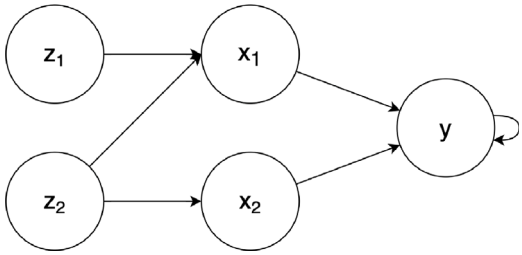


Fig. 2. Full causal model of the generated dataset.

with the observed variables **x**, the connection strength matrix **B** and exogenous variables **e**. The condition of acyclicity allows the matrix **B** to be permuted to become lower triangular with a zero-diagonal. With the additional assumption of the exogenous variables **e**, or in other words the noise, being non-Gaussian, the matrix **B** can be uniquely identified using only the data **x**. This identifiability thus means that the algorithms results in a single causal graph. Different variations on this method exist like models with hidden common causes [54], time-series [55] or non-linearity [56].

In the case of Gaussian noise, only a set of Markov equivalent causal models can be estimated, while under the assumption of non-Gaussian noise this set can be reduced to one full causal model. This assumption is, however, in contrast with the assumption of Gaussian noise that is needed in many inference methods for valid causal inference, including the OLS and autoregressive models we discussed above.

3.4. Generated dataset

To take into account the aforementioned assumptions, we define the following model:

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \epsilon_1, \\ x_{1,t} &= \beta_3 z_{1,t} + \beta_4 z_{2,t} + \epsilon_2, \\ x_{2,t} &= \beta_5 z_{2,t} + \epsilon_3, \\ z_{1,t} &= \epsilon_4, \\ z_{2,t} &= \epsilon_5. \end{aligned} \tag{3}$$

A graphic representation of this structural model, also called the causal graph, is shown in Fig. 2. For the estimation of this causal structure with the different inference methods, we will always assume full observability.

The variables x_1 and x_2 are a linear combinations of the contemporaneous values of z_1 and z_2. The underlying models for these two variables therefore meet the assumptions of the cross-sectional ordinary least squared (OLS) model. OLS should therefore be an appropriate method to estimate the causal effects of z_1 and z_2 on x_1 and x_2. We confirm this in the Results section.

For the variable y_t, extending the assumption on the data to allow for autocorrelation, a first order autoregressive model can infer α on β_1 and β_2.

Finally a variant of LiNGAM for time-series can be used to infer the causal structural model.

While the model was specifically chosen to contain both cross-sectional and time-series causality, it is easy to think of a real-world model that follows this functional form. One example is a simple income process, where the monthly income now depends on the income last month and some contemporaneous features (e.g. employment sector, location) which in turn are distributed according to (conditionally) random distributed preferences.

4. Generative adversarial networks

Generative adversarial networks, or GANs, is a framework for generative machine learning first introduced by Goodfellow et al. in 2014 [10]. A generative model takes a training dataset drawn from a real world distribution as input and tries to replicate this data distribution. The framework has shown great success in generating synthetic images indistinguishable from real images [19,57,58]. While the focus has been on the improvement of the framework for image generation and manipulation, the GAN framework has recently also gathered attention for its possibilities with numerical and categorical data, like tabular and time series data.

For each of the levels we want to consider (cross-sectional, time series and structural) we have selected a version of a GAN that is meant to represent the data at this level well. At the cross-sectional level, the base version of GAN [10] has been chosen as a baseline. For time series, we chose TimeGAN [11] due to its popularity as a general time series generation model and its strong performance. Finally for the structural model we chose CausalGAN [12] as it is the only GAN model that incorporates the causal generative structure of the data. How these models work will be further explained below.

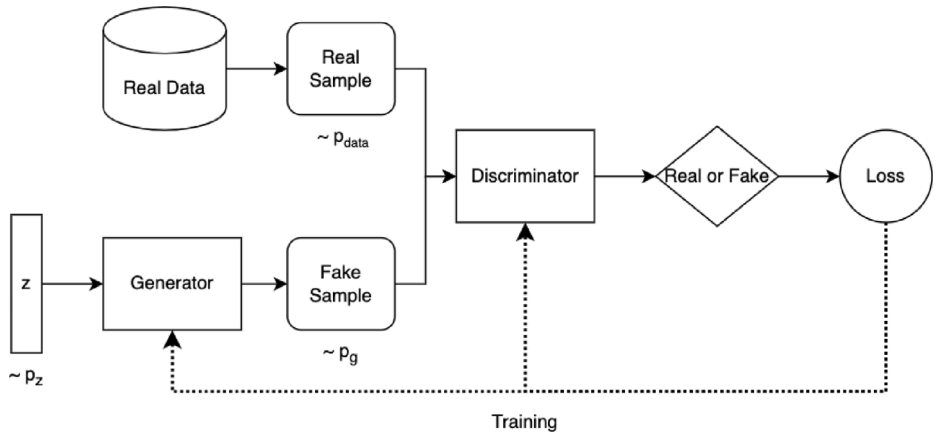


Fig. 3. Generative adversarial networks diagram.

4.1. Framework

A generative adversarial network consists of two competing neural networks: a generator G, which generates fake data, and a discriminator D, that is trained to discern which data is fake (made by the generator) and which data is real. The process can be described as a zero-sum game between the generator and discriminator. During the training process the generator adapts to better fool the discriminator and the discriminator in turn adapts to better detect the fake data. The resulting trained generator can then be extracted to replicate the distribution of the original data.

4.1.1. Architecture

Fig. 3 shows the basic structure of a GAN. The generator G learns to map a latent space p_z to a more complex distribution p_g, which is the distribution meant to mimic the real data distribution p_{data}. Typically, this latent space is a high-dimensional space with each variable drawn from a Gaussian distribution with a mean of zero and a standard deviation of one. The concept is thus that one can insert sample of noise (z) into the generator, which it will learn to map onto a sample of the distribution of the real data. The generating function can then be described by

$$G(z) = X_g, \tag{4}$$

where X_g are samples created by the generator. The discriminator D has the task of distinguishing the fake data X_g from the real data X_{data}. The generator and discriminator are trained by playing a non-cooperative game against each other. The main aim of the generator is to produce samples which are similar to the real data. On the other hand, the main aim of the discriminator is to distinguish between fake samples from the generator and samples from the real data. The discriminator D receives both samples and tries to determine which comes from the real data distribution by assigning a probability D(x), which signifies the certainty the discriminator has in its decision. If D(x) = 1, the sample x is thought to come from p_{data}. On the other hand, if D(x) = 0, the discriminator judges the sample to be from p_g. This prediction from the discriminator and the known ground truth is then used to improve both the generator and the discriminator. During the joint training of the generator and discriminator, G will start to generate increasingly realistic samples to fool the discriminator, while the discriminator learns to better differentiate the real and fake samples. The end goal of the GAN as a whole is that the discriminator can no longer tell the difference between the generated samples X_g (D(x) = 1/2) and the real data samples X_{data} with the discriminator no longer able to improve itself.

Both the generator and discriminator are fully-connected networks to capture the connections between the variables. Both neural network use 3 layers. For the generator these layers contain 128, 64 and 64 nodes respectively. For the discriminator the layers contain 256, 128 and 64 nodes respectively. Both networks use the ReLu activation function for the output of the hidden layers. For the output of the discriminator the sigmoid activation function is used. Formally the networks can be described as:

$$G(z): \begin{cases} \text{Input}: & z \sim \mathcal{N}(0,1)^{d_z}, \\ \text{Hidden layers}: & h_{1,g} = ReLu(W_{1,g}z + b_{1,g}), \\ & h_{2,g} = ReLu(W_2 h_{1,g} + b_{2,g}), \\ & h_{3,g} = ReLu(W_3 h_{2,g} + b_{3,g}), \\ \text{Output}: & G(z) = W_{o,g} h_{3,g} + b_{o,g}, \end{cases}$$

$$D(x): \begin{cases} \text{Input}: & x \sim p_g, p_{data}, \\ \text{Hidden layers}: & h_{1,d} = ReLu(W_{1,d}x + b_{1,d}), \\ & h_{2,d} = ReLu(W_{2,d} h_{1,d} + b_{2,d}), \\ & h_{3,d} = ReLu(W_{3,d} h_{2,d} + b_{3,d}), \\ \text{Output}: & D(x) = \sigma(W_{o,d} h_{3,d} + b_{o,d}), \end{cases}$$

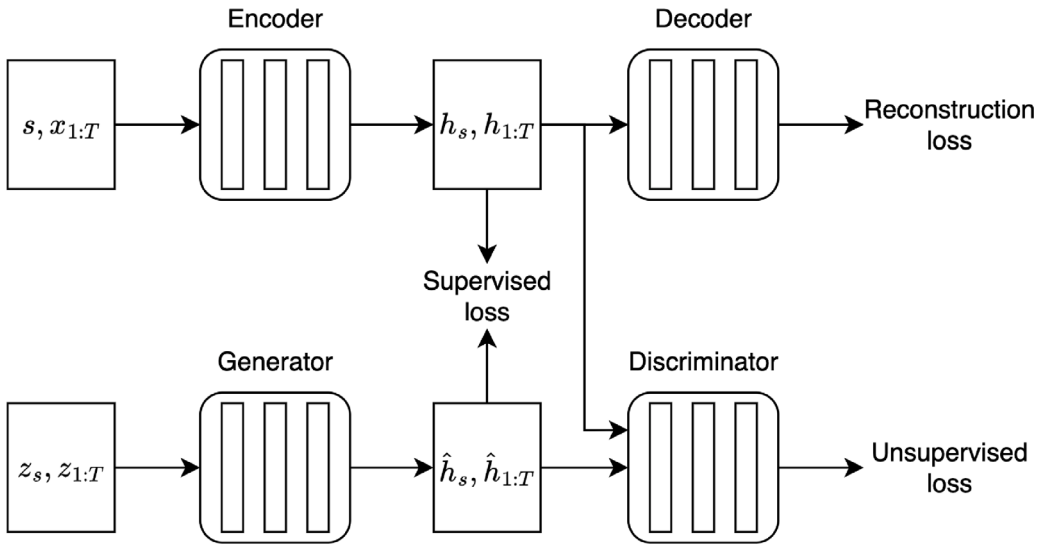


Fig. 4. TimeGAN diagram.

where d_z represents the dimension of the noise vector. The variables $h_{l,nn}$ correspond to the outputs of the lth hidden layer of the neural network nn (g for generator, d for discriminator). The weights and biases for the neural network are denoted by $W_{l,nn}$ and $b_{l,nn}$ respectively.

4.1.2. Loss function

The objective function of the GAN tries to match the real data distribution p_{data} with p_g. The original GAN [10] uses two objective functions. The objective for D is to maximise the probability of assigning the correct label to both real and fake samples. This is done by minimising the negative log-likelihood for binary classification. Simultaneously G is trained to minimise log(1-D(G(z))), thus maximising the probability of the generated samples being classified as real by the discriminator. This results in a mini-max game with objective function V(G,D):

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[log\, D(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))]. \tag{5}$$

The value function V(G,D) is known as the binary cross entropy function, commonly used in binary classification tasks.

4.2. GAN extentions

Many different variations of GANs have been proposed since its inception. In this section different relevant adaptions are presented, ordered by which level of causality they are aiming to improve.

4.2.1. TimeGAN

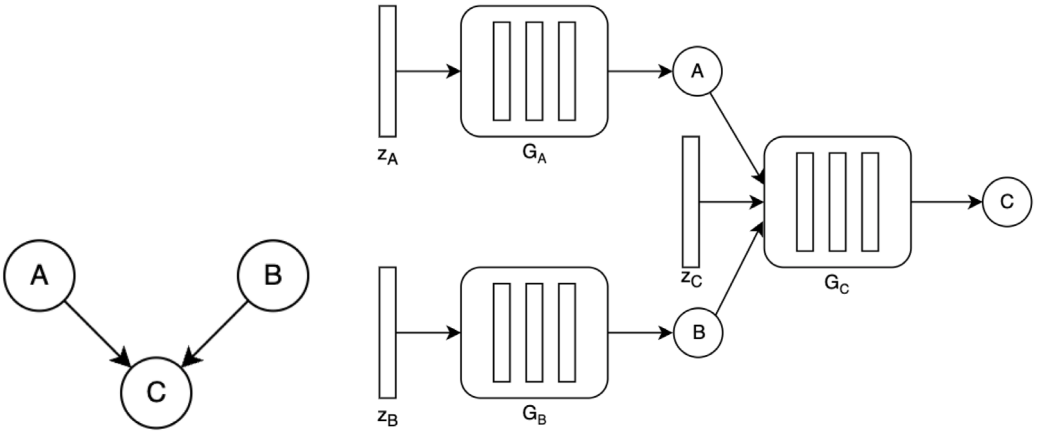
TimeGAN by Yoon et al. [11] is an adaptation of the original GAN framework that aims to improve the preservation of temporal dynamics for time-series data. This means that newly generated sequences should respect the original relationships between variables across time. Two main ideas are combined in the TimeGAN framework, the flexibility of the unsupervised GAN framework and a more controllable supervised autoregressive model. Fig. 4 shows the structure of TimeGAN.

The TimeGAN framework contains the components of a generative adversarial network, as well as an auto-encoder. The latter takes as input a vector of static features, s, and a vector of temporal features, $x_{1:T}$. The encoder is then trained to map the feature space, which s and $x_{1:T}$ belong to, to a latent space. This allows the adversarial network to learn the underlying temporal dynamics of the data via lower-dimensional representations. The output of the encoder are the latent vectors h_s and h_t, being lower-dimensional latent codes of the input s and $x_{1:T}$. In the opposite direction, the decoder takes the static and temporal latent vectors back to their feature representations. The reconstructed static and temporal features are respectively denoted as $\tilde{s}$ and $\tilde{x}_t$.

The other main component in the framework, the generative adversarial network, has a generator that takes as input random noise vectors and outputs latent vectors $\hat{h}_s$ and $\hat{h}_t$. The generator in this framework is autoregressive, meaning it also uses its previous outputs $\hat{h}_{1:t-1}$ for the construction of $\hat{h}_t$. A key difference with a regular GAN architecture is that the generator maps to this latent space instead of the usual feature space. Both the real latent codes h_s and h_t and the synthetic latent codes $\hat{h}_s$ and $\hat{h}_t$ are received by the discriminator, which has the task to classify these codes as either real or fake.

The resulting framework has three loss functions. First, the reconstruction loss:

$$\mathcal{L}_R = \mathbb{E}_{s,x_{1:T}\sim p}[\|s - \tilde{s}\|_2 + \sum_t \|x_t - \tilde{x}_t\|_2]. \tag{6}$$



(a) Causal graph $A \to C \leftarrow B$ (b) Generator architecture for $A \to C \leftarrow B$

Fig. 5. Graphical representation of an example causal graph ($A \to C \leftarrow B$) and the resulting causal generator.

This loss is linked to the auto-encoder component of the framework, quantifying the difference between original features s, x_t and the reconstructed features $\tilde{s}$ and $\tilde{x}_t$.

Second, the unsupervised loss is the same type of loss used in the original GAN framework, maximising (discriminator) or minimising (generator) the likelihood of providing correct classifications. This gives the following loss function:

$$\mathcal{L}_U = \mathbb{E}_{s,x_{1:T}\sim p}[\log y_s + \sum_t \log y_t] + \mathbb{E}_{s,x_{1:T}\sim \hat{p}}[\log(1-\hat{y}_s) + \sum_t \log(1-\hat{y}_t)], \tag{7}$$

where notations y and $\hat{y}$ denote classifications by the discriminator as respectively real or synthetic data.

Lastly, the supervised loss is introduced. The addition of this loss is motivated by the idea that the regular feedback from the discriminator, the unsupervised loss, may be insufficient incentive for the generator to capture the step-wise conditional distributions in the data. To calculate this loss, the autoregressive generator g uses the real latent codes h_s and h_{t-1} instead of the synthetic $\hat{h}_s$ and $\hat{h}_{t-1}$ to generate $\hat{h}_t$, or $g(h_s, h_{t-1}, z_t)$, as shown in the following equation:

$$\mathcal{L}_S = \mathbb{E}_{s,x_{1:T}\sim p}[\sum_t \|h_t - g(h_s, h_{t-1}, z_t)\|_2]. \tag{8}$$

A linear combination of $\mathcal{L}_U$ and $\mathcal{L}_S$ is used to train the generator and the discriminator. $\mathcal{L}_U$ guides the generator to create realistic sequence, while $\mathcal{L}_S$ uses ground-truth targets to ensure that the step-wise transitions are similar. To train the autoencoder components, the encoder and the decoder, a linear combination of $\mathcal{L}_R$ and $\mathcal{L}_S$ is used. By combining the different objectives, TimeGAN is trained to simultaneously encode feature vectors, generate latent codes for these feature vectors, and iterate across time.

4.2.2. CausalGAN

CausalGAN is a generative adversarial framework proposed by Kocaoglu et al. [12]. CausalGAN is an implicit causal generative model that replicates data constraint to a given causal graph. Implicit generative models, which the original GAN model is part of, can sample from a probability distribution, without the ability to provide likelihoods for the samples [59]. Causal implicit generative models can not only sample from a probability distribution but also from conditional and interventional distributions, which causal graphs embeds.

Consider a simple causal graph, $A \to C \leftarrow B$, as depicted in Fig. 5(a). The parent nodes, A and B are assumed to have no other variables influencing their distribution and can be written as $A = G_A(Z_A)$ and $B = G_B(Z_B)$, where Z_* is some chosen noise distribution (e.g. Gaussian), and G_* is a function mapping this distribution to the distribution of the variable. The variable C has two parent nodes and can be written as $C = G_C(A, B, Z_C)$, being a function of both A and B, as well as a chosen distribution. This representation is similar to how the generator of the original GAN framework is structured. Fig. 5(b) shows how a generator can be constructed to represent a given causal graph. For each variable a feedforward neural network is used represent functions G_*, resulting in a larger generator network consisting of linked individual generators. The formal description of a generator for a variable is as follows:

$$G_*(z,c): \begin{cases} \text{Input}: & z \oplus c, \\ \text{Hidden layers}: & h_{1,g} = ReLu(W_{1,g}z + b_{1,g}) \\ & h_{2,g} = ReLu(W_{2,g}h_{1,g} + b_{2,g}), \\ \text{Output}: & G_*(z,c) = W_{o,g}h_{2,g} + b_{o,g}, \end{cases}$$

Table 1
Fitted parameters for all GANs for the OLS model. The OLS model is fitted on the base generated dataset as well as the synthetic datasets generated by GAN, TimeGAN and CausalGAN respectively.

Model	Par.	Real	GAN	TimeGAN	CausalGAN
OLS	β_3	0.9990 ± 0.0051	1.0209 ± 0.0715	0.3762 ± 0.4320	0.9869 ± 0.1087
	β_4	1.0017 ± 0.0052	1.0797 ± 0.1272	1.2249 ± 0.3362	0.9666 ± 0.1029
	β_5	0.9996 ± 0.0057	1.0157 ± 0.1266	1.1066 ± 0.0179	1.0006 ± 0.1625

where the symbol $\oplus$ represents concatenation and c represent the variables that act as causes towards the generator's variable. The primary distinction from the standard GAN implementation is that the generator uses these causes c as additional input. Consider the example in Fig. 5, the generators G_A and G_B for variables A and B have c as an empty set. In the case of variable C, the set of causes c contains the previously generated values $G_A(z_A)$ and $G_B(z_B)$. The result is one generator consisting of several connected smaller generators. The discriminator here is no different than the default GAN implementation. By building in the causal graph into the generator, it will constrain the generated data to the given causal model and not only reproduce joint probabilities, but also the causal relationships. For the implementation of CausalGAN in this paper Causal-TGAN [46] is used. This version uses the same core idea as CausalGAN, with some added adjustments for tabular data.

The downside is that both data and the relevant causal graph needs to be known to train and use the generator. To this end, we use a causal discovery method, in this case the standard- and time-variant of LiNGAM to provide us with the causal graph of the data.

5. Results

Consider the model described in Section 3.4 with the following parameters:

Parameter	Value
α	0.5
$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$	1

where $\epsilon_* \sim N(0, 0.5)$. From this model we sample 10,000 observations to use for further experiments. These observations will further be referred to as generated data. This dataset will be used to train the different GAN models as well as give baseline values for estimated parameters. The experiments assume a perfect scenario where the model is known. For example, it is assumed that x_1 can be modelled using an OLS regression with z_1 and z_2 as explanatory variables. On this dataset multiple causal inference methods are fitted as described in Section 3. The expectation is that the parameters for the model listed above are also found when the causal inference models are fitted to the generated dataset. These same causal inference models are used on the (synthetic) datasets generated by the generative models. The comparison of the results found in these experiments are discussed below for each generative model we consider. Each experiment is done, in its entirety, 10 times and reported results show averages and standard deviations over these 10 runs.

5.1. GAN

First, we train a standard GAN with the generated data described above. From this GAN, we generate 10,000 samples to preserve the statistical power of our inference results. These latter samples will be referred to as the synthetic data. The first model we fit on both datasets is OLS for the following:

$$x_1 = \beta_3 z_1 + \beta_4 z_2 + \epsilon_2,$$
$$x_2 = \beta_5 z_2 + \epsilon_3.$$

The fitted parameters for this OLS model can be seen in Table 1. The results show that on a cross-sectional level, with the underlying model meeting the assumptions in 3.1, the GAN methodology can replicate data with similar causal relationships. We find that the causal relationships identified in the synthetic data generated by the GAN are slightly less accurate compared to those in the original generated data, but the difference is not significant.

While the data the GAN is trained on is time-ordered, the synthetic data produced by the GAN is sampled randomly, without any notion of time. So, as expected, when running an autoregressive model on the y variable in our model, it does not find any time-correlation (α coefficient for y) in the synthetic data. Interestingly, it does capture the cross-sectional relationships for y (β_1 and β_2).

5.2. TimeGAN

Next, TimeGAN is trained on the generated dataset, after which we again sample 10,000 datapoints for a new synthetic dataset. Note that the synthetic data generated by TimeGAN is time-ordered, which was not the case for the data generated by the regular

Table 2
Fitted parameters for all GANs for the autoregressive time series part of the model. The autoregressive model is fitted on the base generated dataset as well as the synthetic datasets generated by GAN, TimeGAN and CausalGAN respectively.

Model	Par.	Real	GAN	TimeGAN	CausalGAN
TS	α	0.5004 ± 0.0011	0.0030 ± 0.0020	0.0233 ± 0.1331	0.0011 ± 0.0064
	β_1	0.9993 ± 0.0040	1.0007 ± 0.1773	1.0597 ± 1.5236	0.9635 ± 0.1896
	β_2	0.9982 ± 0.0045	1.1439 ± 0.1682	0.8796 ± 2.0436	0.9927 ± 0.2035

Table 3
Fitted parameters for TimeGAN using generated data from the alternative model (10).

Model	Parameter	Real	TimeGAN
OLS	β_3	0.9999 ± 0.0002	0.9967 ± 0.0247
	β_4	1.0000 ± 0.0001	1.0049 ± 0.0219
	β_5	0.9999 ± 0.0001	1.0005 ± 0.0024
TS	α	0.4999 ± 0.0008	−0.0128 ± 0.0207
	β_1	1.0000 ± 0.0016	2.0719 ± 0.0680
	β_2	1.0000 ± 0.0016	2.0002 ± 0.1550

GAN. Here the goal is to find the autoregressive term α as well as the cross-sectional terms (β_*) in the following model:

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta_1 x_1 + \beta_2 x_2 + \epsilon_1, \\ x_1 &= \beta_3 z_1 + \beta_4 z_2 + \epsilon_2, \\ x_2 &= \beta_5 z_2 + \epsilon_3. \end{aligned} \tag{9}$$

For completeness, we also fit the autoregressive model to the data generated by the base GAN. The results of this are shown in Table 2. As expected, as there is no time-ordering in the synthetic data produced by the regular GAN, it does not find any time-correlation (α coefficient for y) in the synthetic data. Interestingly, it does capture the cross-sectional relationships for y (β_1 and β_2).

When performing the same experiment with TimeGAN, it is clear that the synthetic data produced by TimeGAN does not properly maintain causal relationships, neither on a cross-sectional level (Table 1) nor over time (Table 2). The results are far from what would be expected and also vary significantly from run to run, resulting in higher standard deviations in the results. This is likely due to there being no auto-correlation in the variables outside of y, and TimeGAN attempting to find time dependent structure where none exists. To confirm this, we also consider the following alternate causal structure, where all variables have some sort of time-dependence (direct or indirect):

$$\begin{aligned} y_t &= \alpha y_{t-1} + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \epsilon_1, \\ x_{1,t} &= \beta_3 z_{1,t} + \beta_4 z_{2,t} + \epsilon_2, \\ x_{2,t} &= \beta_5 z_{2,t} + \epsilon_3, \\ z_{1,t} &= z_{1,t-1} + \epsilon_4, \\ z_{2,t} &= z_{2,t-1} + \epsilon_5. \end{aligned} \tag{10}$$

Table 3 shows the results for TimeGAN in the case of the alternative structure. In this case TimeGAN is able to accurately capture the causal relationships on a cross-sectional level (β_3, β_4, β_5) but still fails to capture the structure in y (α, β_1 and β_2). However, it does not seem like the model completely missed the mark. When we look at the original formulation for y, with the chosen parameters for the experiment, it can be rewritten as follows:

$$\begin{aligned} y_t &= 0.5 y_{t-1} + x_{1,t} + x_{2,t} + \epsilon_t \\ &= 0.25 y_{t-2} + (x_{1,t} + 0.5 x_{1,t-1}) + (x_{2,t} + 0.5 x_{2,t-1}) \\ &\quad + (\epsilon_t + 0.5\epsilon_{t-1}) \\ &= 0.125 y_{t-3} + (x_{1,t} + 0.5 x_{1,t-1} + 0.25 x_{1,t-2}) \\ &\quad + (x_{2,t} + 0.5 x_{2,t-1} + 0.25 x_{2,t-2}) \\ &\quad + (\epsilon_t + 0.5\epsilon_{t-1} + 0.25\epsilon_{t-2}). \end{aligned}$$

This decomposition of y can be continued further until the autoregressive part for y is negligible. Now, if the change in x_1 and x_2 in each time step is limited and thus $x_{1,t} \approx x_{1,t-1}$ and $x_{2,t} \approx x_{2,t-1}$, as is the case here due to the stationarity of y, and using $\sum_{n=0}(\frac{1}{2})^n = 2$, we can write:

$$y_t \approx 2x_{1,t} + 2x_{2,t} + \epsilon,$$

with $\epsilon \sim N(0, \frac{4}{3})$. The results shown in Table 3 thus suggest that TimeGAN has learned this smaller representation of y, using only x_1 and x_2, that results in the same expected values of y over time. This representation, however, does not represent the actual causal model underlying y. This result can be interpreted as TimeGAN having found a shortcut in the way it reconstructs the original data.

Table 4
Causal effects detected by LiNGAM on both the generated dataset and the synthetic dataset generated by CausalGAN. The table contains all significant causal effects (>0.1). Causal effects of less significance (<0.1) are simplified to 0. Bold number indicate that the causal effect is reversed.

Causal effect	Real	CausalGAN	GAN
$z_1 \to x_1$	1.00	0.93	1.03
$z_2 \to x_1$	1.01	0.80	**1.07**
$z_2 \to x_2$	0.99	0.83	**0.16**
$x_1 \to y$	1.02	1.04	**0.14**
$x_2 \to y$	1.01	1.00	**0.39**
$z_1 \to z_2$	0	0	−1.11
$z_1 \to x_2$	0	0	−0.47
$z_1 \to y$	0	0	0.86
$z_2 \to y$	0	0.14	**−0.10**
$x_2 \to x_1$	0	0.14	0.65

5.3. CausalGAN

Lastly, the full structural causal model is considered. Here, a model cannot be directly trained to the data since no such method exists as far as the authors are aware. A two-step approach is taken where first the causal structure is identified with LiNGAM. This extracted structure is then compared to our data generating model (Eq. (3)) to check if LiNGAM is an appropriate and efficient causal discovery method for our case. Then CausalGAN is used to generate data that follows this structure. Lastly, LiNGAM is applied to the synthetic data and its output is compared to the causal structure retrieved from the generated data.

As noted before, LiNGAM uses the assumption of non-Gaussian noise, which is incorrect for model (3) used previously in this section. To start from a correct causal structure for this experiment, we adjust the distribution of the noise our data structure (3) to be uniformly distributed, $\epsilon_* \sim U(-1, 1)$. Under these conditions the time-variant of LiNGAM is able to find the underlying causal model correctly. However, CausalGAN is not equipped to deal with time-series, so we are forced to only consider the cross-sectional causal relations here.

Table 4 shows all causal relationships detected by LiNGAM in both the generated dataset and the synthetic dataset produced by CausalGAN. Additionally, we show the causal relationships detected in synthetic data from a basic GAN trained on the generated data. For this one representative example is chosen since the use of means and standard deviations give warped representations of the results. The synthetic data sampled from CausalGAN consistently maintains causal relationships relatively well. Some deterioration can be seen, as well as introducing small additional causal effects. The basic GAN framework is however not capable of retaining the causal relationships when the whole causal structure is considered. Causal discovery on the synthetic data of the basic GAN gives varying results even when performed multiple times on one synthetic dataset. None of the resulting graphs are close to the original causal graph. This shows that adding the additional information of the (correct) underlying causal graph through the CausalGAN model does help maintaining the causal structure.

6. Real world challenges

In our tests of the causality replicating capabilities of GANs, we have purposely abstracted away from many of the additional challenge that come with working with real-world data. In this section we address some of the most important challenges and give an overview of the variations on the GAN framework that have been proposed to tackle them.

6.1. Computational resources

An aspect of Generative Adversarial Networks that should be mentioned is the need for significant computational resources. One contributing factor is the need for several neural networks. In the base version of GAN, 2 neural networks need to be trained. This increases with more complex frameworks like TimeGAN and CausalGAN which use several additional or larger neural networks. More complex models usually also involve tuning more hyperparameters. While this was not an issue for the data considered in the experiments conducted, it can become computationally expensive for more complex datasets. Overall, while GANs offer powerful capabilities in generative modelling, the significant computational resources required are a key consideration, particularly for applications that demand high fidelity and scalability.

6.2. Privacy

Privacy concerns are one of the main drivers for the recent rise in interest in synthetic data. While in general synthetic data is sampled from a reconstruction of the distribution of the original data, fear of replicating real samples due to overfitting remain [14,60]. Membership inference attacks also form a common concern in the field of privacy [61,62]. These attacks leverage the fact that machine learning models generally perform better on the data it was trained on to reconstruct the training data.

These concerns have sparked the search for GAN variants that give certain privacy guarantees. The models used in the experiments (GAN, TimeGAN and CausalGAN) for example do not offer any such guarantees. One possible privacy guarantee is differential privacy. An algorithm is differentially private if an observer seeing the output cannot tell if a particular datapoint was used in the computation. In the case where the observer has access to the generated samples but not the generator, recent work has shown that the base form of GAN has some privacy guarantees in terms of both differential privacy and robustness to membership inference attacks [63]. These guarantees get stronger for larger training datasets. If additionally the generator is available, several differential privacy GANs have been proposed, such as DPGAN [64], PPGAN [65] and PATE-GAN [13].

Privacy guarantees, however, come at the price of replication quality since you in some form or another adding noise to the data by limiting the impact a training sample can have on the model, even though it might be highly informative [66,67].

6.3. Fairness

Machine learning has an increasingly large impact on current day decision making, scaling decisions made on a micro-scale to a macro-scale in an often opaque manner. This trend has raised concerns about building in, or scaling up biases in decisions. Fairness in machine learning is a recently growing area of research that studies how to ensure that such biases and model inaccuracies do not lead to discriminatory models on the basis of sensitive attributes such as gender or ethnicity. Using synthetic data can help by debiasing the data before it even gets used for further analysis. In such a framework a generative model is trained on unfair data to generate synthetic fair data.

A first challenge to fairness is defining what it actually is, which is often highly dependent on the context of the business decisions that is being made with the model. One often used interpretation is that certain features, also called protected or sensitive features (e.g. gender, ethnicity), should not have any impact on the outcome of the model. This orthogonalisation of the model outcome and the protected features comes with two major challenges. First, it requires outside definition of what the protected features are. Second, if you want to rid observational data of such biases, it is not enough to just delete the features, you need to know the relevant causal structures to exclude both the direct and indirect impact the protected attribute has on the outcome [68,69]. Otherwise the model can just learn the protected features by using different proxies which are correlated to them [70]. CFGAN [71] and DECAF [70] are two methods to generate fair data that are rooted in this approach to fairness. Both methods therefore require a causal graph as additional input, something we saw in our results is not generally feasible with current causal discovery methods.

FairGAN [72] and Fairness GAN [73] have also been suggested for the purpose of generating fair data. FairGAN uses an additional discriminator on top of the classical GAN architecture to determine whether samples are from the protected or unprotected group. Fairness GAN uses an added loss function that encourages demographic parity. Demographic parity is satisfied if the decisions made from the data are not dependent on a given sensitive attribute. This requires a specification of the explanatory variables x, the target variables y and the sensitive variables s, where y does not need specification in other methods. FairGAN is applied to low-dimensional structured data, while Fairness GAN is applied to high dimensional image data.

6.4. Tabular data

Tabular data is data that contains both discrete and continuous columns and is one of the most commonly encountered data formats in both business and research [74]. Tabular data, and especially the discrete features within them are challenging for GAN methods since the continuous functions used in neural nets are ill-equipped to fit the non-continuous distributions of discrete variables.

The generator of a regular GAN cannot generate discrete samples because the generator is trained by the loss from the discriminator via backpropagation [10]. To tackle this problem, MedGAN [75] adds an autoencoder model to the regular GAN framework to generate high-dimensional discrete variables. The TimeGAN model [11] used in the experiments uses one-hot encoding for discrete variables. Both TGAN [74] and TableGAN [76] look to improve the performance on the continuous distributions as well. TGAN clusters numerical variables to deal with the multi-modal distribution for continuous features and adjusts the loss function to effectively generate discrete features. TableGAN uses a classifier neural network to predict synthetic records' labels to improve consistency in generated records. An additional loss, information loss, is introduced as well. This loss is the difference in key statistical values of both the real and synthetic data. In the paper the mean and standard deviation are used as key statistical properties.

Besides the mix of continuous and discrete columns, the distributions of data often differs from the standard Gaussian-like distributions found in typical generative applications like image generation. To this end CTGAN [42] addresses additional concerns about non-Gaussian and multi-modal distributions, and imbalanced categorical columns. CTAB-GAN [77] looks further into these issues and tackles data imbalance and long-tail distributions. The previously mentioned Causal-TGAN [46] combines ideas of CTGAN and CausalGAN [12] to leverage knowledge about the causal structure for a better performance while also being able to handle tabular data.

7. Conclusion

Data has become a driving force in both business, research, and policy, but the increasing use of personal data raises significant privacy and ethical concerns. Regulatory bodies are responding by setting boundaries, but a potential solution is generative modelling: models that generate data to replicate high-dimensional distributions without revealing identifiable information. This

approach supports accurate modelling for predictive tasks (e.g., "If I observe X, what will Y be?") without compromising privacy. However, for interventional questions (e.g., "If I do X, how will Y change?"), where causal inference is needed, synthetic data must not only have the right distribution but also accurately reflect the underlying causal relationships, a challenge current methods struggle to address.

We evaluate how well these causal relationships are replicated by the generative modelling techniques that are typically used for synthetic data. As far as we know, we are the first to do so with a focus on causality. We find that in the case where the assumptions are met that make correlation equal causation, causal inference on the real and synthetic data yield the same results only if the simplest model that can generate the distribution of the features equals the real one. This points at the principle of Occam's razor, that is the foundation for regularisation in machine learning to counter overfitting, is actually working against us in the case where we want to replicate causal relationships. Moreover, we find that for the replication of time series the generative model TimeGAN creates a "shortcut" in the representation. The generated data looks similar to the original data, but the underlying causal structure has changed.

When nothing is known about the causal structure, and the analyst can thus not easily construct a functional form to test with classic causal inference methods like OLS, causal discovery can be used. Causal discovery tries to find the causal structure in observational data, which can then be used as input for a generative model that can generate synthetic data explicitly according to the causal structure. We find that, while this works in simple cases (e.g. in the case of cross-sectional correlation with non-gaussian noise), the necessary assumptions on both the causal discovery and generation side seem too restrictive to be widely applicable in real-world contexts.

A path forward seems to be to augment the observational data fed to the GAN models with additional information such as knowledge on different environments in which the data was collected or interventional data from experiments [39]. While this can present a way forward for many fields, it is often not applicable in the context of businesses related to people's finances or health.

Organisations that want to improve their decision making by leveraging synthetic data should thus be careful about what the current state-of-the-art is actually capable of.

Data availability

No data was used for the research described in the article.

Acknowledgements

The author gratefully acknowledges support from BOF KU Leuven (project C14/21/089).

References

[1] McKinsey & Company, Global survey: The state of AI in 2021, 2021, https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2021. (Accessed 15 December 2023).
[2] European Commission, Regulation of the European parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts, 2021, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52021PC0206.
[3] Gartner Research, Maverick research: Forget about your real data — Synthetic data is the future of AI, 2021, https://www.gartner.com/en/documents/4002912. (Accessed 15 December 2023).
[4] S. Athey, The impact of machine learning on economics, in: The Economics of Artificial Intelligence: An Agenda, University of Chicago Press, 2018, pp. 507–547.
[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.
[7] A. Koenecke, H. Varian, Synthetic data generation for economists, 2020, arXiv preprint arXiv:2011.01374.
[8] N. Taleb, The Black Swan, Random House Trade Paperbacks, 2010, p. 444.
[9] J.L. Fernández-Martínez, Z. Fernández-Muñiz, The curse of dimensionality in inverse problems, J. Comput. Appl. Math. 369 (2020) 112571.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[11] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks, in: NeurIPS, 2019.
[12] M. Kocaoglu, C. Snyder, A.G. Dimakis, S. Vishwanath, Causalgan: Learning causal implicit generative models with adversarial training, 2017, arXiv preprint arXiv:1709.02023.
[13] J. Jordon, J. Yoon, M. Van Der Schaar, PATE-GAN: Generating synthetic data with differential privacy guarantees, in: International Conference on Learning Representations, 2018.
[14] Q. Feng, C. Guo, F. Benitez-Quiroz, A.M. Martinez, When do gans replicate? on the choice of dataset size, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6701–6710.
[15] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of gans for improved quality, stability, and variation, 2017, arXiv preprint arXiv:1710.10196.
[16] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015, arXiv preprint arXiv:1511.06434.
[17] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5907–5915.
[18] H. Wu, S. Zheng, J. Zhang, K. Huang, Gp-gan: Towards realistic high-resolution image blending, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 2487–2495.
[19] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4681–4690.

[20] H. Chen, X. He, H. Yang, J. Feng, Q. Teng, A two-stage deep generative adversarial quality enhancement network for real-world 3D CT images, Expert Syst. Appl. 193 (2022) 116440.
[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[22] D.M. Nguyen, T.P. Le, D.M. Vo, S.-W. Lee, UnfairGAN: An enhanced generative adversarial network for raindrop removal from a single image, Expert Syst. Appl. 210 (2022) 118232.
[23] Y. Mirsky, W. Lee, The creation and detection of deepfakes: A survey, ACM Comput. Surv. 54 (1) (2021) 1–41.
[24] R. He, X. Li, G. Chen, G. Chen, Y. Liu, Generative adversarial network-based semi-supervised learning for real-time risk warning of process industries, Expert Syst. Appl. 150 (2020) 113244.
[25] Q. Chen, W. Wang, K. Huang, S. De, F. Coenen, Multi-modal generative adversarial networks for traffic event detection in smart cities, Expert Syst. Appl. 177 (2021) 114939.
[26] Y. Zou, L. Yu, K. He, Forecasting crude oil risk: A multiscale bidirectional generative adversarial network based approach, Expert Syst. Appl. 212 (2023) 118743.
[27] H. Jung, Y. Cho, G. Ko, J.-i. Song, D. Yu, Comparison study of synthetic data generation methods for credit card transaction data, Korean Data Inf. Sci. Soc. 34 (1) (2023) 49–72.
[28] B. Yilmaz, Housing GANs: Deep generation of housing market data, Comput. Econ. (2023) 1–16.
[29] M.-P. Cote, B. Hartman, O. Mercier, J. Meyers, J. Cummings, E. Harmon, Synthesizing property & casualty ratemaking datasets using generative adversarial networks, 2020, arXiv preprint arXiv:2008.06110.
[30] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant GANs: deep generation of financial time series, Quant. Finance 20 (9) (2020) 1419–1440.
[31] B. Yilmaz, C. Laudagé, R. Korn, S. Desmettre, Electricity GANs: Generative adversarial networks for electricity price scenario generation, Commodities 3 (3) (2024) 254–280.
[32] B. Yilmaz, A scenario framework for electricity grid using generative adversarial networks, Sustain. Energy Grids Netw. 36 (2023) 101157.
[33] C. Esteban, S.L. Hyland, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional gans, 2017, arXiv preprint arXiv:1706.02633.
[34] H. Ni, L. Szpruch, M. Sabate-Vidales, B. Xiao, M. Wiese, S. Liao, Sig-wasserstein GANs for time series generation, in: Proceedings of the Second ACM International Conference on AI in Finance, 2021, pp. 1–8.
[35] J. Fonseca, F. Bacao, Tabular and latent space synthetic data generation: a literature review, J. Big Data 10 (1) (2023) 115.
[36] E. Brophy, Z. Wang, Q. She, T. Ward, Generative adversarial networks in time series: A systematic literature review, ACM Comput. Surv. 55 (10) (2023) 1–31.
[37] S. Beery, G. Van Horn, P. Perona, Recognition in terra incognita, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 456–473.
[38] Y. Wang, H. Wang, Distributionally robust unsupervised domain adaptation, J. Comput. Appl. Math. 436 (2024) 115369.
[39] B. Schölkopf, F. Locatello, S. Bauer, N.R. Ke, N. Kalchbrenner, A. Goyal, Y. Bengio, Toward causal representation learning, Proc. IEEE 109 (5) (2021) 612–634.
[40] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant GANs: deep generation of financial time series, Quant. Finance 20 (9) (2020) 1419–1440.
[41] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223.
[42] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Adv. Neural Inf. Process. Syst. 32 (2019).
[43] S. Athey, G.W. Imbens, J. Metzger, E. Munro, Using wasserstein generative adversarial networks for the design of monte carlo simulations, J. Econometrics (2021).
[44] Y. Gao, L. Shen, S.-T. Xia, DAG-gan: Causal structure learning with generative adversarial nets, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 3320–3324.
[45] A. Sauer, A. Geiger, Counterfactual generative networks, 2021, arXiv preprint arXiv:2101.06046.
[46] B. Wen, L.O. Colon, K. Subbalakshmi, R. Chandramouli, Causal-TGAN: Generating tabular data using causal generative adversarial networks, 2021, arXiv preprint arXiv:2104.10680.
[47] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
[48] A. Kotelnikov, D. Baranchuk, I. Rubachev, A. Babenko, Tabddpm: Modelling tabular data with diffusion models, in: International Conference on Machine Learning, PMLR, 2023, pp. 17564–17579.
[49] Y. Li, X. Lu, Y. Wang, D. Dou, Generative time series forecasting with diffusion, denoise, and disentanglement, Adv. Neural Inf. Process. Syst. 35 (2022) 23009–23022.
[50] M. Arjovsky, L. Bottou, I. Gulrajani, D. Lopez-Paz, Invariant risk minimization, 2019, arXiv preprint arXiv:1907.02893.
[51] X. Zheng, B. Aragam, P.K. Ravikumar, E.P. Xing, Dags with no tears: Continuous optimization for structure learning, Adv. Neural Inf. Process. Syst. 31 (2018).
[52] M.J. Vowels, N.C. Camgoz, R. Bowden, D'ya like dags? A survey on structure learning and causal discovery, ACM Comput. Surv. 55 (4) (2022) 1–36.
[53] S. Shimizu, P.O. Hoyer, A. Hyvärinen, A. Kerminen, M. Jordan, A linear non-Gaussian acyclic model for causal discovery, J. Mach. Learn. Res. 7 (10) (2006).
[54] P.O. Hoyer, S. Shimizu, A.J. Kerminen, M. Palviainen, Estimation of causal effects using linear non-Gaussian causal models with hidden variables, Internat. J. Approx. Reason. 49 (2) (2008) 362–378.
[55] A. Hyvärinen, K. Zhang, S. Shimizu, P.O. Hoyer, Estimation of a structural vector autoregression model using non-gaussianity, J. Mach. Learn. Res. 11 (5) (2010).
[56] K. Zhang, A. Hyvärinen, On the identifiability of the post-nonlinear causal model, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, AUAI Press, Arlington, Virginia, USA, 2009, pp. 647–655.
[57] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, 2018, arXiv preprint arXiv:1809.11096.
[58] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.
[59] S. Mohamed, B. Lakshminarayanan, Learning in implicit generative models, 2016, arXiv preprint arXiv:1610.03483.
[60] R. Webster, J. Rabin, L. Simon, F. Jurie, Detecting overfitting of deep generative networks via latent recovery, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11273–11282.
[61] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, Logan: Membership inference attacks against generative models, 2017, arXiv preprint arXiv:1705.07663.
[62] D. Chen, N. Yu, Y. Zhang, M. Fritz, Gan-leaks: A taxonomy of membership inference attacks against generative models, in: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, pp. 343–362.
[63] Z. Lin, V. Sekar, G. Fanti, On the privacy properties of gan-generated samples, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1522–1530.
[64] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network, 2018, arXiv preprint arXiv:1802.06739.

[65] Y. Liu, J. Peng, J. James, Y. Wu, PPGAN: Privacy-preserving generative adversarial network, in: 2019 IEEE 25Th International Conference on Parallel and Distributed Systems, ICPADS, IEEE, 2019, pp. 985–989.
[66] C. Huang, P. Kairouz, X. Chen, L. Sankar, R. Rajagopal, Context-aware generative adversarial privacy, Entropy 19 (12) (2017) 656.
[67] Z. Lin, A. Jain, C. Wang, G. Fanti, V. Sekar, Using gans for sharing networked time series data: Challenges, initial promise, and open questions, in: Proceedings of the ACM Internet Measurement Conference, 2020, pp. 464–483.
[68] M.J. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual fairness, Adv. Neural Inf. Process. Syst. 30 (2017).
[69] J. Zhang, E. Bareinboim, Fairness in decision-making—the causal explanation formula, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[70] B. van Breugel, T. Kyono, J. Berrevoets, M. van der Schaar, Decaf: Generating fair synthetic data using causally-aware generative networks, Adv. Neural Inf. Process. Syst. 34 (2021) 22221–22233.
[71] D. Xu, Y. Wu, S. Yuan, L. Zhang, X. Wu, Achieving causal fairness through generative adversarial networks, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019.
[72] D. Xu, S. Yuan, L. Zhang, X. Wu, Fairgan: Fairness-aware generative adversarial networks, in: 2018 IEEE International Conference on Big Data, Big Data, IEEE, 2018, pp. 570–575.
[73] P. Sattigeri, S.C. Hoffman, V. Chenthamarakshan, K.R. Varshney, Fairness gan, 2018, arXiv preprint arXiv:1805.09910.
[74] L. Xu, K. Veeramachaneni, Synthesizing tabular data using generative adversarial networks, 2018, arXiv preprint arXiv:1811.11264.
[75] E. Choi, S. Biswal, B. Malin, J. Duke, W.F. Stewart, J. Sun, Generating multi-label discrete patient records using generative adversarial networks, in: Machine Learning for Healthcare Conference, PMLR, 2017, pp. 286–305.
[76] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, Y. Kim, Data synthesis based on generative adversarial networks, 2018, arXiv preprint arXiv:1806.03384.
[77] Z. Zhao, A. Kunar, R. Birke, L.Y. Chen, Ctab-gan: Effective table data synthesizing, in: Asian Conference on Machine Learning, PMLR, 2021, pp. 97–112.