Applied Intelligence (2025) 55:602
https://doi.org/10.1007/s10489-025-06410-8

DAGAF: A directed acyclic generative adversarial framework for joint structure learning and tabular data synthesis

Hristo Petkov[1] · Calum MacLellan[1] · Feng Dong[1]

Accepted: 23 February 2025

Abstract
Understanding the causal relationships between data variables can provide crucial insights into the construction of tabular datasets. Most existing causality learning methods typically focus on applying a single identifiable causal model, such as the Additive Noise Model (ANM) or the Linear non-Gaussian Acyclic Model (LiNGAM), to discover the dependencies exhibited in observational data. We improve on this approach by introducing a novel dual-step framework capable of performing both causal structure learning and tabular data synthesis under multiple causal model assumptions. Our approach uses Directed Acyclic Graphs (DAG) to represent causal relationships among data variables. By applying various functional causal models including ANM, LiNGAM and the Post-Nonlinear model (PNL), we implicitly learn the contents of DAG to simulate the generative process of observational data, effectively replicating the real data distribution. This is supported by a theoretical analysis to explain the multiple loss terms comprising the objective function of the framework. Experimental results demonstrate that DAGAF outperforms many existing methods in structure learning, achieving significantly lower Structural Hamming Distance (SHD) scores across both real-world and benchmark datasets (Sachs: 47%, Child: 11%, Hailfinder: 5%, Pathfinder: 7% improvement compared to state-of-the-art), while being able to produce diverse, high-quality samples.

Keywords Adversarial causal discovery · Tabular data synthesis · Directed acyclic graph learning · Post-nonlinear model · Additive noise model · Linear on-gaussian acyclic model

1 Introduction

Understanding causal relationships between variables in a dataset is a crucial aspect of data analysis, as it can lead to numerous scientific discoveries. Although randomized controlled trials, which involve manipulating data through interventions, are still considered the gold standard for learning causal structures, such experiments are often impractical or even impossible due to many ethical, technical, or resource constraints. Addressing this challenge has led to a growing demand for causal studies to identify causal relationships from passive observational data.

In the last few decades, numerous approaches have emerged for performing observational causal discovery across various scientific fields, including bioinformatics [1–3], economics [4], biology [5, 6], climate science [7, 8], and social sciences [9]. Most causal studies employ conditional independence-based algorithms, such as PC [10], FCI [11], and RFCI [12]; discrete score-based methods like GES [13], GES-mod [14], and GIES [15]; or continuous optimization techniques, including NOTEARS [16], DAG-GNN [17], GraN-DAG [18], and DAG-WGAN [19]. All these methodologies for causal structure learning have been rigorously tested and demonstrated substantial empirical evidence of their ability to produce accurate graphical representations of dependencies within datasets. However, strong performance does not necessarily resolve the issue of non-uniqueness in causal models, where multiple causal graphs can be used to define the same distribution.

To resolve the issue of non-uniqueness in causal models (e.g. Markov equivalent), where a single observed dataset

Calum MacLellan and Feng Dong contributed equally to this work.

✉ Hristo Petkov
hristo.petkov@strath.ac.uk

Calum MacLellan
calum.maclellan@strath.ac.uk

Feng Dong
feng.dong@strath.ac.uk

[1] Department of Computer and Information Sciences, University of Strathclyde, 16 Richmond Street, Glasgow, Lanarkshire G1 1XQ, United Kingdom

Published online: 31 March 2025

may have multiple underlying structures, researchers often introduce additional assumptions [20]. They employ Functional Causal Models (FCM) parameterized with various structural equations to ensure that a unique causal graph is identified from a given distribution. Currently, there exist a significant amount of works that apply various identifiable (in most cases) models to learn causal structures from observational data. Noteworthy examples include the extensively researched linear non-Gaussian acyclic model (LiNGAM) [21], the additive noise model (ANM) [22], which provides limited support for non-linearity by assuming the relationships between variables are additive and the post-nonlinear model (PNL) [23] designed for studying complex non-linear relationships.

Among the aforementioned FCMs, the post-nonlinear (PNL) model is notable for being realistic and more accurately representing the sensor or measurement distortions commonly observed in real-world data [24]. It is also considered a superset that encompasses both ANM and LiNGAM. The PNL model consists of two functions: 1) an initial function that transforms data variables, with independent noise subsequently added to all transformations; and 2) an invertible function that applies an additional post-nonlinear transformation to each variable. Although the PNL model is one of the most general FCMs for modeling causal mechanisms in real data distributions, it is less studied than other identifiable models due to challenges associated with its post-nonlinearity and invertibility constraints.

Several approaches have been developed to investigate causal structure learning under the assumption of post-nonlinear (PNL) models, with most focusing on accurately approximating the invertibility function. For example, AbPNL [25] uses an autoencoder architecture to learn a function and its inverse by minimizing a combination of independence and reconstruction loss terms. This model is applied to both bivariate and multivariate causal discovery within the context of PNL. Another approach, DeepPNL [26], parameterizes both functions of the PNL model using deep neural networks. Similarly, CAF-PoNo [27] employs normalizing flows to model the invertibility constraint associated with PNL. Rank-PNL, proposed by [28], adapts rank-based methods to estimate the invertible function of the causal model. The latest work in this area, MC-PNL [29], aims to efficiently perform structure learning for PNL estimation by modeling nonlinear causal relationships using a novel objective function and block coordinate descent optimization. Despite recent advances in PNL estimation, causal structure learning under this functional causal model assumption remains relatively unexplored compared to other models such as ANM.

Most existing causality learning methods typically focus on applying a single identifiable causal model to discover the dependencies exhibited in observational data. This presents a significant disadvantage as such approaches have no way to determine whether the model they assumed can learn an accurate representation of the underlying structure in a dataset. This is a critical problem to address, as misidentification of causal relationships in a dataset can result in incorrect data analysis, leading to bias in classification or inaccurate predictions. Moreover, causal discovery is also closely related to tabular data synthesis, where externally learned causal mechanisms are applied in Deep Generative Models (DGM) (e.g. DECAF [30], Causal-TGAN [31] and TabFairGAN [32]) to synthesize new data samples. This method has its limitations because the accuracy of the causal knowledge must be evaluated prior to its application, which requires the availability of the true underlying structure of the dataset. This assumption proves to be impractical for real-world data, as such datasets are usually complex and extensive, with their causal structures often remaining unknown.

Recent advancements in generative modeling, including Digital Twins and transformer-based multi-attention networks, provide alternative approaches for modeling complex data relationships. Digital Twin models aim to create virtual representations of real-world systems, making them highly relevant for synthetic data generation. Similarly, attention-based architectures, such as multi-attention networks, dynamically weigh dependencies between variables. As generative models continue to gain popularity, there is significant potential to integrate them with causal discovery under a unified framework, enabling more accurate and interpretable data generation that remains faithful to underlying causal structures.

In this paper, we aim to address some of the challenges outlined above by proposing a novel framework called DAGAF, which is capable of modeling causality resembling the underlying causal mechanisms of the input data (i.e learnable causal structure approximations) and employing them to synthesize diverse, high-fidelity data samples. DAGAF learns multivariate causal structures by applying various functional causal models and determines through experimentation which one best describes the causality in a tabular dataset. Specifically, the framework supports the PNL model along with its subsets, which include LiNGAM and ANM. Unlike other methods that assume data generation is limited to a single causal model, DAGAF satisfies multiple semi-parametric assumptions. Additionally, supporting such a broad spectrum of identifiable models enables us to extensively compare our approach against the state-of-the-art in the field. We complete our study by investigating the quality of the discovered causality from a tabular data generation standpoint. We hypothesize that a precise approximation of the original causal mechanisms in a given probability distribution can be leveraged to produce realistic data samples. To prove our hypothesis, DAGAF incorporates an adversarial tabular data synthesis step, based on transfer learning, into our causal discovery framework.

The contributions made throughout this work are outlined as follows:

- We unify causal structure learning and tabular data synthesis under a single framework capable of approximating the generative process of observational data and producing realistic samples. This approach allows us to generate quality synthetic data from the input, while preserving its causality (Section 3).
- The proposed framework seamlessly integrates ANM, LiNGAM, and PNL models by leveraging a multi-objective loss function that combines adversarial loss, reconstruction loss, KL divergence, and MMD. This flexible formulation enables robust causal structure learning under diverse data-generating assumptions. Additionally, we provide a theoretical analysis to elucidate the contributions of these loss terms and how they complement each other in guiding convergence toward the true causal structure. We also analyze causal identifiability, providing conditions under which causal relationships can be uniquely determined, and examine how real-world data characteristics-such as noise, missing values, and distribution shifts-can impact identifiability (Sections 3.1 and 4).
- We employ transfer learning in the context of causally-aware tabular data synthesis. DAGAF uses a two-step iterative approach that combines causal knowledge acquisition with high-quality data generation. The causal relationships identified in the first step are transferred and leveraged in the second step to facilitate causal-based tabular data generation. This enables more faithful synthetic data generation, preserving the underlying causal mechanisms (Section 3.2).
- We validate the effectiveness of DAGAF on synthetic, benchmark, and real-world datasets. Our results show significant improvement in DAG learning in comparison with other methods (Sachs: 47%, Child: 11%, Hailfinder: 5%, Pathfinder: 7% improvement compared to state-of-the-art). They also demonstrate that the learned causal mechanism approximations can be used to generate high-quality, realistic data (Section 5).

2 Prerequisites

This section explores the mathematical aspects of causality, relevant to the field of machine learning. In particular, we provide a brief overview of Functional Causal Models (FCM) [33] and the assumptions employed in our causal structure learning algorithm.

Let χ denote a tabular dataset such that $\mathbf{X} = \{X_1, \ldots, X_d\}$ is a set of d random data variables, and $\chi \subseteq \mathbb{R}^{n \times d}$ represents a dataset consisting of n samples $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ drawn from the joint distribution $P(\mathbf{X})$. Individual data points and their attributes are denoted as $\mathbf{X}_i$ and X_j, respectively. Additionally, let $\mathcal{G}_\mathcal{A} \in \mathbb{D}$ be a ground truth Directed Acyclic Graph (DAG) representing the relationships between all the attributes $\{X_1, \ldots, X_d\}$. Then, $P(\mathbf{X})$ can be expressed using a functional causal model (FCM), which describes the relationships within $\{X_1, \ldots, X_d\}$. In this context, FCMs facilitate causal discovery from tabular datasets by encoding variables as nodes, and edges between them represent the underlying causal mechanisms responsible for data generation.

According to theory, an FCM is formulated as a triplet $\mathcal{M}_{\mathcal{G}_\mathcal{A}}\langle \mathbf{X}, \mathcal{F}, \mathcal{Z} \rangle$, where $\mathbf{X} = \{X_1, \ldots, X_d\}$ is a set of endogenous variables, $\mathcal{F} = \{f_1, \ldots, f_d\}$ is a set of structural equations, and $\mathcal{Z} = \{\mathcal{Z}_1, \ldots, \mathcal{Z}_d\}$ is a set of exogenous (noise) variables. Under the local Markov condition and the causal sufficiency assumption, the joint distribution of $\mathbf{X}$ can be factorized as $P(\mathbf{X}) = \prod_{j=1}^{d} P(X_j \mid Pa_j)$, where X_j is a child of its parent variables Pa_j in the graph $\mathcal{G}_\mathcal{A}$. Each X_j can be modeled in its non-parametric form as:

$$X_j := f_j(\mathrm{Pa}_j, \mathcal{Z}_j). \tag{1}$$

This representation of $P(\mathbf{X})$ allows us to sequentially model the causal mechanisms underlying χ, defining its generative process.

Furthermore, we assume faithfulness, which enables the discovery of causal structures from continuous observational data using various nonlinear and semi-parametric models. Our framework is applied to several types of models, including: Linear non-Gaussian Acyclic Models (LiNGAM), Additive Noise Models (ANM), and Post-Nonlinear Models (PNL). Each of these models has been proven to be causally identifiable under specific assumptions:

- **LiNGAM**: The causal identifiability of LiNGAM is guaranteed under the assumption of non-Gaussianity in the noise terms. Specifically, if the noise variables are non-Gaussian and independent from X, it has been shown that the underlying causal structure can be uniquely identified [21].
- **ANM**: Additive Noise Models (ANM) assume that the Gaussian noise term $\mathcal{Z}_j$ is independent of the parent variables Pa_j. This assumption enables the identifiability of the causal direction between variables. Additionally, the function $f_j(\cdot)$ must be non-linear and three times differentiable, to ensure that the application of this model results in a unique determination of the causal direction between variables [22].
- **PNL**: Post-Nonlinear Models (PNL) extend the ANM framework by introducing an additional non-linear transformation $g_j(\cdot)$ after the function $f_j(\cdot)$. The key assumptions for identifiability in PNL include the independence

of the Gaussian noise terms and the non-linear and invertible nature of the function $g_j(\cdot)$. Under these conditions, the causal structure can be identified, even in the presence of complex non-linear interactions [23].

3 DAGAF: A general framework for simultaneous causal discovery and tabular data synthesis

DAGAF learns DAG structures from input data to simulate the generative process of their probability distribution. We model G_A to represent the causal relationships within a dataset χ. The model is capable of facilitating realistic sample synthesis with minimal loss of fidelity and diversity. We formalize our goal as follows.

Goal Given n i.i.d. observations $\mathbf{X} \sim P(\mathbf{X}) \in \chi$, we propose a general framework to learn $G_A \approx \mathcal{G}_\mathcal{A} \in \mathbb{D}$ together with a set of structural equations $\mathcal{F} = \{f_1, ...f_d\}$, such that $\tilde{X}_j := f_j(Pa_j, \mathcal{Z}_j)$ yields $\tilde{\mathbf{X}} \sim P_{G_A}(\tilde{\mathbf{X}}) \in \tilde{\chi}$ matching the input.

The DAGAF framework focuses on learning an approximation of the causal mechanisms $\{f_j(Pa_j, \mathcal{Z}_j)\}$ involved in the generation of observations $\mathbf{X}$. The (semi)parametric assumptions outlined in Section 2 define each node $X_j \in G_A$ as a function $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$. Under such circumstances, the general nonparametric form $\mathbb{E}[X_j|X_{pa(j)}] := \mathbb{E}_\mathcal{Z}(f_j(X, \mathcal{Z}))$ can be reduced to one of the following: 1) **Linear non-Gaussian Acyclic Models (LiNGAM)**: $\tilde{X} := f(X) + \mathcal{Z}$, where $f(X)$ is a linear function of X and $\mathcal{Z}$ is a non-Gaussian noise term independent of X; 2) **Additive Noise Models (ANM)**: $\tilde{X}_j := f_j(Pa_j) + \mathcal{Z}_j$, where f_j is a nonlinear function of the parent variables Pa_j, and $\mathcal{Z}_j \perp\!\!\!\perp f_j(Pa_j)$, $\mathcal{Z}_j \sim \mathcal{N}(\mu, \sigma_j^2)$; 3) **Post-Nonlinear Models (PNL)**: $\tilde{X}_j := g_j(f_j(Pa_j)+\mathcal{Z}_j)$, where g_j is a nonlinear function and $\mathcal{Z}_j \perp\!\!\!\perp f_j(Pa_j)$, $\mathcal{Z}_j \sim \mathcal{N}(\mu, \sigma_j^2)$.

Algorithm 1 provides an overview of the training process. Section 3.1 details Step 1, which focuses on causal structure learning. Furthermore, since the framework recovers the causal structure by learning the underlying data generative process of $\mathbf{X}$, it is naturally well-suited for data synthesis. However, it requires training a separate Deep Generative Model (DGM) involving a discriminator and a generator in an additional training phase, which is explained in detail in Section 3.2. The architecture and training procedure of DAGAF are described in Section 3.3. A visual representation of the model pipeline is provided in Fig. 1.

Algorithm 1 DAGAF training algorithm.

Require: Sample n observational data points $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ from the training data and d noise vectors $\{Z_1, \ldots, Z_d\}$ from normal distributions. Generate n synthetic data samples $\{\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n\}$, with data attributes $\tilde{X} := f(X) + \mathcal{Z}$, $\tilde{X}_j := f_j(Pa_j) + \mathcal{Z}_j$ or $\tilde{X}_j := g_j(f_j(Pa_j) + \mathcal{Z}_j)$ depending on whether LiNGAM, ANM or PNL is assumed.

Ensure: The acyclicity constraint value $h(A^{L_0}(f))$ is higher than its tolerance of error h_tol set to 1e-8. Each step during training has its own instance of DAG-Notears-MLP. Causal information is transferred from the FCM into the DGM architecture.

Step 1: Poly-assumptive causal structure learning
LiNGAM, ANM → learn f by minimizing a combination of loss terms including
adversarial loss (2), Mean Squared Error (3), Kullback-Lieber divergence (4),
Maximum Mean Discrepancy (5) and the acyclicity constraint from [34]
PNL → learn f using the loss terms described in the LiNGAM, ANM case and
g^{-1} by solving (8)
This step recovers a graph representation G_A of the causal mechanisms in $\mathbf{X}$.

Step 2: Generative process simulation under multiple causal model assumptions
LiNGAM, ANM → learn f by computing (2)
PNL → learn f and g by finding the optimal value for (2)
This step models a generative process involving G_A through adversarial
training, producing new data samples.

3.1 Loss functions for causal structure learning

In Step 1 of DAGAF training, the goal is to model DAGs using a sophisticated objective function that integrates a combination of loss terms used for causal structure learning. In its basic form, the framework covers LiNGAM and ANM by utilizing adversarial training and reconstruction loss, along with some regularization terms, to learn how to generate $\tilde{\mathbf{X}}$ from $\mathbf{X}$. One benefit of our framework is its flexibility, allowing the basic approach to be easily adapted to support causal structure learning using PNL. The advanced form further extends the functionality of the framework to cover PNL by adding an additional reconstruction loss term to model the non-linear function g_j.

3.1.1 Adversarial loss with gradient penalty

DAGAF simulates $\mathbf{X}$ by learning how to generate $\tilde{\mathbf{X}}$ using causal mechanism approximations of $\{f_j(Pa_j, \mathcal{Z}_j)\} \in P(\mathbf{X})$. To achieve this, we do not directly model $\tilde{\mathbf{X}}$ but instead focus on recovering the causal mechanisms $\mathcal{F} = \{f_1, \ldots, f_d\}$, where each f_j is defined as $f_j(Pa_j; W_j^1, \ldots, W_j^L) + \mathcal{Z}_j$. Learning the causal mechanisms involves determining the immediate parents of each variable, which are encoded in the causal structure of $\mathbf{X}$. We minimize the Wasserstein distance $\mathbb{W}_p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}}))$ through adversarial training, which implicitly refines the causal structure

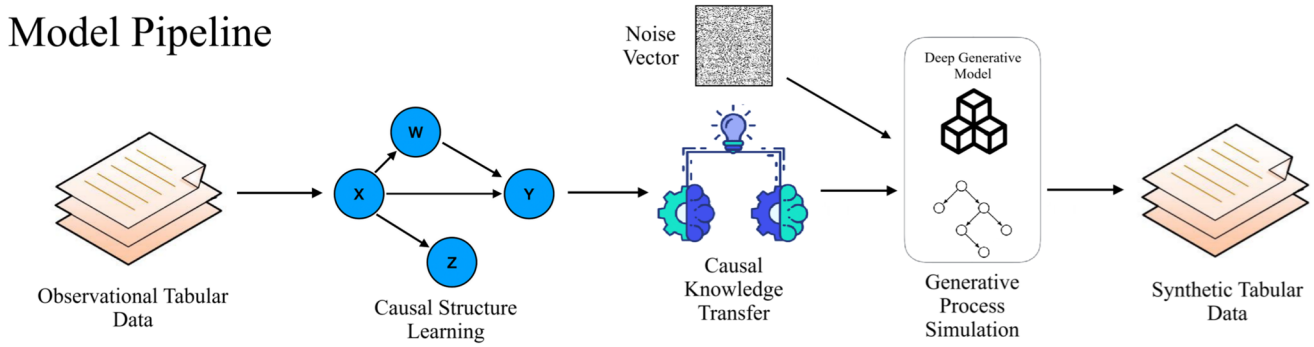


Fig. 1 Pipeline of the DAGAF algorithm

G_A, facilitating the identification of the causal mechanisms. The Wasserstein distance with gradient penalty loss term is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}}) &= \sup_{\|\phi\|_L \leq 1} \mathbb{E}_{\mathbf{X}\sim P(\mathbf{X})}[\phi(\mathbf{X})] - \mathbb{E}_{\tilde{\mathbf{X}}\sim P_{G_A}(x|G)}[\phi(\tilde{\mathbf{X}})] \\ &= \mathbb{E}_{\mathbf{X}\sim P(\mathbf{X})}[D(\mathbf{X})] - \mathbb{E}_{\tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[D(\tilde{\mathbf{X}})] \\ &\quad + \mathbb{E}_{\hat{\mathbf{X}}\sim P(\hat{\mathbf{X}})}[(||\nabla_{\hat{\mathbf{X}}} D(\hat{\mathbf{X}})||_2 - 1)^2], \end{aligned} \tag{2}$$

where $\phi(\mathbf{X})$ is a 1-Lipschitz function used to approximate the Wasserstein distance $\mathbb{W}p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}}))$. The function $D(\mathbf{X})$ serves as the discriminator, which is trained adversarially to learn $\phi(\mathbf{X})$ and distinguish between real and generated samples.

In this framework, adversarial training to optimise (2) involves learning the set of structural equations $\mathcal{F} = \{f_1, \ldots, f_d\}$, where each f_j models the causal mechanism of node X_j. The FCM-based generator $\mathcal{M}$ learns to generate synthetic data that mimics the true distribution, while the discriminator $D(\mathbf{X})$ evaluates the divergence between real and generated samples. The objective is formulated as a min-max optimization, where $\mathcal{M}$ aims to minimize the discrepancy measured by $D(\mathbf{X})$, while $D(\mathbf{X})$ is trained to distinguish between real and generated distributions, typically using the Wasserstein distance. Theoretically, this min-max optimization problem achieves its optimal point typically characterized as a Nash equilibrium, when the generator can yield synthetic data that is indistinguishable from $\mathbf{X}$, thereby approximating the generative process of $\mathbf{X}$ (i.f.f. the causal structure in G_A is correctly identified).

Proposition 1 *Let the ground-truth DAG $\mathcal{G}_{\mathcal{A}}$ be uniquely identifiable from $P(\mathbf{X})$, then minimizing the adversarial loss ensures that the implicitly generated distribution $P_{G_A}(\tilde{\mathbf{X}})$ aligns with $P(\mathbf{X})$.*

$$\inf_{G_A \in \mathbb{D}} \mathcal{L}_{adv}(\mathbf{X}, \tilde{\mathbf{X}}) = 0 \implies P_{G_A}(\tilde{\mathbf{X}}) = P(\mathbf{X}) \quad \textit{and consequently} \quad G_A = \mathcal{G}_{\mathcal{A}}.$$

Proof The proof of Proposition 1 is available in Appendix A.1. □

3.1.2 Reconstruction loss

We add a reconstruction loss to enhance causal structure learning. In this context, we use Mean Squared Error (MSE) as the reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) = \mathbb{E}_{\mathbf{X},\tilde{\mathbf{X}}}(||\mathbf{X} - \tilde{\mathbf{X}}||_2) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} ||X_{ij} \\ - \{f_j(Pa_j; W_j^1, ..., W_j^L) + \mathcal{Z}_j\}_i||_2 \end{aligned} \tag{3}$$

By reducing (3) through parameter optimization, we minimize the residual distance between individual samples $||\mathbf{X} - \tilde{\mathbf{X}}||$ such that our model produces $\tilde{\mathbf{X}} \sim P_{G_A}(\tilde{\mathbf{X}})$ by implicitly learning the causal dependencies of $\mathbf{X}$ represented in G_A. Essentially, this reconstruction process results in a closer approximation of the causal mechanisms responsible for producing $\mathbf{X}$.

Proposition 2 *The MSE loss ensures point-wise alignment between the data and the prediction of the model, improving the smoothness of the gradient and the stability of adversarial optimization.*

$$\inf_{G_A \in \mathbb{D}} \mathcal{L}_{MSE}(\mathbf{X}, \tilde{\mathbf{X}}) = 0 \Rightarrow \forall i, \tilde{\mathbf{X}}_i = \mathbf{X}_i$$

Proof The proof of Proposition 2 is available in Appendix A.2. □

The MSE loss plays a key role in DAG learning, as evidenced by our experiments. This aligns with the approach taken by most existing works in DAG-learning, where MSE is the most commonly used loss function.

3.1.3 Kullback-Leibler divergence

We introduce Kullback-Leibler divergence (KLD) [35] as a regularization term for nonlinear cases with additive Gaussian noise in ANM to prevent overfitting of $\mathbf{X}$ and inaccurate causal mechanisms in the generative process of $\tilde{\mathbf{X}}$. The KLD term is typically applied in Variational

Autoencoders (VAE) as a regularization component of the Evidence Lower Bound (ELBO) loss function for latent variables. It is defined as $D_{KL}\left(\mathcal{N}(\mu, \sigma^2) \| \mathcal{N}(0,1)\right) = \frac{1}{2}\sum_{i=1}^{n}\left(\sigma_i^2 + \mu_i^2 - \log(\sigma_i^2) - 1\right)$ where μ and σ denote the mean and standard deviation of $\tilde{\mathbf{X}}$. In our setup, we apply this to regularize $\tilde{\mathbf{X}}$. Additionally, we only model the mean of $P_{G_A}(\tilde{\mathbf{X}})$ and set its variance to 1, hence reducing the regularization function to:

$$\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) = D_{KL}(P(\mathbf{X}) || P_{G_A}(\tilde{\mathbf{X}})) = \frac{1}{2}\sum_{i=1}^{n}(\mu_i^2). \quad (4)$$

We use the Kullback-Leibler divergence (KLD) as a regularization term for $\tilde{\mathbf{X}}$, the model-generated data, to simulate an additive noise scenario where noise is incorporated into each data point. By applying KLD to $\tilde{\mathbf{X}}$, we encourage the model to produce $\tilde{\mathbf{X}}$ that closely matches the true data distribution while accounting for the variability introduced by noise. This regularization helps the model avoid overfitting by ensuring that the generated data reflects the natural variations present in the real data, leading to more robust and realistic samples. As our model involves learning causal mechanisms, this prevents the model from learning incorrect causal structures, such as misidentifying child nodes as parent nodes.

Proposition 3 *The $\mathcal{L}_{KLD}(\mathbf{X}, \tilde{\mathbf{X}})$ regularization provides a statistical prior on the learned distribution $P_{G_A}(\tilde{\mathbf{X}})$, ensuring it adheres to a Gaussian assumption. It also acts as a stabilizing factor in optimization, particularly under the additive Gaussian noise model. It complements the adversarial and MSE losses, ensuring both alignment and smoothness of $P_{G_A}(\tilde{\mathbf{X}})$.*

Proof The proof of Proposition 3 is available in Appendix A.3. □

Note, this is not applicable to the LiNGAM causal model, due to the non-Gaussianity of the noise term $\mathcal{Z}$ under that specific assumption.

3.1.4 Maximum mean discrepancy

The reconstruction loss and its regularization term focus solely on learning the mean of $P(\mathbf{X})$, while completely disregarding its variance. This implies that the reconstruction process involved in DAGAF is highly sensitive to rare occurrences (i.e. outliers) in $P(\mathbf{X})$. To address this issue, we further reduce the residual distance between the input distribution $\mathbf{X} \sim P(\mathbf{X})$ and the generated data distribution $\tilde{\mathbf{X}} \sim P_{G_A}(\tilde{\mathbf{X}})$ by introducing the Maximum Mean Discrepancy (MMD) [36]. We apply the kernel trick [37] to compute the solution to this formula.

$$\begin{aligned}\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) &= ||\mathbb{E}_{\mathbf{X}\sim P(\mathbf{X})}[k(\mathbf{X})] - \mathbb{E}_{\tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[k(\tilde{\mathbf{X}})]||^2_{\mathcal{H}} \\ &= \frac{1}{n}\sum_{i\neq j}^{n} k(\mathbf{X}_i, \mathbf{X}_j) - \frac{2}{n}\sum_{i\neq j}^{n} k(\mathbf{X}_i, \tilde{\mathbf{X}}_j) + \frac{1}{n}\sum_{i\neq j}^{n} k(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j),\end{aligned} \quad (5)$$

where $\mathcal{H}$ denotes the reproducing kernel Hilbert space (RKHS) and $k \in \mathcal{H}$ is a kernel function.

The MMD maximizes mutual information between $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$, leading to alignment in both their means and overall shapes. Specifically, by matching the shapes of the distributions, the MMD term can help bring their variances closer together. Hence, by applying (5) we indirectly model the standard deviation of $P_{G_A}(\tilde{\mathbf{X}})$ to mitigate mode collapse in $\tilde{\mathbf{X}}$ and discover the causal mechanisms responsible for producing its outliers.

Proposition 4 *Minimizing the Maximum Mean Discrepancy (MMD) loss $\mathcal{L}_{MMD}(\mathbf{X}, \tilde{\mathbf{X}})$ aligns higher-order statistics of $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$, complementing adversarial loss to achieve overall distributional alignment.*

Proof The proof of Proposition 4 is available in Appendix A.4. □

Our ablation study in Appendix B indicates that the MMD term incorporated from DAG-GAN [38] makes contributions to causal discovery.

3.1.5 Post-nonlinear FCM

So far, we have discussed the loss terms for the LiNGAM and ANM cases, where $\tilde{\mathbf{X}}$ generated using causal mechanism approximations $\tilde{X} := f(X) + \mathcal{Z}$ or $\tilde{X}_j = f_j(Pa_j) + \mathcal{Z}_j$ is treated as the final output of the model to mimic the training data $\mathbf{X}$ via minimizing $||P(\mathbf{X}) - P_{G_A}(\tilde{\mathbf{X}})||$. One of the key advantages of DAGAF is its flexibility, allowing this to be extended to handle Post-Nonlinear Models (PNL).

PNL is crucial for causal discovery as it provides a more realistic approach to modeling causality by capturing nonlinear effects in observational data. Furthermore, PNL is considered a general superset that encompasses other identifiable models, such as ANM [39] and LiNGAM [21].

$$X_j := g_j(f_j(Pa_j) + \mathcal{Z}_j), \forall j, \mathcal{Z}_j \perp\!\!\!\perp f_j(Pa_j) \quad (6)$$

Without loss of generality, we rearrange (6) into

$$\mathcal{Z}_j = g_j^{-1}(X_j) - f_j(Pa_j), \quad (7)$$

where g^{-1} is the inverse of g. Under this setting (from the rearranged equation), the problem has been broken into two parts, which are to learn $f(\cdot)$ and $g^{-1}(\cdot)$ respectively.

Learning $f(\cdot)$ follows the same process as in the ANM and LiNGAM cases, as described so far in Sections 3.1.1 to 3.1.4. However, learning $g^{-1}(\cdot)$ is an additional step specific to the PNL case. In practice, both functions $g^{-1}(\cdot)$ and $f(\cdot)$ are modeled using two different neural networks, where $f(\cdot)$ is the same as before and $g^{-1}(\cdot)$ is the inverse of a general MLP. There is an additional Mean Squared Error (MSE) term involved in the training procedure, which we define as:

$$\mathcal{L}_{\text{PNL}}(\hat{\mathbf{X}}, \tilde{\mathbf{X}}) = MSE(\hat{\mathbf{X}}, \tilde{\mathbf{X}}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} ||g_j^{-1}(X_j)_i - f_j(Pa_j)_i||_2, \tag{8}$$

where $\hat{\mathbf{X}}$ is the output of g^{-1}.

It is worth noting that the reason why the loss terms in Sections 3.1.1-3.1.4 (where $f(.)$ is treated as the final output of the model) can be used by the PNL case is based on the idea of skip connections, as those used in ResNet. Although the output from $f(\cdot)$ in the PNL case is not the final output, we can still use it directly in these loss terms by essentially skipping the final function $g(\cdot)$, allowing the model to apply the same loss terms as in the ANM and LiNGAM cases. For more information on this concept, see [40].

3.1.6 Causal structure acyclicity

Minimizing the reconstruction and adversarial loss terms does not guarantee that G_A will be acyclic. To prevent cycles from occurring in the learned causal structures, we employ the implicit acyclicity constraint from [34] $h(A^{L_0}(f)) = 0$, where $A^{L_0} \in \mathbb{R}^{d\times d}$ is the weighted adjacency matrix described implicitly by the model weights. More details can be found in [34].

3.2 Simulating data generative processes

In the second stage of Algorithm 1, we focus on synthesizing realistic tabular data samples using the causal graph G_A produced from Step 1. Our data generation process assumes a different instance of the FCM $\mathcal{M}$ used in the causal discovery step, which we refer to as generator G here. Causal knowledge is transferred between FCM instances by loading W^{L_0} from $\mathcal{M}$ into $L_0 \in G$. To enable tabular data synthesis, we incorporate an additional noise vector $Z = \{Z_1, ...Z_d\} \sim \mathcal{N}(\mu, \sigma^2)$ into the architecture of the generator.

The models used in this step are trained adversarially to ensure that $P_{G_A}(\tilde{\mathbf{X}})$ closely approximates $P(\mathbf{X})$. Specifically, the network G creates samples while competing against a discriminator $D : \mathbb{R}^d \to \mathbb{R}$, whose aim is to distinguish between synthetic samples and observational samples. We apply Wasserstein-1 with gradient penalty to train our DGM, resulting in realistic samples indistinguishable from $\mathbf{X}$. The loss function is the same as (2). More specifically, we consider each connected layer $\alpha(L_j) \in \{\alpha(L_1), ...\alpha(L_d)\}$ as an individual generator $G_j(Z_j) \in \{G_1(Z_1), ...G_d(Z_d)\}$. This approach enables us to model each causal mechanism $f_j \in \{f_1, ...f_d\}$ such that $\tilde{X}_j$ is generated as either $\tilde{X} := G(X) + \mathcal{Z}$; $\tilde{X}_j := G_j(Pa_j) + Z_j$ or $\tilde{X}_j := g_j(G_j(Pa_j) + Z_j)$, depending on whether we assume LiNGAM, ANM or PNL. In other words, we generate a synthetic tabular dataset $\tilde{\mathbf{X}} \in \tilde{\chi} \subseteq \mathbb{R}^{n\times d} = \mathcal{F}(Z) = \{f_j(Pa_j, Z_j)\}$. During training, we only update the parameters $W = \{W^1, ..., W^L\}$ of the locally connected hidden layers, since modifying the weights of L_0 would affect the structural equations $\mathcal{F}$ used to produce $\tilde{\mathbf{X}}$.

Our experiments in Section 5.4 indicate that our DGM can produce high-quality data under both the ANM and PNL structural assumptions.

3.3 Model architecture and training specifications

Figure 2 presents the overall architecture of the DAGAF framework. Figure 2a illustrates the ANM and LiNGAM setting, where input data $\mathbf{X}$ is processed by function f to produce $\hat{\mathbf{X}}$. The optimization is guided by multiple loss terms: $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$, $\mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$, $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$, and $\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$, with $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ specifically excluded in the LiNGAM case. Figures 2b extends 2a by incorporating the PNL model. The right-hand branch follows the same structure as Fig. 2a, while the additional left-hand branch applies g^{-1} to invert $\mathbf{X}$. This inversion contributes to computing $\mathcal{L}_{\text{PNL}}(\hat{\mathbf{X}}, \tilde{\mathbf{X}})$, which is then integrated with the other loss terms from the right-hand branch, forming a unified optimization framework. Figure 2c depicts the data generation process used to synthesize artificial data, demonstrating how the framework facilitates structured data synthesis.

We incorporate the Multi-Layer Perceptron (MLP) from [34] as an FCM $\mathcal{M}$ to model f in the causal structure learning step. Its architecture consists of two components: 1) an initial linear layer L_0, which constitutes an implicit definition of G_A, enabling the modelling of causal structures and 2) a set of locally connected hidden layers $L = \{\alpha(L_1), ..., \alpha(L_d)\}$, with α being a nonlinear transformation applied to each layer, designed to approximate and learn $\mathcal{F} = \{f_1, ..., f_d\} \in G_A$. Meanwhile, g is a general MLP with five linear layers [d - $10d$ - $10d$ - $10d$ - d] (1 input, 3 hidden and 1 output) and nonlinearity applied using the ReLU activation function (only used in the PNL case). More specifically, each feature in $\mathbf{X}$ is modeled with a neural network of L hidden layers $f_j(Pa_j, \mathcal{Z}_j; W_j^1, ..., W_j^L)$, $j \in [1, d]$, where W_j^l

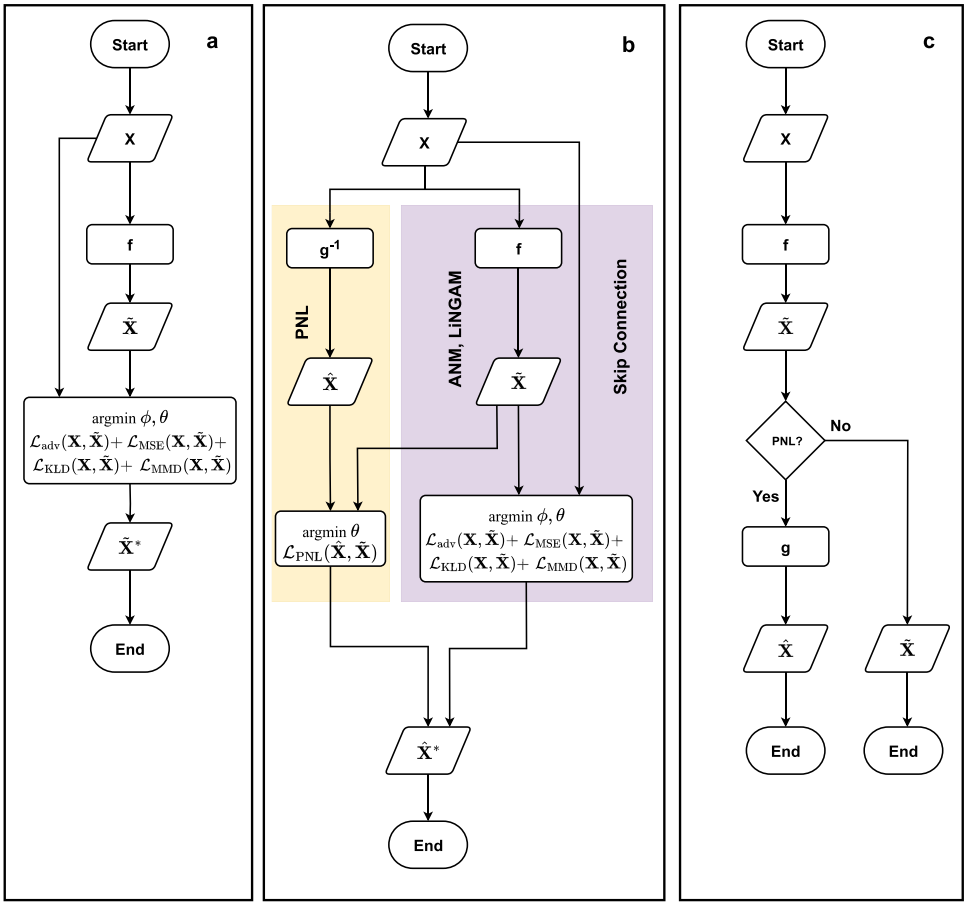


Fig. 2 A Visual Representation of DAGAF. (a) The optimization structure under ANM and LiNGAM, where input data is processed to reconstruct $\tilde{\mathbf{X}}$ using multiple loss terms, excluding $\mathcal{L}_{\text{KLD}}$ in the LiNGAM case. (b) The extended framework integrating ANM, LiNGAM, and PNL, where an additional inversion function g^{-1} is introduced to compute $\mathcal{L}_{\text{PNL}}$, unifying the optimization process. (c) The synthetic data generation process, illustrating how the framework enables structured data synthesis while preserving underlying causal relationships

denotes the parameters of the l^{th} layer. Let $W_j^{(0)} \in \mathbb{R}^{h \times d}$ be the weight matrix within L_0 connecting to the local neural network modeling X_j, where h is the latent size and d is the number of input variables. For each pair of variables X_j and X_k, the Ridge regression norm of the weights connecting X_k to all latent units in the network for X_j is computed as:

$$A_{jk} = \left\| W_{j,k,:}^{(1)} \right\|_2 = \sqrt{\sum_{m=1}^{h} \left(W_{j,k,m}^{(1)} \right)^2}, \tag{9}$$

where $W_{j,k,m}^{(1)}$ represents the weight connecting the k-th input variable X_k to the m-th latent unit in the first layer of the network for X_j.

Throughout the training process, a learning rate of 3×10^{-3} is employed, with a batch size set at 1000. Ridge regression regularization is applied in both steps by setting the weight decay of both discriminators to 1×10^{-6}. The models within our framework undergo iterative optimization, with their parameters updated through gradient descent.

The adversarial loss is applied to the reconstructed distribution $P_{G_A}(\tilde{\mathbf{X}})$, hence, in the causal structure learning step, a noise vector is not involved during training. Once the parameters in A^{L_0} have been updated, we convert A^{L_0} to G_A using the post-processing step $G_A = \sqrt{A^{L_0}(f)}$, $w_{jk}^2 \in A^{L_0}(f)$ followed by thresholding with value 0.3, considered best by existing works such as DAG-GNN [17], GAE [41] and many others. These final two steps are required to recover the weights $w_{jk} \in G_A$ from $A^{L_0}(f)$ and to reduce the number of false discoveries in G_A.

To learn g^{-1} for the PNL case, we need to invert the architecture and training procedure of g such that $\tilde{\mathbf{X}}$ is used as input to produce the original $\mathbf{X}$. We opt to focus on the training algorithm only as due to the generality of g inverting its architecture will not result in any changes to its configuration.

Remark 1 *The output data $\tilde{\mathbf{X}}$ from Step 1 is solely used to compute the loss terms during training and then it is discarded. This happens because the reconstruction loss used to learn the causal structure of $\mathbf{X}$ significantly reduces the range of the generated samples, resulting in $\tilde{\mathbf{X}}$ with high fidelity but low diversity.*

We treat the training as a constraint continuous optimization problem because of the requirement to adjust the parameters of the acyclicity constraint together with the weights of the model. Hence, we use the modified version of the augmented Lagrangian [42] employed in DAG-Notears-MLP to solve it.

3.4 Computational complexity analysis

The DAGAF framework comprises three distinct models: the FCM/Generator ($\mathcal{M}/G$), the Discriminator D (in the ANM and LiNGAM settings), and an additional MLP g for the PNL case. These models are trained using an algorithm that integrates three interconnected components: Causal Structure Learning, Tabular Data Synthesis, and Augmented Lagrangian-based Continuous Optimization. This complex architecture and training methodology make DAGAF significantly more intricate compared to other state-of-the-art methods, such as DAG-GNN [17], GraN-DAG [18], DECAF [30], and Causal-TGAN [31], which focus solely on causal discovery or tabular data synthesis and involve fewer models. This complexity motivated us to assess the efficiency and practicality of our approach.

We examine the resource requirements of DAGAF for performing causal structure learning and tabular data synthesis simultaneously. To achieve this, we provide pseudo-code for Algorithm 1 and analyze its time complexity. This alternative representation of the training process for our framework is presented in Appendix E. The space complexity of DAGAF is $\mathcal{O}(d)$, where d represents the number of variables in $\mathbf{X}$, aligning with the complexity of Notears and its extensions.

To perform a thorough time complexity analysis of our framework, we evaluate the efficiency of each stage in the pseudo-code from Appendix E separately. This analysis also incorporates the augmented Lagrangian and causal knowledge transfer components. The total computational complexity is determined by summing the individual complexities of each component in the pseudo-code for Algorithm 1 and identifying the most resource-intensive stage. We start with the initial phase of the framework, which involves declaring variables, hyperparameters, and model instances. These operations are treated as atomic and require constant time $\mathcal{O}(1)$.

Next, the training procedure is executed by directly applying the augmented Lagrangian, which involves three nested loops: 1) governed by k_max_iter, 2) constrained by the range of values for c, and 3) determined by the number of *epochs* in the training process. In the worst-case scenario, each loop runs to its maximum limit, and each has linear complexity. Assuming the range for each loop is constant, the time complexity of optimizing the augmented Lagrangian parameters depends solely on the number of data variables in the input dataset, resulting in a complexity of $\mathcal{O}(d)$ per each individual loop, where d represents the number of variables in the observational data. Considering the three nested loops and the parameter optimization step (which takes constant time, $\mathcal{O}(1)$), the overall computational complexity of the augmented Lagrangian is cubic, $\mathcal{O}(d^3)$.

Inside the augmented Lagrangian, the training algorithm is divided into two stages: causal structure learning and tabular data synthesis, with an additional step for transferring causal knowledge between the stages, which takes constant time $\mathcal{O}(1)$. Both stages utilize stochastic gradient descent (SGD) for optimizing model parameters. Generally, the computational complexity of SGD is $\mathcal{O}(knd)$, where k is the number of epochs, n is the number of samples, and d is the number of variables in $\mathbf{X}$. For DAGAF, both k and n are constant hyperparameters, meaning the optimization complexity depends solely on the number of data attributes in the input. Therefore, the total computational complexity for both stages is linear, $\mathcal{O}(d)$.

The overall time complexity of Algorithm 1 is given by $\mathcal{O}(d)^3 + 2\mathcal{O}(d)$, which simplifies to $\mathcal{O}(d)^3$ as we focus on the fastest-growing term. This analysis shows that DAGAF has a cubic computational complexity, aligning with results reported for similar algorithms in previous studies [16, 18].

4 Causal identifiability

Our theoretical analysis demonstrates that the DAG model is unique and hence identifiable under the assumptions of the DAGAF framework, which include ANM, LiNGAM, and PNL. This analysis is conducted under the assumption that the data is continuous and follows i.i.d. conditions.

Proposition 5 *Under the Additive Noise Model (ANM), Linear non-Gaussian Acyclic Model (LiNGAM) or Post-Nonlinear Model (PNL) assumption, there exists a unique DAG $\mathcal{G}_{\mathcal{A}}$ capable of defining the observed joint distribution $P(\mathbf{X})$.*

Proof The proof of Proposition 5 is available in Appendix A.5. □

Proposition 5 establishes that for a joint distribution $P(\mathbf{X})$ over random variables $\{X_1, ..., X_d\}$ generated by a true causal graph $\mathcal{G}_{\mathcal{A}}$, there exists an identifiable causal graph G_A such that $G_A = \mathcal{G}_{\mathcal{A}}$, provided that the causal model follows the ANM, LiNGAM, or PNL assumptions.
In addition, we analyze how the loss terms used to train DAGAF behave under challenging conditions, including non-i.i.d. data, missing values, and discrete variables.

4.1 Impact of non-i.i.d. conditions

Now we consider some real-world data case, where the samples $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$ are no longer independent (i.e. $\mathbf{X}_i \not\perp\!\!\!\perp \mathbf{X}_j$) and each data point $\mathbf{X}_i$ is drawn from heterogeneous distributions $P_i(\mathbf{X})$. In such settings, the empirical distribution $P'(\mathbf{X})$ becomes a biased estimate of the true distribution $P(\mathbf{X})$, impacting the optimization.

We assume that the true and the implicitly generated distributions are defined as $P'(\mathbf{X}) = P(\mathbf{X}) + \delta(\mathbf{X})$ and

$P'_{G_A}(\tilde{\mathbf{X}}) = P_{G_A}(\tilde{\mathbf{X}}) + \delta(\tilde{\mathbf{X}})$, where $\delta(\mathbf{X})$ and $\delta(\tilde{\mathbf{X}})$ capture deviations from the i.i.d. assumptions.

4.1.1 Adversarial loss and identifiability

Under non i.i.d. condition, $\mathcal{L}'_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}}) = D(P'(\mathbf{X})||P_{G_A}(\tilde{\mathbf{X}}))$. The bias $\delta(\mathbf{X})$ affects the gradients of $\mathcal{L}'_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$:

$$\nabla_\phi \mathcal{L}'_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}}) = \nabla_\phi D(P(\mathbf{X})||P_{G_A}(\tilde{\mathbf{X}})) + \nabla_\phi D(\delta(\mathbf{X})||P_{G_A}(\tilde{\mathbf{X}})).$$

The additional term $\nabla_\phi D(\delta(\mathbf{X})||P_{G_A}(\tilde{\mathbf{X}}))$ can destabilize optimization by adding spurious gradient components due to dependencies or heterogeneity, and by amplifying sensitivity to noise in the data.

4.1.2 MSE loss and identifiability

Under the non-i.i.d. conditions:

$$\mathcal{L}'_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) = \mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta(\mathbf{X}).$$

If $\delta(\mathbf{X})$ introduces correlations between samples $\mathbf{X}_i$ and $\mathbf{X}_j$, this violates the independence of the noise terms $\mathcal{Z}_j$. As a result, the non-i.i.d. MSE loss term $\mathcal{L}'_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$ may incorrectly fit spurious patterns across samples. In turn, the output of $f_j(Pa_j)$ may no longer capture the true functional relationship.

Furthermore, the gradient of $\mathcal{L}'_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$ with respect to θ is:

$$\nabla_\theta \mathcal{L}'_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) = \nabla_\theta \mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) + \nabla_\theta \delta(\mathbf{X}).$$

The additional term $\nabla_\theta \delta(\mathbf{X})$ introduces instability due to spurious gradients from dependencies across samples, and heterogeneity-induced noise in gradients. This instability makes optimization sensitive to the choice of initialization and hyperparameters, thus reducing convergence reliability.

4.1.3 Kullback-Leibler divergence loss and identifiability

The empirical estimate of the KLD under non-i.i.d. conditions becomes:

$$\mathcal{L}'_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n}\sum_{i=1}^{n} \log \frac{P_{G_A}(\tilde{\mathbf{X}}_i)}{P'(\mathbf{X}_i)}.$$

Expanding $\mathcal{L}'_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ and applying a first-order Taylor expansion $P(\mathbf{X}_i)$, we have

$$\mathcal{L}'_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) \approx \mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) - \frac{1}{n}\sum_{i=1}^{n} \frac{\delta(\mathbf{X}_i)}{P(\mathbf{X}_i)}.$$

The term $\frac{\delta(\mathbf{X}_i)}{P(\mathbf{X}_i)}$ introduces bias, particularly when $\delta(\mathbf{X}_i)$ varies significantly across samples. This bias skews the optimization of $P_{G_A}(\tilde{\mathbf{X}})$, which potentially leads to an approximate distribution $P_{G_A}(\tilde{\mathbf{X}})$ that deviates from $P(\mathbf{X})$.

The gradient of the KLD loss under non-i.i.d. conditions is defined as:

$$\nabla_\theta \mathcal{L}'_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) \approx \nabla_\theta \mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) - \int \nabla_\theta P_{G_A}(\tilde{\mathbf{X}}) \frac{\delta(\mathbf{X})}{P(\mathbf{X})} d\mathbf{X} d\tilde{\mathbf{X}}.$$

The additional term $\int \nabla_\theta P_{G_A}(\tilde{\mathbf{X}}) \frac{\delta(\mathbf{X})}{P(\mathbf{X})} d\mathbf{X} d\tilde{\mathbf{X}}$ adds noise to the gradients, reducing the stability of optimization. This may introduce spurious directions in the parameter space, which make convergence to the true distribution $P(\mathbf{X})$ more challenging.

4.1.4 MMD loss and identifiability

Expanding all instances of $k(.)$, we have:

$$k(\mathbf{X}_i, \mathbf{X}_j) = k(P(\mathbf{X}_i), P(\mathbf{X}_j)) + \Delta_{P(\mathbf{X})}(\mathbf{X}_i, \mathbf{X}_j),$$
$$k(\mathbf{X}_i, \tilde{\mathbf{X}}_j) = k(P(\mathbf{X}_i), P_{G_A}(\tilde{\mathbf{X}}_j)) + \Delta_{P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})}(\mathbf{X}_i, \tilde{\mathbf{X}}_j),$$
$$k(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j) = k(P_{G_A}(\tilde{\mathbf{X}}_i), P_{G_A}(\tilde{\mathbf{X}}_j) + \Delta_{P_{G_A}(\tilde{\mathbf{X}})}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j),$$

where $\Delta_{P(\mathbf{X})}(\mathbf{X}_i, \mathbf{X}_j)$, $\Delta_{P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})}(\mathbf{X}_i, \tilde{\mathbf{X}}_j)$ and $\Delta_{P_{G_A}(\tilde{\mathbf{X}})}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)$ represent perturbations due to non-i.i.d. effects. The empirical MMD becomes:

$$\mathcal{L}'_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) \approx \mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) + \Delta,$$

where the non-i.i.d. effect Δ is defined as follows:

$$\Delta = \frac{1}{n}\sum_{i\neq j}^{n} \Delta_{P(\mathbf{X})}(\mathbf{X}_i, \mathbf{X}_j) - \frac{2}{n}\sum_{i\neq j}^{n} \Delta_{P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})}(\mathbf{X}_i, \tilde{\mathbf{X}}_j) + \frac{1}{n}\sum_{i\neq j}^{n} \Delta_{P_{G_A}(\tilde{\mathbf{X}})}(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)$$

The term Δ introduces bias into the empirical MMD estimate, which may no longer converge to the true population MMD even as $n \to \infty$.

The gradient of $\mathcal{L}'_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$ with respect to model parameters θ is:

$$\nabla_\theta \mathcal{L}'_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) = 2\left(\mathbb{E}_{\mathbf{X}, \mathbf{X}' \sim P'(\mathbf{X})}[\nabla_\theta k(\mathbf{X}, \mathbf{X}')] - \mathbb{E}_{\mathbf{X} \sim P'(\mathbf{X}), \tilde{\mathbf{X}} \sim P'_{G_A}(\tilde{\mathbf{X}})}[\nabla_\theta k(\mathbf{X}, \tilde{\mathbf{X}})]\right).$$

The additional perturbations $\Delta_{P(\mathbf{X})}$, $\Delta_{P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})}$ and $\Delta_{P_{G_A}(\tilde{\mathbf{X}})}$ introduce noise into the gradients, potentially destabilizing optimization and making convergence difficult.

Table 1 DAG structures recovered from linear data

Model	SHD (5000 linear samples)			
	d=10	d=20	d=50	d=100
DAG-Notears	8.6 ± 7.2	13.8 ± 9.6	41.8 ± 29.4	102.8 ± 53.2
DAG-Notears-MLP	4.6 ± 4.3	7.6 ± 6.3	29.6 ± 18.5	74 ± 30.6
DAG-GNN	6 ± 6.9	11.4 ± 8.2	33.6 ± 21.2	85.4 ± 46.4
GAE	5.5 ± 4.9	10.3 ± 7.2	31.3 ± 13.8	80.2 ± 24.6
GraN-DAG	3.4 ± 5.2	6.4 ± 7.5	25.2 ± 14.6	68.4 ± 25.8
CAREFL	2.7 ± 4.8	5.9 ± 7.1	24.9 ± 14.1	66.9 ± 24.7
DAG-NF	2.4 ± 4.6	5.2 ± 6.9	23.1 ± 13.4	64.2 ± 24.3
VI-DP-DAG	2.1 ± 4.5	4.5 ± 6.7	22.4 ± 12.7	63.7 ± 23.5
DCRL	1.8 ± 2.7	3.1 ± 4.8	18.7 ± 11.9	53.3 ± 21.9
DAG-WGAN	5.2 ± 3.8	9.2 ± 5.7	19.6 ± 12.3	58.6 ± 22.7
DAG-WGAN+	3.7 ± 3.1	5.6 ± 4.9	17.2 ± 10.5	49.1 ± 20.1
DAGAF	1.4 ± 2.3	2 ± 4.4	16.4 ± 9.8	38.8 ± 18.3

4.2 DAG identifiability in discrete variables

Different DAGs can give rise to the *same* joint distribution in the discrete setting, thereby leading to non-uniqueness in identifying the true DAG $\mathcal{G}_{\mathcal{A}}$. For simplicity, consider two DAGs $\mathcal{G}_{1_{\mathcal{A}_1}}$ and $\mathcal{G}_{2_{\mathcal{A}_2}}$ that are structurally different but induce the same joint distribution. In a *discrete setting*, the symmetry between causal relations often implies that reversing edges or reparameterizing certain relationships leads to the same joint distribution. More formally:

$$P(X_i \mid \text{Pa}(X_i)) = P(X_j \mid \text{Pa}(X_j)) \quad \text{for some } (X_i, X_j) \text{ such that } X_j \in \text{Pa}(X_i) \text{ or } X_i \in \text{Pa}(X_j).$$

This symmetry implies that the conditional distributions from both DAG are equal. Thus, the *identifiability of the DAG* is lost in the discrete setting due to the *equivalence* of the conditional distributions, even though the underlying structural graph may differ.

4.3 Impact of missing data

Missing data in real-world datasets can arise from different mechanisms. If data is Missing Completely at Random, the missingness is unrelated to any variables, reducing sample size but preserving identifiability with sufficient data. Missing at Random occurs when missingness depends only on observed variables, potentially introducing bias in independence tests but still allowing DAG discovery with robust imputation. Missing Not at Random is the most problematic, as missingness depends on unobserved factors, making the dataset unrepresentative of the true causal structure.

As the identifiability of the true DAG $\mathcal{G}_{\mathcal{A}}$ relies heavily on correctly testing conditional independence relationships (e.g., $\mathcal{Z}_j \perp\!\!\!\perp Pa_j$ in the PNL model), missing data reduces the statistical power of these tests. Missing large portions of data may lead to unreliable or incorrect conditional independence tests. Spurious dependencies or independencies may arise due to imputation strategies or biased sampling. The ANM,

Table 2 DAG structures recovered from non-linear-1 data

Model	SHD (5000 non-linear-1 samples)			
	d=10	d=20	d=50	d=100
DAG-Notears	11.4 ± 4.5	28.2 ± 10.2	55 ± 23.1	105.6 ± 48.3
DAG-Notears-MLP	5.2 ± 1.8	15.4 ± 4.6	43.8 ± 15.4	86.2 ± 29.8
DAG-GNN	9.2 ± 3.3	23.4 ± 8.4	50.2 ± 19.5	98.6 ± 37.6
GAE	8.6 ± 2.2	20 ± 5.7	47.5 ± 10.2	92.3 ± 18.9
GraN-DAG	4 ± 2.4	11.2 ± 6.5	36.4 ± 11.9	72.8 ± 21.7
CAREFL	3.8 ± 2.2	10.9 ± 6.2	34.1 ± 11.2	71.7 ± 19.1
DAG-NF	3.4 ± 2.1	10.4 ± 5.6	31.6 ± 10.7	69.5 ± 17.3
VI-DP-DAG	3.1 ± 2	9.8 ± 5.1	28.7 ± 9.3	68.1 ± 16.5
DCRL	2.9 ± 1.7	7.5 ± 4	24.3 ± 7.8	61.4 ± 14.9
DAG-WGAN	6.4 ± 1.4	18.6 ± 3.7	22 ± 8.6	64.6 ± 15.2
DAG-WGAN+	4.9 ± 1.2	14.2 ± 3.3	20.5 ± 6.9	57.1 ± 14.5
DAGAF	2.6 ± 1	5.2 ± 2.8	18.8 ± 6.2	50.2 ± 13.4

Table 3 DAG structures recovered from non-linear-2 data

Model	SHD (5000 non-linear-2 samples)			
	d=10	d=20	d=50	d=100
DAG-Notears	10.4 ± 3.9	22.4 ± 8.1	47.6 ± 21.2	112.8 ± 57.8
DAG-Notears-MLP	5.4 ± 1.5	13.8 ± 4.3	30.4 ± 15.7	85.6 ± 35.6
DAG-GNN	8.4 ± 3.2	19.2 ± 7.7	36.2 ± 18.6	91.8 ± 49.3
GAE	7.3 ± 1.8	17.4 ± 5.1	33.7 ± 13.7	88.4 ± 26.6
GraN-DAG	4.2 ± 2.1	11.6 ± 5.6	25.2 ± 14.5	71.6 ± 29.7
CAREFL	3.8 ± 1.8	10.5 ± 5.3	24.8 ± 13.8	69.9 ± 26.1
DAG-NF	3.3 ± 1.7	9.7 ± 4.9	24.3 ± 13.1	68.1 ± 24.3
VI-DP-DAG	2.8 ± 1.6	9.3 ± 4.7	23.8 ± 13.3	67.3 ± 23.8
DCRL	2.2 ± 1.3	7.1 ± 2.9	15.1 ± 9.4	59.5 ± 17.2
DAG-WGAN	6.6 ± 1.2	15.2 ± 3.4	22.6 ± 12.9	64.2 ± 21.5
DAG-WGAN+	5.1 ± 1.1	12.3 ± 2.5	17.5 ± 10.2	56.7 ± 18.4
DAGAF	1.4 ± 0.9	5.8 ± 2.2	14.2 ± 8.3	51.8 ± 16.2

LiNGAM and PNL model assume that the noise term $\mathcal{Z}_j$ is independent of its parents ($\mathcal{Z}_j \perp\!\!\!\perp Pa_j$). Missing data can obscure or distort observed relationships, making it difficult to separate noise from modeled contributions.

In addition, the functional forms f_j (nonlinear for ANM, linear for LiNGAM) and g_j (nonlinear for PNL) are assumed to be known or learnable. However, the data incompleteness characteristic often associated with real-world data violates this assumption. In the LiNGAM case, non-Gaussian noise becomes harder to test.

Identifiability relies on correctly estimating marginal distributions. Missing data distorts these estimates, especially when parent variables or structural nodes are disproportionately missing.

5 Experimental results

We conduct a range of experiments on the proposed general framework for causal structure learning using various datasets that include continuous and discrete data types to assess the following aspects:

- Structure learning accuracy, which assesses the effectiveness of modeling the relationships between features in observational data.
- Synthetic data quality, which investigates the quality of the data produced from the learned generative process.
- Ablation study and sensitivity analysis to assess the configuration of the loss terms and the hyper-parameter settings for the training. - for more information, the reader is referred to Appendices B and C.

In this section, we outline the configurations for the causal discovery and data quality experiments, and present the results along with the metrics employed for their evaluation.

For evaluating structure learning, our model is compared with leading DAG-learning methods, including DAG-WGAN [19], DAG-WGAN+ [43], DAG-Notears-MLP [34], Dag-Notears [16], DAG-GNN [17], GraN-DAG [18], GAE [41], CAREFL [44], DAG-NF [45], DCRL [46] and VI-DP-DAG [47]. The metric used throughout all experiments to measure the quality of the discovered causality is the Structural Hamming Distance (SHD) [48]. We selected SHD because it integrates several individual metrics, including True Positive Rate (TPR), False Discovery Rate (FDR), and False Positive Rate (FPR). It is important to acknowledge that the set of metrics SHD $=$ {TPR, FDR, FPR} used in this study is not the only approach to evaluating the accuracy of the learned structures. Other metrics, such as Area Under Curve (AUC) and Area Over Curve (AOC), can also be employed.

Table 4 DAG structures recovered from post-non-linear-1 data

Model	SHD (5000 post-non-linear-1 samples)			
	d=10	d=20	d=50	d=100
DAG-GNN	13.7 ± 9.2	21.7 ± 10.4	63.7 ± 31.2	118.6 ± 50.1
GAE	12.3 ± 8.1	19.1 ± 8.8	56.2 ± 24.6	101.3 ± 37.4
CAREFL	11.8 ± 6.4	18.5 ± 7.9	52.1 ± 22.8	97.2 ± 34.9
DAG-NF	11.2 ± 5.3	16.2 ± 6.1	47.3 ± 19.5	92.5 ± 31.3
DAG-WGAN	10.5 ± 4.7	15.6 ± 5.8	44.5 ± 17.7	88.7 ± 29.6
DAG-WGAN+	8.4 ± 3.3	12.8 ± 4.3	32.8 ± 13.6	66.1 ± 21.2
DAGAF	5.6 ± 2.5	7.3 ± 3.2	25.4 ± 11.3	52.4 ± 15.7

Table 5 DAG structures recovered from post-non-linear-2 data

Model	SHD (5000 post-non-linear-2 samples)			
	d=10	d=20	d=50	d=100
DAG-GNN	10.8 ± 8.7	16.1 ± 11.9	37.1 ± 30.3	128.3 ± 48.2
GAE	9.1 ± 6.3	14.3 ± 9.5	31.5 ± 24.8	105.7 ± 34.4
CAREFL	8.3 ± 5.8	13.5 ± 8.3	29.8 ± 22.4	92.1 ± 32.3
DAG-NF	7.7 ± 5.5	12.8 ± 7.4	28.4 ± 21.7	84.8 ± 28.5
DAG-WGAN	7.2 ± 5.2	11.4 ± 6.2	25.2 ± 18.6	76.5 ± 27.6
DAG-WGAN+	4.5 ± 3.6	8.6 ± 5.1	21.7 ± 12.3	69.4 ± 19.1
DAGAF	2.9 ± 2.4	5.7 ± 3.6	18.6 ± 10.5	47.2 ± 14.7

We also analyze the quality of the synthetic data produced by DAGAF. In particular, we conduct various tests to examine the statistical properties of $\tilde{\mathbf{X}}$. We evaluate the similarity between $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$ using boxplot analysis and marginal distributions. Additionally, we calculate the correlation matrices for both χ and $\tilde{\chi}$ to explore the interdependencies among their covariates.

5.1 Continuous data

We conduct tests on continuous data types using simulated data produced from predefined structural equations and Directed Acyclic Graph (DAG) structures. Specifically, we construct an Erdos-Renyi [49] causal graph with an expected node degree of 3, which serves as the ground-truth DAG $\mathcal{G}_{\mathcal{A}}$ and can be represented by a weighted adjacency matrix A. Afterwards, we generate 5000 observational data samples for each test by utilizing different equations (namely linear: $\tilde{X} := A^T X + \mathcal{Z}$, non-linear-1: $\tilde{X} := Acos(X + 1) + \mathcal{Z}$, non-linear-2: $\tilde{X} := 2sin(A(X + 0.5)) + A(X + 0.5) + \mathcal{Z}$, post-non-linear-1: $\tilde{X} := sinh(Acos(X + 1) + \mathcal{Z})$, and post-non-linear-2: $\tilde{X} := tanh(2sin(A(X + 0.5)) + A(X + 0.5) + \mathcal{Z})$). These structural equations have been widely used in numerous papers in DAG learning, including the DAG-GNN model [17], Gran-DAG [18], GAE [41], DAG-WGAN [19], DAG-WGAN+ [43] and Notears-MLP [34] - to name but a few. The application of these popular equations allow us to perform a comprehensive and robust comparison against other leading models in the field. The final two equations are modifications of the second and third ones designed to provide suitable test cases for experiments involving the PNL assumption. Ensuring the acyclicity of $\mathcal{G}_{\mathcal{A}}$ and satisfying the causal model assumptions outlined in Section 2, with the given above equations, enables us to generate i.i.d. samples that are appropriate for causal structure learning under the faithfulness condition.

Remark 2 *Although the list of equations provided in this section serves as a good collection of test cases for the continuous data experiments, it is not exhaustive. Other equations can be used as well.*

Our work follows the same methodology used in most other state-of-the-art DAG learning models, such as DAG-GNN, GraN-DAG, DAG-Notears and GAE among others, where the process of splitting data into training and validation sets is not as commonly applied as in traditional machine learning. Train-test splitting or cross-validation is typically used in predictive modeling tasks, but causal structure identification is focused on structural constraints and conditional independencies rather than predictive accuracy. Since causal relationships are structural, they are generally assumed to hold throughout the dataset, and therefore, partitioning the data may not provide significant additional benefit in terms of discovering the structure.

To evaluate the scalability of the model, we perform experiments with datasets that have 10, 20, 50, and 100 columns. To account for sample randomness and ensure fairness, each experiment is repeated five times, and the average Structural Hamming Distance (SHD) is reported. The results are shown in Tables 1, 2, 3, 4 and 5.

The results presented in Tables 1, 2, 3, 4 and 5 demonstrate that our proposed general framework for causal discovery consistently outperforms state-of-the-art DAG-learning methods across all tested scenarios-linear, non-linear-1, non-linear-2, post-nonlinear-1, and post-nonlinear-2-regardless of whether the underlying data-generating process follows

Table 6 DAG structures recovered from benchmark data

Datasets	Nodes	SHD			
		DAG-WGAN	DAG-WGAN+	DAG-GNN	DAGAF
Child	20	20	19	30	17
Alarm	37	36	35	55	43
Hailfinder	56	73	66	71	63
Pathfinder	109	196	194	218	181

Table 7 DAG structures recovered from real data

Model	Sachs Dataset SHD
DAG-WGAN	17
DAG-WGAN+	15
DAG-NF	15
DAG-GNN	25
GAE	20
GraN-DAG	17
VI-DP-DAG	16
DAGAF	ANM 9 / PNL 8

LiNGAM, ANM, or PNL assumptions. Notably, the gap in SHD between our model and the others grows further in our favor with the increase in data dimensionality. This observation highlights the enhanced performance of our approach for DAG-learning in datasets with a large number of variables.

It is also worth mentioning that, according to our results, DAGAF surpasses both traditional models in the field, including Notears, GAE, DAG-GNN, and GraN-DAG, as well as more recent approaches like DAG-WGAN(+), CAREFL, DAG-NF, DCRL and VI-DP-DAG, demonstrating the superiority of our model.

5.2 Benchmark experiments

In our experiments, we also included discrete datasets as part of an empirical study to demonstrate how our framework performs on such data. However, from our theoretical analysis presented in Section 4, we recognize that identifiability issues arise when applying our method to discrete datasets.

Specifically, we obtained the Child, Alarm, Hailfinder, and Pathfinder benchmark datasets, with their ground truths, from the Bayesian Network Repository https://www.bnlearn.com/bnrepository. These datasets are specifically organized

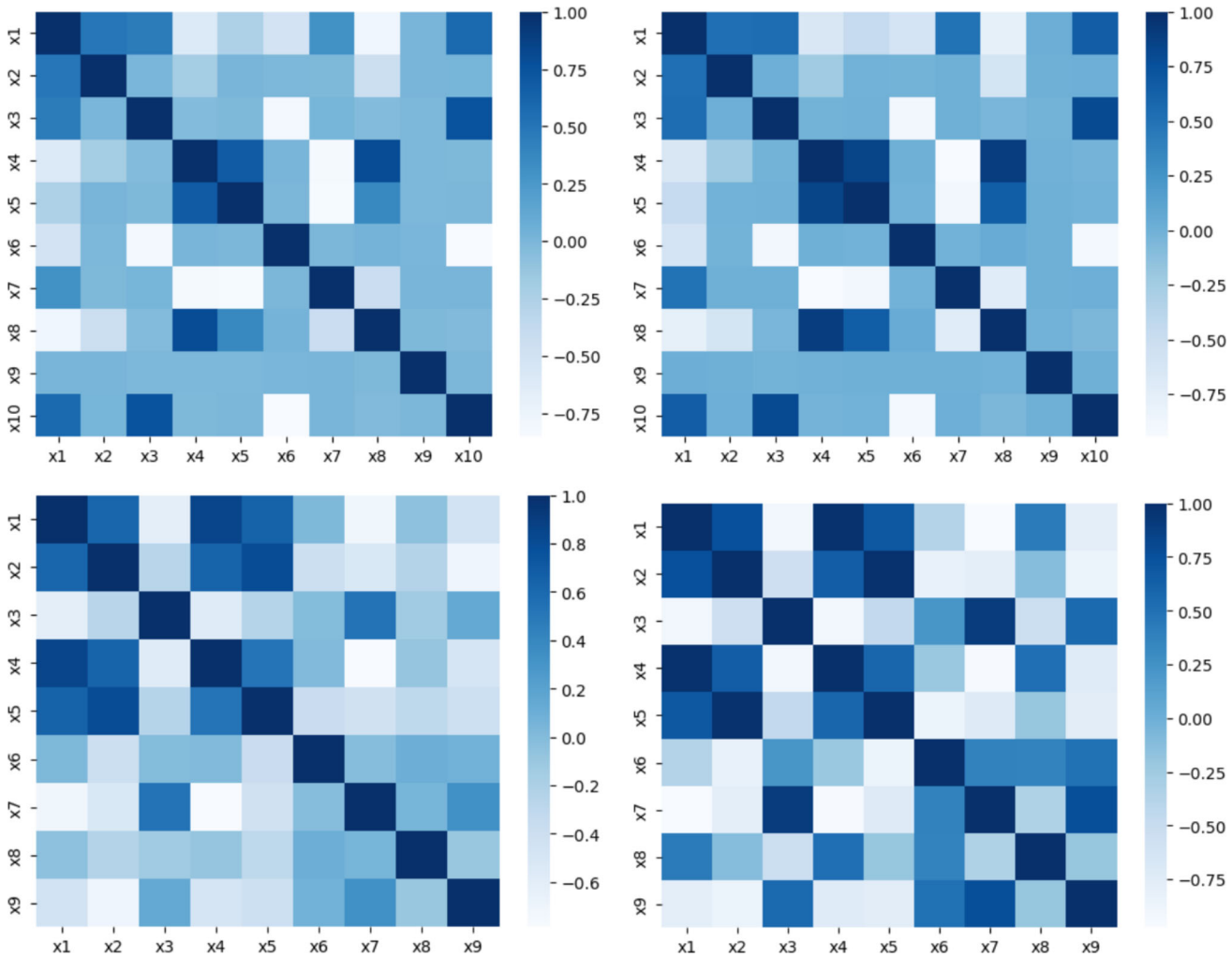


Fig. 3 Comparison of the correlation matrices for real (left) and synthetic (right) features reveals that the statistical correlations across the feature space for both real and synthetic data are nearly identical, in both the ANM (first row) and the PNL (second row) case

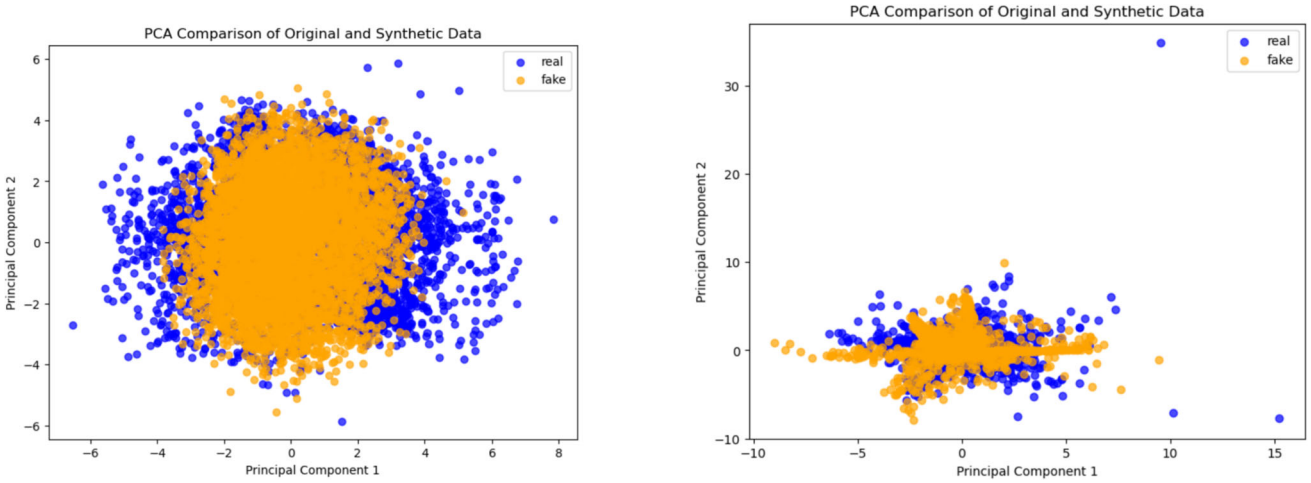


Fig. 4 Principal Component Analysis (PCA) between the original and synthetic samples for both the ANM (left) and the PNL (right) case. We observe both the input and the synthetic samples have similar clusters and outliers. The results indicate that the implicitly generated distribution resembles the original distribution in both mean and standard deviation, making them indistinguishable from each other

to facilitate scalability testing and enable a fair comparison with state-of-the-art methods. We evaluated our model against DAG-GNN and both versions of DAG-WGAN, with the results presented in Table 6.

According to the benchmark experiment results shown in Table 6, our method significantly outperforms DAG-GNN across all four datasets (Child, Alarm, Hilfinder, and Pathfinder). Additionally, both DAG-WGAN and its

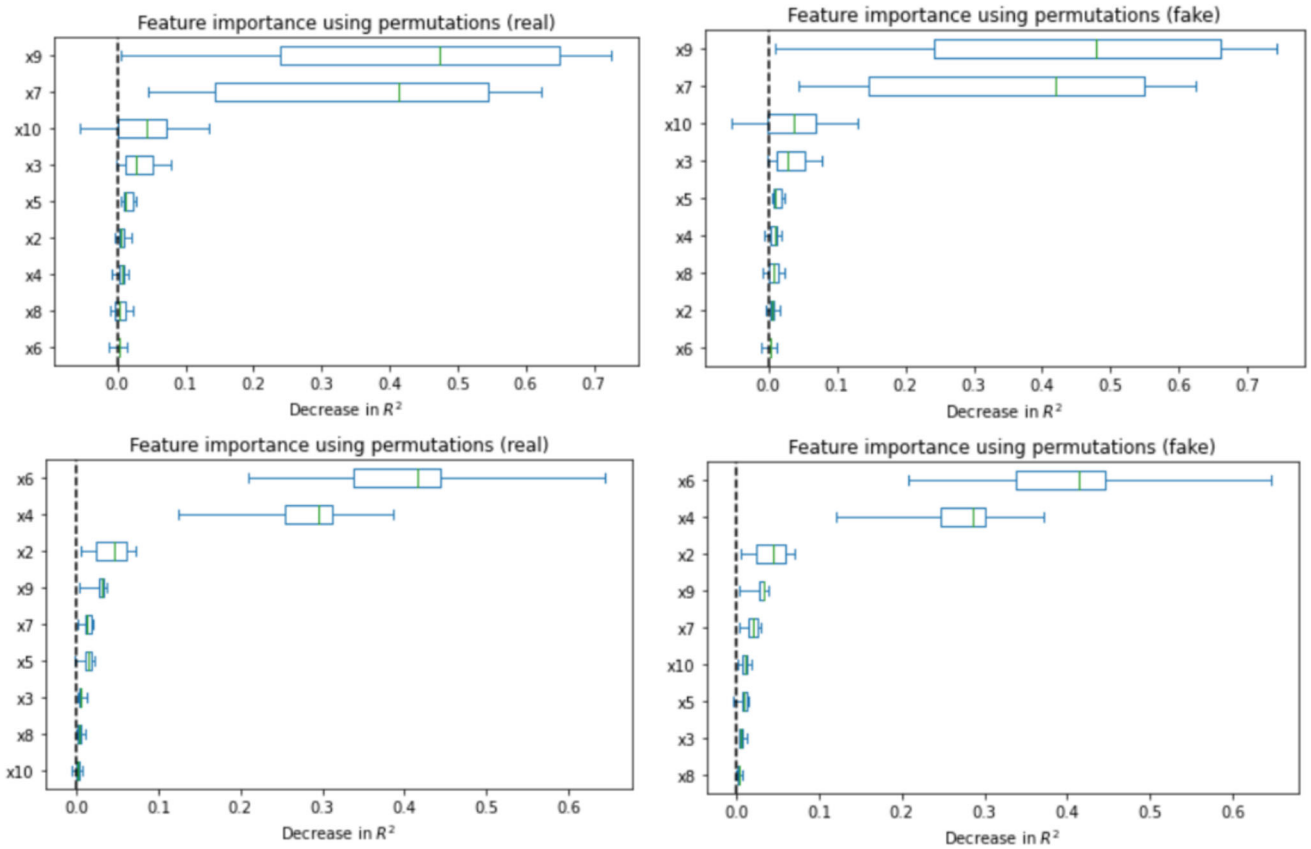


Fig. 5 Feature importance comparison between real (left) and synthetic (right) data, in both the ANM (first row) and the PNL (second row) case. The synthetic features with their relevance are indistinguishable from the original ones, allowing for their application in regression tasks

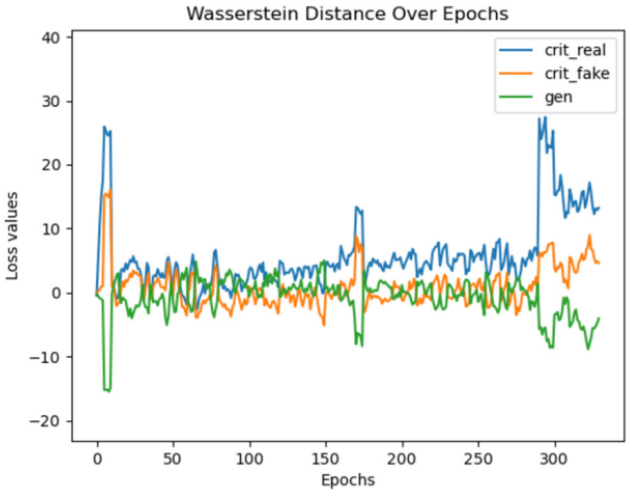


Fig. 6 Visualizing the Wasserstein distance between the original and synthetic data over the course of the augmented Lagrangian algorithm. The significant discrepancy between the real and the generated samples (165-170 and from 300 epochs onward) occurs because of fluctuations in the SHD, courtesy of the parameter-tuning for the continuous optimization approach. Conversely, the lowest SHD is detected when the Wasserstein Distance is at its lower conversions (50-150 and 175 - 275 epochs)

improved version, DAG-WGAN+, deliver inferior results compared to our framework on three out of the four datasets. Similar outcomes are observed in experiments with continuous datasets, where the SHD gap between our method and the others widens as the number of data variables increases.

5.3 Real data experiments

While our experiments with simulated data show the ability of DAGAF to generate decent results, they are not entirely conclusive, as simulations differ from real-world scenarios. To address this issue, we conducted experiments using a well-known real-world dataset called Sachs [50], which is widely recognized in the research community. This dataset comprises 7466 samples across 11 columns, with an estimated ground truth containing 20 edges. Additionally, our approach assumed both ANM and PNL during this test and compared the SHD produced by these FCM to determine whether the post-nonlinear model is superior when applied to real-world data. The results are presented in Table 7.

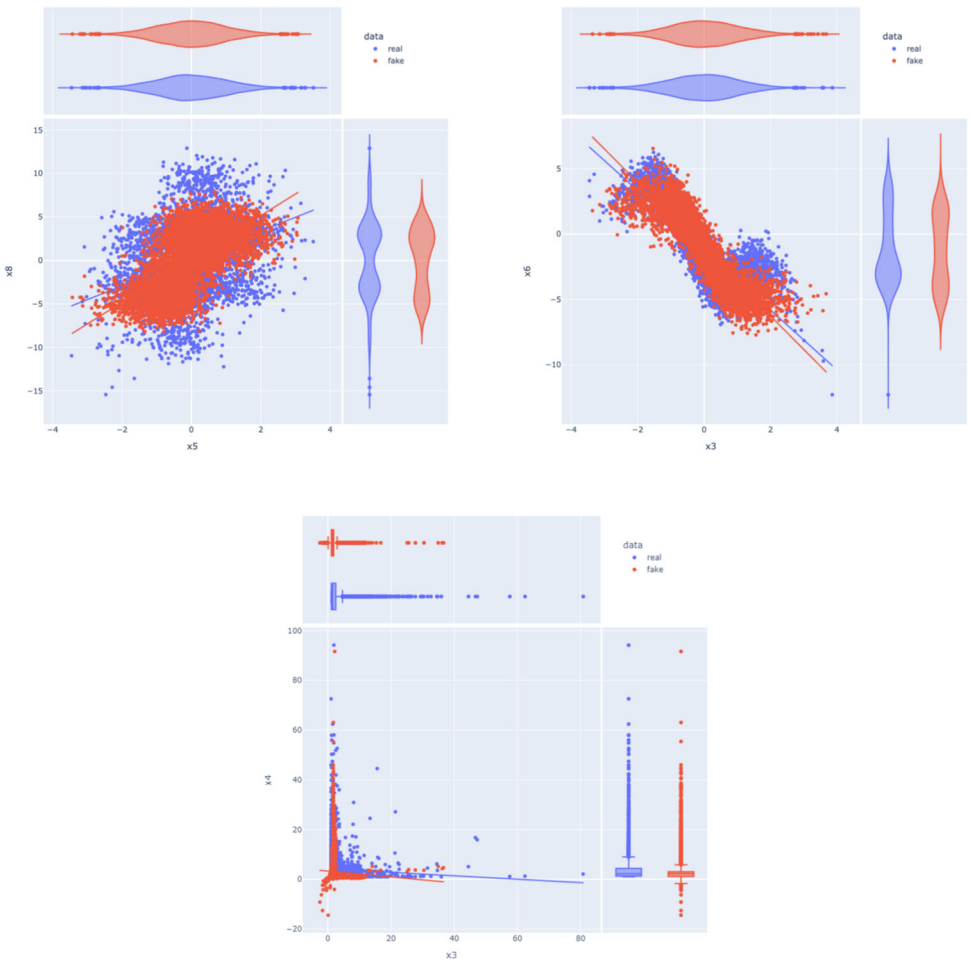


Fig. 7 Visualizing the distributions of the real and synthetic features, we plotted x5 against x8 (left), x3 against x6 (right), in the case of ANM, and x3 against x4 for the PNL case. The joint and marginal distributions are accurately modeled with no significant differences between the real and synthetic features

The experiment with the Sachs dataset shows that our method can also accurately discover DAG structures from real data. As indicated in Table 7, our framework significantly outperforms all other state-of-the-art algorithms involved in the study. Additionally, the empirical evidence suggests that the PNL assumption enables our approach to learn a more precise causal structure approximation compared to the application of other identifiable causal models.

5.4 Synthetic data quality

In this work, we have advocated for the superiority of our method over current state-of-the-art models by combining causality learning with synthetic data generation. To further support this claim, we compare the features (d=10) from two tabular datasets of simulation data (one based on the ANM and the other on the PNL assumption) with the features generated by our approach. We consider the special case where our model achieves an SHD of 0 on the simulation data, as this would result in the highest quality samples due to the complete knowledge of causal mechanisms in the generative process.

Our finding demonstrate that the synthetic samples generated by the proposed framework accurately replicate the correlations (Fig. 3) and capture the underlying patterns and structure of the original data (Fig. 4). Furthermore, the generated data contains enough predictive information to support regression tasks (Fig. 5). Minimizing the Wasserstein distance between observations and simulations (Fig. 6) implies that both the means and standard deviations of their respective probability distributions are matched, yielding an overlap of the joint and marginal distributions of their features (Fig. 7). We present only a few examples of each analysis in this section; additional results can be found in Appendix D.

6 Conclusion & future work

This research introduces a novel framework for multivariate causal structure learning aimed at holistically discovering DAG structures in a dataset to model its generative mechanisms and produce synthetic samples that closely resemble real data. We conducted a theoretical analysis demonstrating that the Wasserstein-1 distance metric can be leveraged for structure learning and explained how the integration of regularization and reconstruction loss terms in our training process can enhance the identification of causal relationships from observational data. Furthermore, we showcased the performance of our approach through extensive experiments, where the method significantly outperformed state-of-the-art DAG-learning techniques. The experimental results demonstrate that our method effectively handles numerical and categorical data types to accurately recover DAG structures under LiNGAM, ANM or PNL assumptions, while generating realistic data samples. The analysis of our results suggests that the Wasserstein distance plays a significant role in enhancing DAG learning. Our findings also indicate a close relationship between the simultaneous generation of diverse high-quality data and the learning of accurate DAG structures, suggesting that the synthesis of realistic data samples is facilitated by the recovery of meaningful variable relationships.

All results are generated using LiNGAM, ANM or PNL, which are proven to be identifiable [21–23, 52]. However, our experiments have been restricted to these models, which is a limitation. In future work, we plan to explore other identifiable structures, such as generalized linear models, polynomial regression and index models. Furthermore, our tabular data synthesis experiments have also been quite limited, focusing only on analyzing primitive features of datasets. We plan to extend our investigations by comparing the output of DAGAF with other causality-based tabular data generation methods [30–32]. This comparison will be conducted using more appropriate metrics, such as Cross-Validation Score (CVS) [53], Kolmogorov-Smirnov (KS) test [54] or Chi-Square test [55], to offer a more comprehensive qualitative analysis of the data generation capabilities of our framework.

In essence, our approach identifies DAG structures by integrating MLE with adversarial loss components and enforcing an acyclicity constraint via an augmented Lagrangian. Consequently, our model exhibits high computational complexity and a complicated loss function. We plan to explore more efficient structure learning methods and adversarial loss training to develop a faster model that relies exclusively on the Wasserstein loss.

The proposed causal learning-based synthetic data generation framework is closely connected to recent advances in generative modeling, including Digital Twins and transformer-based architectures. DAG learning naturally embodies the essence of attention mechanisms by identifying the direct causal parents of each variable, similar to how transformers dynamically weigh relevant dependencies. Moreover, our approach aligns with the principles of Digital Twins, which aim to simulate real-world systems and generate data that accurately reflect their underlying causal structures. This study establishes a unified framework for causal discovery and generative modeling, leveraging adversarial learning, MSE, MMD, and KLD regularization to ensure robust structure learning and high-fidelity synthetic data generation.

Our future work will include several mitigation strategies to address missing data. We will employ data imputation techniques such as mean/mode imputation, multiple impu-

tation, and advanced methods like matrix completion and variational autoencoders (VAEs), while acknowledging that imputation introduces assumptions about missingness that may bias results. Additionally, we will leverage structural information, using partial knowledge of the directed acyclic graph (DAG), such as domain expertise, to help compensate for missing data. Another approach involves explicitly modeling missingness mechanisms by introducing a missingness variable into the DAG to represent whether a specific variable is missing. Moreover, we will also apply causal inference techniques, including latent variable models and specialized methods designed for incomplete data, to ensure robust and accurate analyses.

Finally, as part of our future work, we will examine the flexibility of our framework by experimenting with different combinations of FCM and DGM to identify the optimal configuration for enhancing the output quality of the proposed method and extending its application to time-series data. For example, recently developed concepts such as digital twin layer via multi-attention networks [56, 57] can offer exciting avenues for future exploration. This can be achieved through their multi-attention mechanisms, which effectively highlight relevant features while filtering out irrelevant noise and misleading correlations. Their ability to adaptively handle mixed-variable datasets, align higher-order statistics of distributions, and dynamically capture multi-modal dependencies can complement the causal discovery framework presented in this work. Future research could focus on integrating these mechanisms to improve the robustness and scalability of causal discovery and synthetic data generation for complex real-world datasets. Such integration would bridge the gap between foundational theoretical insights and practical applications, addressing challenges like non-i.i.d. data and variable heterogeneity while enabling the creation of robust, high-fidelity synthetic datasets for downstream tasks.

The novel setup will be supported by an extensive study of hyper-parameters to determine their best possible values, resulting in more realistic data samples generated through a more accurately simulated generative process.

Appendix A Mathematical proofs

This appendix provides the proofs associated with the propositions and theorems found in Section 3.

A.1 Proof of proposition 1

Proposition 1 *Let the ground-truth DAG $\mathcal{G}_{\mathcal{A}}$ be uniquely identifiable from $P(\mathbf{X})$, then minimizing the adversarial loss ensures that the implicitly generated distribution $P_{G_A}(\tilde{\mathbf{X}})$ aligns with $P(\mathbf{X})$.*

$$\inf_{G_A \in \mathbb{D}} \mathcal{L}_{adv}(\mathbf{X}, \tilde{\mathbf{X}}) = 0 \implies P_{G_A}(\tilde{\mathbf{X}}) = P(\mathbf{X}) \quad \textit{and consequently} \quad G_A = \mathcal{G}_{\mathcal{A}}.$$

Proof Let $\tilde{\mathbf{X}} \sim P_{G_A}(\tilde{\mathbf{X}})$ denote the distribution generated by a DAG G_A. Assume the true data distribution $\mathbf{X} \sim P(\mathbf{X})$ is generated from the ground-truth graph $\mathcal{G}_{\mathcal{A}}$. The adversarial loss $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$ based on the Wasserstein distance $\mathbb{W}_p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}}))$ is expressed in (2). Therefore, minimizing $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$ aligns $P_{G_A}(\tilde{\mathbf{X}})$ with $P(\mathbf{X})$:

$$P_{G_A}(\tilde{\mathbf{X}}) = P(\mathbf{X}) \implies \mathbb{W}_p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})) = 0,$$

at the global minimum of the distance metric

$$\mathbb{W}_p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})) = 0 \implies P_{G_A}(\tilde{\mathbf{X}}) = P(\mathbf{X}).$$

For $G_A \neq \mathcal{G}_{\mathcal{A}}$, the generated distribution $P_{G_A}(\tilde{\mathbf{X}})$ cannot match $P(\mathbf{X})$ because the structure G_A is incorrect:

$$\mathbb{W}_p(P(\mathbf{X}), P_{G_A}(\tilde{\mathbf{X}})) > 0.$$

Therefore, minimizing $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$ aligns $P_{G_A}(\tilde{\mathbf{X}})$ with $P(\mathbf{X})$, and the identifiability assumption guarantees that this occurs only when $G_A = \mathcal{G}_{\mathcal{A}}$, thus concluding the proof. □

A.2 Proof of proposition 2

Proposition 2 *The MSE loss ensures point-wise alignment between the data and the prediction of the model, improving the smoothness of the gradient and the stability of adversarial optimization.*

$$\inf_{G_A \in \mathbb{D}} \mathcal{L}_{MSE}(\mathbf{X}, \tilde{\mathbf{X}}) = 0 \Rightarrow \forall i, \tilde{\mathbf{X}}_i = \mathbf{X}_i$$

Proof From the definition of $\mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$, it is minimized if and only if:

$$\|\mathbf{X}_i - \tilde{\mathbf{X}}_i\|^2 = 0, \quad \forall \mathbf{X}_i \in P(\mathbf{X}), \forall \tilde{\mathbf{X}}_i \in P_{G_A}(\tilde{\mathbf{X}}), \quad \forall i \in \{1, \ldots, n\},$$

which implies $\mathbf{X}_i = \tilde{\mathbf{X}}_i, \quad \forall i \in \{1, \ldots, n\}$.

The gradient of $\mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$ with respect to the model parameters θ (which define G_A) is given by:

$$\nabla_\theta \mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot ||\mathbf{X}_i - \tilde{\mathbf{X}}_i|| \cdot \nabla_\theta \tilde{\mathbf{X}}_i.$$

As the model predictions $\tilde{\mathbf{X}}_i$ approach the true data $\mathbf{X}_i$ the residual distance $\|\mathbf{X}_i - \tilde{\mathbf{X}}_i\|$ becomes smaller:

$$\|\mathbf{X}_i - \tilde{\mathbf{X}}_i\| \to 0 \quad \implies \quad \nabla_\theta \mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}}) \to 0.$$

This behavior arises because the residual distance $\|\mathbf{X}_i - \tilde{\mathbf{X}}_i\|$ directly scales the gradient. As $\tilde{\mathbf{X}}_i$ aligns with $\mathbf{X}_i$, the gradient magnitude decreases, reducing the size of updates during optimization. Therefore, the MSE loss offers optimization stability by smooth gradients. By steady convergence as $\tilde{\mathbf{X}}_i \to \mathbf{X}_i$, preventing oscillatory behavior, thus concluding the proof. □

A.3 Proof of proposition 3

Proposition 3 *The $\mathcal{L}_{KLD}(\mathbf{X}, \tilde{\mathbf{X}})$ regularization provides a statistical prior on the learned distribution $P_{G_A}(\tilde{\mathbf{X}})$, ensuring it adheres to a Gaussian assumption. It also acts as a stabilizing factor in optimization, particularly under the additive Gaussian noise model. It complements the adversarial and MSE losses, ensuring both alignment and smoothness of $P_{G_A}(\tilde{\mathbf{X}})$.*

Proof This term is used to ensure that the residual noise $\mathcal{Z}_j$ conditioned on Pa_j is Gaussian. The residual $\mathcal{Z}_j$ can be expressed as $\mathcal{Z}_j = X_j - f_j(Pa_j)$. By minimizing $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$, the model is encouraged to fit f_j such that $\mathcal{Z}_j \sim \mathcal{N}(0, \sigma_j^2)$, namely: $P(\mathcal{Z}_j|Pa_j) \approx \mathcal{N}(0, \sigma_j^2)$. Let $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ act as a penalty on deviations of $P(\mathcal{Z}_j|Pa_j)$ from $\mathcal{N}(0, \sigma_j^2)$. The gradient of $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ with respect to G_A is:

$$\nabla_{G_A}\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}}) = \sum_{j=1}^{d} \mathbb{E}_{Pa_j}\left[\nabla_{G_A} \log \frac{P(\mathcal{Z}_j \mid Pa_j)}{\mathcal{N}(\mathcal{Z}_j; 0, \sigma_j^2)}\right].$$

The term $\log \mathcal{N}(\mathcal{Z}_j; 0, \sigma_j^2)$ is quadratic in $\mathcal{Z}_j$, making $\nabla_{G_A}\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ smooth and less sensitive to small variations in G_A. This prevents overfitting to noise in X_j, stabilizing the optimization of f_j. Hence, the KLD term can improve the overall stability of our model by approximating the implicitly generated distribution $P_{G_A}(\tilde{\mathbf{X}})$ to a normal (Gaussian) distribution.

The KLD term also complements other loss terms. The adversarial loss $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$ ensures global alignment of $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$, but does not directly enforce the additive Gaussian assumption. The MSE loss $\mathcal{L}_{\text{MSE}}(\mathbf{X}, \tilde{\mathbf{X}})$ focuses on point-wise alignment of $\mathbf{X}_i$ and $\tilde{\mathbf{X}}_i$, but does not account for statistical properties of $\mathcal{Z}_j$. The KLD regularization $\mathcal{L}_{\text{KLD}}(\mathbf{X}, \tilde{\mathbf{X}})$ explicitly enforces the Gaussianity of $\mathcal{Z}_j$, ensuring $\mathcal{Z}_j$ matches the additive Gaussian assumption, preventing f_j from overfitting to non-Gaussian noise, thus concluding the proof. □

A.4 Proof of proposition 4

Proposition 4 *Minimizing the Maximum Mean Discrepancy (MMD) loss $\mathcal{L}_{MMD}(\mathbf{X}, \tilde{\mathbf{X}})$ aligns higher-order statistics of $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$, complementing adversarial loss to achieve overall distributional alignment.*

Proof The MMD loss term is

$$\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{n}\sum_{i\neq j}^{n} k(\mathbf{X}_i, \mathbf{X}_j) - \frac{2}{n}\sum_{i\neq j}^{n} k(\mathbf{X}_i, \tilde{\mathbf{X}}_j) + \frac{1}{n}\sum_{i\neq j}^{n} k(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j).$$

The gradient of $\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$ with respect to the parameters θ defining the model G_A can be written as:

$$\nabla_\theta \mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}}) = 2(\mathbb{E}_{\tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[\nabla_\theta k(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)] - \mathbb{E}_{\mathbf{X}\sim P(\mathbf{X}), \tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[\nabla_\theta k(\mathbf{X}_i, \tilde{\mathbf{X}}_j)]),$$

where $\tilde{\mathbf{X}} \sim P_{G_A}(\tilde{\mathbf{X}})$ are samples from the model-generated distribution, $\mathbf{X} \sim P(\mathbf{X})$ are samples from the true distribution and $k(\mathbf{X}, \tilde{\mathbf{X}})$ is a positive-definite kernel, often chosen as a Gaussian kernel or other characteristic kernel.

The kernel function $k(\mathbf{X}, \tilde{\mathbf{X}})$ implicitly captures higher-order statistics of the distributions $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$, including the internal consistency of the model distribution via the third term in $\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$, $\mathbb{E}_{\tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[k(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)]$, which aligns model-generated samples $\tilde{\mathbf{X}}_i$ and $\tilde{\mathbf{X}}_j$ to ensure that the higher-order moments within $P_{G_A}(\tilde{\mathbf{X}})$ are coherent. It also allows alignment with the true distribution via the second term, $\mathbb{E}_{\mathbf{X}\sim P(\mathbf{X}), \tilde{\mathbf{X}}\sim P_{G_A}(\tilde{\mathbf{X}})}[k(\mathbf{X}_i, \tilde{\mathbf{X}}_j)]$.

$\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$ explicitly captures higher-order discrepancies through the kernel-induced feature mappings $k(.)$. This provides a complementary mechanism to adversarial losses, ensuring both global and fine-grained alignment between $P(\mathbf{X})$ and $P_{G_A}(\tilde{\mathbf{X}})$. Together, $\mathcal{L}_{\text{MMD}}(\mathbf{X}, \tilde{\mathbf{X}})$ and $\mathcal{L}_{\text{adv}}(\mathbf{X}, \tilde{\mathbf{X}})$ form a robust framework for distributional alignment, addressing both large-scale and higher-order mismatches, thus completing the proof. □

A.5 Proof of proposition 5

Proposition 5 *Under the Additive Noise Model (ANM), Linear non-Gaussian Acyclic Model (LiNGAM) or Post-Nonlinear Model (PNL) assumption, there exists a unique DAG $\mathcal{G}_\mathcal{A}$ capable of defining the observed joint distribution $P(\mathbf{X})$.*

Proof We split the proposition into two lemmas for identifiability under: 1) LiNGAM and ANM; 2) PNL, respectively. □

Lemma 6 *Under the additive noise model (ANM) or the linear non-Gaussian acyclic model (LiNGAM) assumption, the*

true DAG $\mathcal{G}_{\mathcal{A}}$ is uniquely identifiable from $P(\mathbf{X})$

$$P(\mathbf{X}) \neq P'(\mathbf{X}) \implies \mathcal{G}_{\mathcal{A}} \neq \mathcal{G}'_{\mathcal{A}'}.$$

Proof Let the dataset χ consist of $X = \{X_1, ..., X_d\}$ data attributes, where each X_j is generated under the ANM or LiNGAM assumption, both described using the following equation:

$$X_j = f_j(Pa_j) + \mathcal{Z}_j,$$

where $f_j : \mathbb{R}^d \to \mathbb{R}$ are deterministic functions (nonlinear in ANM, linear in LiNGAM), $\mathcal{Z}_j \sim P(\mathcal{Z})$ are independent noise variables (non-Gaussian in LiNGAM, Gaussian in ANM), Pa_j represents the set of direct parents of X_j in the DAG.

For both ANM and LiNGAM, the independence of $\mathcal{Z}_j$ from Pa_j plays a crucial role: $\mathcal{Z}_j \perp\!\!\!\perp Pa_j$. The independence of $\mathcal{Z}_j$ in the true DAG $\mathcal{G}_{\mathcal{A}}$ imposes strong constraints on the functional relationships in $\mathcal{G}_{\mathcal{A}}$:

$$P(\mathcal{Z}_j) = P_{\mathcal{Z}_j}(X_j - f_j(Pa_j)),$$

where $\mathcal{Z}_j$ is the independent noise term.

In the case when $\mathcal{G}'_{\mathcal{A}'} \neq \mathcal{G}_{\mathcal{A}}$, the functional relationships $f'_j \in \mathcal{G}'_{\mathcal{A}'}$ must satisfy:

$$P(\mathcal{Z}'_j) = P_{\mathcal{Z}'_j}(X_j - f'_j(Pa'_j)),$$

where $\mathcal{Z}'_j$ are the noise terms under $\mathcal{G}'_{\mathcal{A}'}$.

However, when $\mathcal{G}'_{\mathcal{A}'} \neq \mathcal{G}_{\mathcal{A}}$, the new functional relationships f'_j will be different from f_j in the true DAG. Furthermore, the new noise terms $\mathcal{Z}'_j$ will not remain independent of Pa'_j because the independence of $\mathcal{Z}_j$ is specific to the true causal structure in $\mathcal{G}_{\mathcal{A}}$. This implies that $\mathcal{G}'_{\mathcal{A}'}$ cannot satisfy the independence assumptions simultaneously with $\mathcal{G}_{\mathcal{A}}$, leading to a contradiction.

Hence, under the assumptions of the ANM with nonlinear functions and independent noise or the LiNGAM model with linear functions and non-Gaussian noise, there exists no other DAG $\mathcal{G}'_{\mathcal{A}'} \neq \mathcal{G}_{\mathcal{A}}$ that can generate the same observational data distribution $P(\mathbf{X})$. Therefore, the true DAG $\mathcal{G}_{\mathcal{A}}$ is uniquely identifiable only from $P(\mathbf{X})$, thus concluding the proof. □

Lemma 7 *Under the Post-Nonlinear (PNL) model assumption, there exists an identifiable DAG $\mathcal{G}_{\mathcal{A}}$ that generates the observed joint distribution of the data variables $\{X_1, ..., X_d\}$.*

Proof Let χ be a dataset consisting of $\{X_1, ..., X_d\}$ data attributes, where each X_j is described as follows:

$$X_j := g_j(f_j(Pa_j) + \mathcal{Z}_j), \forall j, \mathcal{Z}_j \perp\!\!\!\perp f_j(Pa_j), \mathcal{Z}_j \sim \mathcal{N}(\mu, \sigma_j^2),$$

where Pa_j is the set of parent nodes for X_j, f_j are nonlinear functions modeling parent contributions, g_j is a nonlinear function applied post-summation and $\mathcal{Z}_j$ is an independent Gaussian noise term, satisfying $\mathcal{Z}_j \perp\!\!\!\perp Pa_j$.

Moreover, let N_j be the input to g_j such that:

$$N_j = f_j(Pa_j) + \mathcal{Z}_j.$$

Under the assumption that Pa_j is the true parent set, the noise term $\mathcal{Z}_j$ is independent of its parents:

$$\mathcal{Z}_j \perp\!\!\!\perp Pa_j.$$

In addition, g_j does not affect the independence structure. Thus, for the true set of parents Pa_j, the residual noise $\mathcal{Z}_j$ remains independent of the parent variables.

Under this setting, the statistical relationship between X_j, its parents, and the residual noise satisfies specific invariances:

$$P(X_j, Pa_j) = P(X_j|Pa_j)P(Pa_j),$$

where $P(X_j|Pa_j)$ is derived from the PNL structure.

Now, consider any alternative parent set $Pa'_j \neq Pa_j$. For this incorrect set of parents, the residual noise $\mathcal{Z}_j$ is reconstructed as:

$$\mathcal{Z}_j = N_j - f_j(Pa'_j).$$

In this case, the core independence condition $\mathcal{Z}_j \perp\!\!\!\perp Pa'_j$ is violated. Therefore, when the parent set is incorrect, the residual noise $\mathcal{Z}_j$ will exhibit statistical dependencies with the variables in Pa'_j. This implies that the conditional distribution $P(X_j|Pa'_j)$ cannot reproduce the same invariance due to the introduced dependencies, thus concluding the proof. □

Corollary 7.1 *Under Lemmas 6 and 7, the uniqueness property of G_A allows us to reconstruct the generative process of* $\mathbf{X}$.

Corollary 7.1 implies that under the causal model assumption employed in DAGAF, we can accurately generate synthetic samples with preserved causal structures, which is only possible if $G_A = \mathcal{G}_{\mathcal{A}}$. In turn, this implies that the implicitly generated distribution $P_{G_A}(\tilde{\mathbf{X}})$ is the same as the observed distribution $P(\mathbf{X})$. Therefore, we have demonstrated that there exists a single unique DAG capable of constructing the input data distribution, thus concluding the proof.

Appendix B Ablation study

We conducted an ablation study to determine the optimal configuration of the terms in the loss function for

Table 8 DAGAF ablation study

Loss function	SHD			
	Sachs	ECOLI70	MAGIC-IRRI	ARTH150
w/o recon loss	21	115	163	377
recon loss (MSE)	14	91	117	288
recon loss (NLL)	16	106	132	320
MSE + MMD	10	57	80	189
NLL + MMD	14	91	117	288
MSE + KLD	12	69	99	221
NLL + KLD	12	69	99	221
MSE + KLD + MMD	9	52	71	175
NLL + KLD + MMD	11	60	86	197

Step 1. We carried out nine experiments on the Sachs, ECOLI70, MAGIC-IRRI and ARTH150 datasets under the ANM assumption, testing various combinations of loss terms. These continuous (Gaussian) datasets are available at https://www.bnlearn.com/bnrepository/. All cases include the Wasserstein-1 distance. The first configuration is labeled "**w/o recon loss**", where the reconstruction loss with its regularization is excluded from the training algorithm. The rest are named according to the terms included in the reconstruction loss, such as MSE [58] and NLL [59]. We also tested combinations of additional terms such as MMD [36] and KLD [35]. The results of this study are shown in Table 8.

The ablation study reveals the optimal combination of loss terms for our method. As shown in Table 8, the best set of loss terms in Step 1 includes MSE, KLD, MMD, and adversarial training. Further details on each of these metrics and regularization are provided in Section 3.1.

Appendix C Sensitivity analysis

To ensure model robustness, we perform a sensitivity analysis to examine how the training responds to different hyper-parameter settings. This study measures the accuracy of DAG reconstruction (i.e., SHD) under various hyper-parameters, including learning and dropout rates (**lr, dropout**), noise vector and batch sizes (**z-size, batch-size**). We begin with a baseline setting of **lr = 0.001, dropout = 0.5, z-size = 1, batch-size = 100**, then modify each value individually to observe the changes in SHD. All experiments were conducted on the Sachs dataset by applying the ANM causal model, and the results are presented in Table 9.

Table 9 DAGAF sensitivity analysis

Hyper-parameters	Sachs Dataset
	SHD
lr = 3e-3, dropout = 0.5, z-size = 1, batch-size = 100	9
lr = 3e-3, dropout = 0.0, z-size = 1, batch-size = 100	10
lr = 3e-3, dropout = 0.5, z-size = 2, batch-size = 100	10
lr = 3e-3, dropout = 0.5, z-size = 5, batch-size = 100	11
lr = 3e-3, dropout = 0.5, z-size = 1, batch-size = 500	9
lr = 3e-3, dropout = 0.5, z-size = 1, batch-size = 1000	10
lr = 2e-4, dropout = 0.5, z-size = 1, batch-size = 100	11
lr = 1e-3, dropout = 0.5, z-size = 1, batch-size = 100	12

The results from Table 9 indicate that lowering the learning and dropout rates significantly affects the performance of our model. On the other hand, increasing the size of the noise vector and the input data batch results in only minor variations in the accuracy of the algorithm.

Appendix D Additional results

In this section, we present further examples to reinforce the data quality analysis discussed in Section 5.4. We provide real-synthetic statistical comparisons for all features (Table 10), additional visualizations of the synthetic feature distributions (Fig. 8), and the remaining machine learning regression results (Fig. 9).

Table 10 Mann-Whitney t-test results for all real and synthetic features to supplement Fig. 7

Feature	p-value
x1	7.7952e-07
x2	0.5004
x3	0.1683
x4	0.0020
x5	0.8563
x6	0.9127
x7	0.0364
x8	0.1747
x9	0.2089
x10	6.4502e-26

We observe some failure cases, where the real and synthetic features differ significantly ($p < 0.05$)

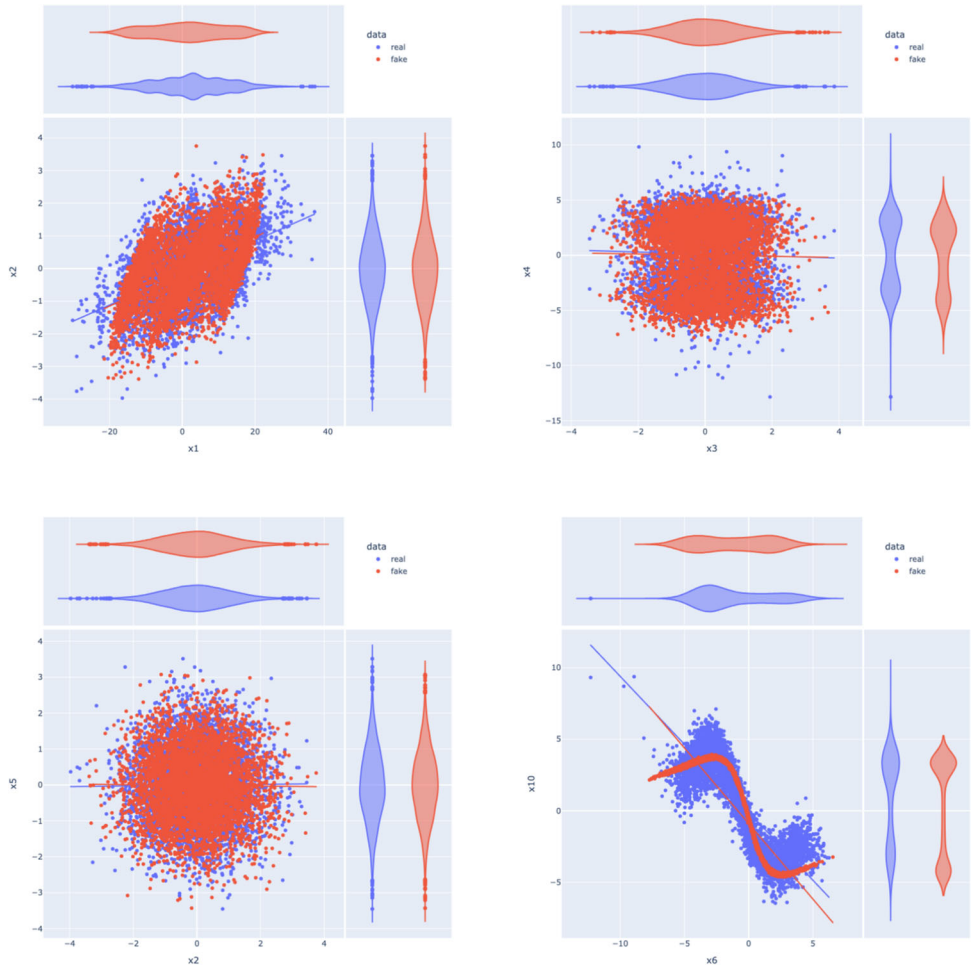


Fig. 8 Further examples of the synthetic joint and marginal distributions for our method on the dataset presented in Section 5.4. We observe multiple cases with different distribution shapes. Additionally, we depict one case of severe mode collapse (bottom) in the produced data from DAGAF

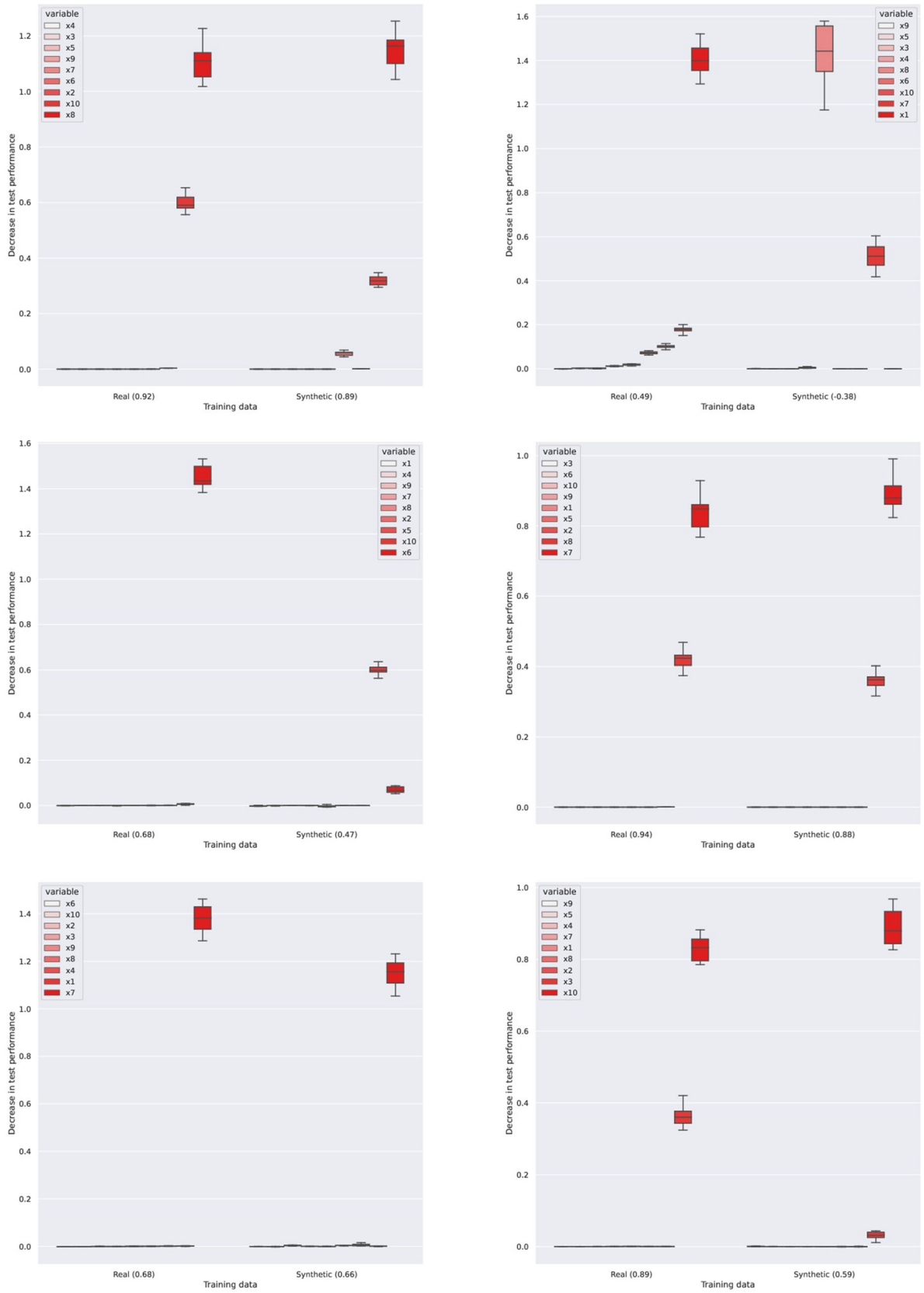


Fig. 9 Remaining examples of feature importances to supplement the results in Section 5.4. We observe some failure cases, where the synthetic features differ significantly from their real counterparts

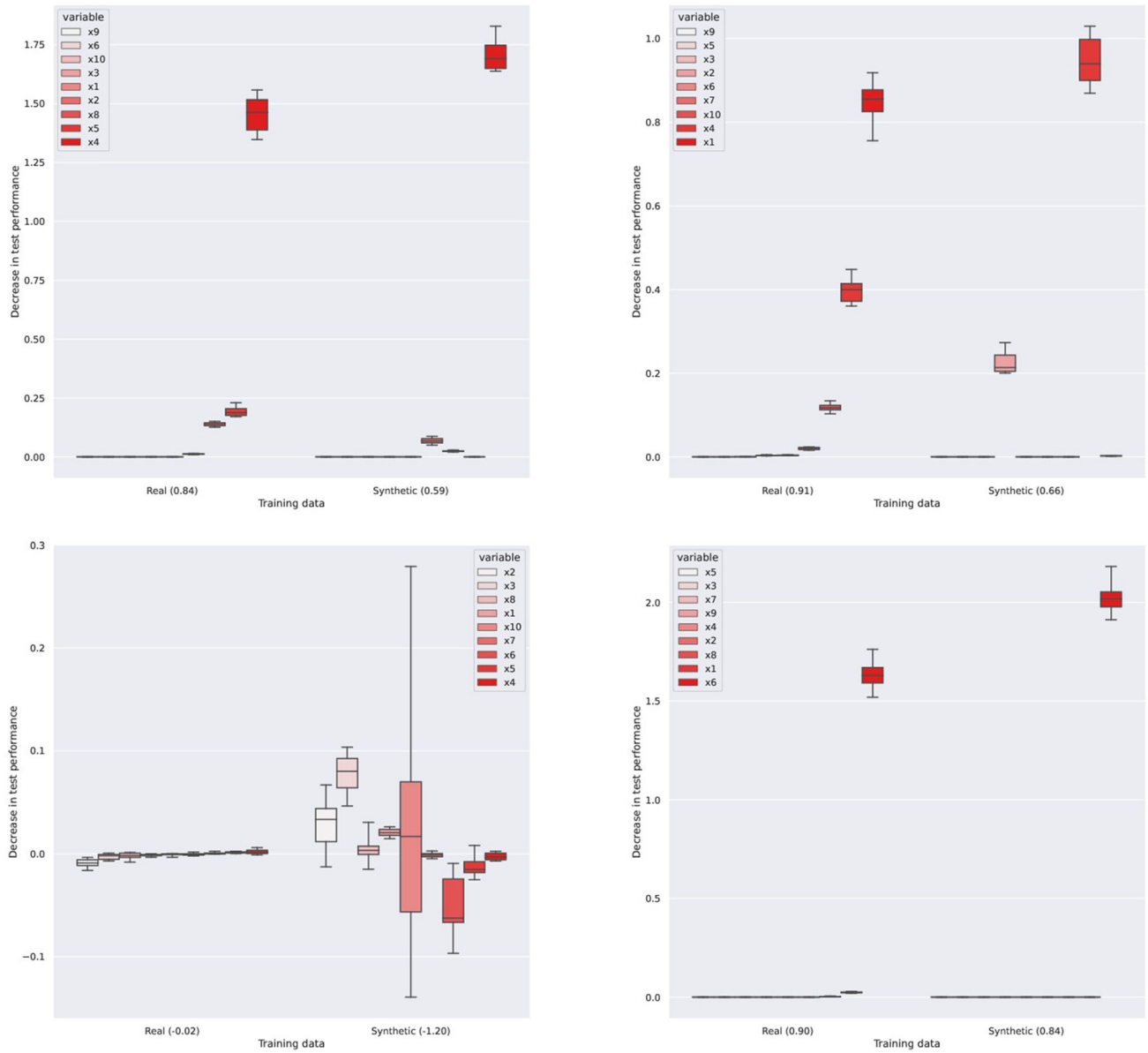


Fig. 9 continued

Appendix E DAGAF pseudo-code

$\lambda \leftarrow 0, c \leftarrow 1$
$current\_h(A^{L_0}(f)) \leftarrow \infty, h\_tol \leftarrow 1e-8$
$k\_max\_iter \leftarrow 100, epochs \leftarrow 300$
for k < k_max_iter **do**
  while $c < 1e+20$ **do**
    for epoch < $epochs$ **do**

      if $pnl == True$ **then** ▷ The beginning of the Causal Discovery (CD) Step
        $\tilde{X} := \{g_1(f_1(Pa_1; W_1^1, ..., W_1^L) + \mathcal{Z}_1), ..., g_d(f_d(Pa_d; W_d^1, ..., W_d^L) + \mathcal{Z}_d)\}$
      else
        $\tilde{X} := \{f_1(Pa_1; W_1^1, ..., W_1^L) + \mathcal{Z}_1, ..., f_d(Pa_d; W_d^1, ..., W_d^L) + \mathcal{Z}_d\}$
      end if
      DiscLoss = $\mathcal{L}_{adv}(\mathbf{X}, \tilde{\mathbf{X}})$
      GenLoss = $\mathcal{L}_G(\mathbf{X})$
      RecLoss = $\mathcal{L}_{MSE}(\mathbf{X}, \tilde{\mathbf{X}}) + \mathcal{L}_{KLD}(\mathbf{X}, \tilde{\mathbf{X}}) + \mathcal{L}_{MMD}(\mathbf{X}, \tilde{\mathbf{X}}) + \frac{c}{2}|h(A^{L_0})|^2 + \lambda h(A^{L_0})$
      PnlLoss = $\mathcal{L}_{PNL}(\hat{\mathbf{X}}, \tilde{\mathbf{X}})$ ▷ if PNL is assumed
      DiscGradients = DiscLoss.backward()
      GenGradients = GenLoss.backward()
      RecGradients = RecLoss.backward()

PnlGradients = PnlLoss.backward() ▷ if PNL is assumed
DiscParameters = DiscParameters - $1e-3$ * DiscGradients
GenParameters = GenParameters - $1e-3$ * GenGradients
RecParameters = RecParameters - $1e-3$ * RecGradients
PnlParameters = PnlParameters - $1e-3$ * PnlGradients ▷ if PNL is assumed
$DS\{W^{L_0}\} \leftarrow CD\{W^{L_0}\}$ ▷ Parameter transfer between steps

if $pnl == True$ **then** ▷ The beginning of the Data Synthesis (DS) Step
$\tilde{X} := \{g_1(G_1(Pa_1; W_1^1, ..., W_1^L) + Z_1), ..., g_d(G_d(Pa_d; W_d^1, ..., W_d^L) + Z_d)\}$
else
$\tilde{X} := \{G_1(Pa_1; W_1^1, ..., W_1^L) + Z_1, ..., G_d(Pa_d; W_d^1, ..., W_d^L) + Z_d\}$
end if
DiscLoss = $\mathcal{L}_{adv}(\mathbf{X}, \tilde{\mathbf{X}})$
GenLoss = $\mathcal{L}_G(Z)$
DiscGradients = DiscLoss.backward()
GenGradients = GenLoss.backward()
DiscParameters = DiscParameters - $1e-3$ * DiscGradients
GenParameters = GenParameters - $1e-3$ * GenGradients
end for
if $h(A^{L_0}(f)) > 0.25$ **then**
$c \leftarrow c * 10$
else
$break$
end if
end while
$current\_h(A^{L_0}(f)) \leftarrow h(A^{L_0}(f))$
$\lambda \leftarrow c * current\_h(A^{L_0}(f))$
if $current\_h(A^{L_0}(f)) \leq h\_tol$ **then**
$break$
end if
end for

Author Contributions Hristo Petkov (First Author) is responsible for software development, theoretical analysis, conducting causal experiments and draft preparation. Calum MacLellan (Second Author) is responsible for performing data synthesis experiments and draft revision. Feng Dong (Third Author) is responsible for overall draft proofreading and refactoring.

Funding The authors declare that their work has been funded by the United Kingdom Medical Research Council (Grant Reference: MR/X005925/1) throughout the duration of their associated research project (Virtual Clinical Trial Emulation with Generative AI Models, Duration: Sept 2022 - Feb 2023). EPSRC (Grant Refference: EP/X029778/1), Causal Counterfactual visualization for human causal decision making 2023-2025.

Data availability The authors confirm that all data (with their corresponding repository and citation links) relevant to the research carried out to support their work are included in this article.

Declarations

Competing interests The authors declare that they have no competing financial or non-financial interests in relation to this work.

Ethical and informed consent for data used Not applicable.

References

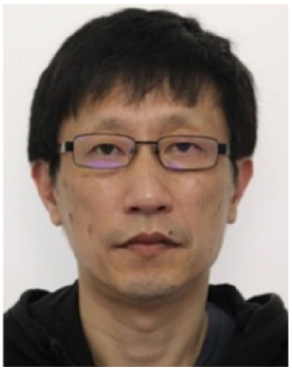
1. Choi J, Chapkin RS, Ni Y (2020) Supplementary material of "bayesian causal structural learning with zero-inflated poisson bayesian networks". In: NIPS
2. Foraita R, Friemel J, Günther K, Behrens T, Bullerdiek J, Nimzyk R, Ahrens W, Didelez V (2020) Causal discovery of gene regulation with incomplete data. J Royal Stat Soc Series A (Stat Soc) 183
3. Shen X, Ma S, Vemuri P, Simon GJ (2020) Challenges and opportunities with causal discovery algorithms: Application to alzheimer's pathophysiology. Sci Rep 10
4. Moneta A, Entner D, Hoyer PO, Coad A (2013) Causal inference by independent component analysis: Theory and applications. Econometric & Statistical Methods - Special Topics eJournal, Econometrics
5. Opgen-Rhein R, Strimmer K (2007) From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. BMC Syst Biol 1:37–37
6. Londei A, D'Ausilio A, Basso D, Belardinelli MO (2006) A new method for detecting causality in fmri data of cognitive processing. Cogn Process 7:42–52
7. Ebert-Uphoff I, Deng Y (2012) Causal discovery for climate research using graphical models. J Clim 25:5648–5665
8. Runge J, Bathiany S, Bollt EM, Camps-Valls G, Coumou D, Deyle ER, Glymour C, Kretschmer M, Mahecha MD, Muñoz-Marí J, Nes EH, Peters J, Quax R, Reichstein M, Scheffer M, Scholkopf B, Spirtes P, Sugihara G, Sun J, Zhang K, Zscheischler J (2019) Inferring causation from time series in earth system sciences. Nat Commun 10

9. Morgan SL, Winship C (2007) Counterfactuals and causal inference: Methods and principles for social research. In: Cambridge University Press
10. Spirtes PL, Glymour C, Scheines R (2001) Causation, Prediction, and Search, 2nd edn. MIT press, Cambridge, Massachusetts
11. Spirtes P, Glymour C, Scheines R, Kauffman SA, Aimale V, Wimberly FC (2000) Constructing bayesian network models of gene expression networks from microarray data. In: Atl Symp Comput Biol
12. Colombo D, Maathuis MH, Kalisch M, Richardson TS (2011) Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann Stat 40:294–321
13. Chickering DM (2003) Optimal structure identification with greedy search. J Mach Learn Res 3:507–554
14. Alonso-Barba JI, Ossa L, Gamez JA, Puerta JM (2011) Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes. In: Int J Approx Reason
15. Hauser A, Bühlmann P (2011) Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. J Mach Learn Res 13(1):2409–2464
16. Zheng X, Aragam B, Ravikumar P, Xing EP (2018) Dags with no tears: Continuous optimization for structure learning. In: Neural Inf Process Syst
17. Yu Y, Chen J, Gao T, Yu M (2019) Dag-gnn: Dag structure learning with graph neural networks. In: Int Conf Mach Learn
18. Lachapelle S, Brouillard P, Deleu T, Lacoste-Julien S (2020) Gradient-based neural dag learning. Arxiv arXiv:1906.02226
19. Petkov HH, Hanley C, Dong F (2022) Dag-wgan: Causal structure learning with wasserstein generative adversarial networks. ArXiv arXiv:2204.00387
20. Peters J, Mooij J, Janzing D, Schölkopf B (2012) Identifiability of causal graphs using functional models. arXiv preprint arXiv:1202.3757
21. Shimizu S, Hoyer PO, Hyvärinen A, Kerminen AJ (2006) A linear non-gaussian acyclic model for causal discovery. J Mach Learn Res 7:2003–2030
22. Hoyer PO, Janzing D, Mooij JM, Peters J, Scholkopf B (2008) Nonlinear causal discovery with additive noise models. In: Neural Inf Process Syst
23. Zhang K, Hyvärinen A (2009) On the identifiability of the post-nonlinear causal model. In: Conf Uncertain Artif Intell
24. Zhang K, Hyvärinen A (2010) Distinguishing causes from effects using nonlinear acyclic causal models. In: Causality: Objectives and Assessment, PMLR, pp 157–164
25. Uemura K, Takagi T, Takayuki K, Yoshida H, Shimizu S (2022) A multivariate causal discovery based on post-nonlinear model. In: Conference on Causal Learning and Reasoning, PMLR, pp 826–839
26. Chung Y, Kim J, Yan T, Zhou H (2019) Post-nonlinear causal model with deep neural networks
27. Hoang N, Duong B, Nguyen T (2024) Enabling causal discovery in post-nonlinear models with normalizing flows. arXiv preprint arXiv:2407.04980
28. Keropyan G, Strieder D, Drton M (2023) Rank-based causal discovery for post-nonlinear models. In: International conference on artificial intelligence and statistics, PMLR, pp 7849–7870
29. Zhang T, Yin F, Luo ZQ (2024) Post-nonlinear causal relationship with finite samples: A maximal correlation perspective
30. Breugel B, Kyono T, Berrevoets J, Schaar M (2021) Decaf: Generating fair synthetic data using causally-aware generative networks. In: Neural Inf Process Syst
31. Wen B, Colon LO, Subbalakshmi KP, Chandramouli R (2021) Causal-tgan: Generating tabular data using causal generative adversarial networks. ArXiv arXiv:2104.10680
32. Rajabi A, Garibay OO (2021) Tabfairgan: Fair tabular data generation with generative adversarial networks. Mach Learn Knowl Extr 4:488–501
33. Pearl J (2009) Causality. Cambridge university press, ???
34. Zheng X, Dan C, Aragam B, Ravikumar P, Xing EP (2019) Learning sparse nonparametric dags. ArXiv arXiv:1909.13189
35. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22:79–86
36. Tolstikhin IO, Sriperumbudur BK, Schölkopf B (2016) Minimax estimation of maximum mean discrepancy with radial kernels. NIPS
37. Khemakhem I, Monti R, Leech R, Hyvarinen A (2021) Causal autoregressive flows. In: International conference on artificial intelligence and statistics, PMLR, pp 3520–3528
38. Gao Y, Shen L, Xia ST (2021) Dag-gan: Causal structure learning with generative adversarial nets. In: ICASSP 2021 - 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 3320–3324. https://doi.org/10.1109/ICASSP39728.2021.9414770
39. Peters J, Mooij JM, Janzing D, Schölkopf B (2013) Causal discovery with continuous additive noise models. J Mach Learn Res 15:2009–2053
40. He K, Zhang X, Ren S, Sun J (20115) Deep residual learning for image recognition. 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
41. Ng I, Zhu S, Chen Z, Fang Z (2019) A graph autoencoder approach to causal structure learning. ArXiv arXiv:1911.07420
42. Bertsekas D (1999) Nonlinear Programming. Athena Scientific, 2nd Ed
43. Petkov HH, Dong F (2023) Efficient generative adversarial dag learning with no-curl. 2023 international conference automatics and informatics (ICAI), pp 164–169
44. Khemakhem I, Monti RP, Leech R, Hyvärinen A (2020) Causal autoregressive flows. ArXiv arXiv:2011.02268
45. Wehenkel A, Louppe G (2020) Graphical normalizing flows. In: Int Conf Artif Intell Stat
46. Mamaghan AMK, Dittadi A, Bauer S, Johansson KH, Quinzan F (2024) Diffusion-based causal representation learning. Entropy 26
47. Charpentier B, Kibler S, Günnemann S (2022) Differentiable dag sampling. ArXiv ArXiv:2203.08509
48. Jongh M, Druzdzel MJ (2009) A comparison of structural distance measures for causal bayesian network models. Recent Adv Intell Inf Syst 443–456
49. Erdös P (1959) On random graphs, i
50. Sachs K, Perez O, Peér D, Lauffenburger DA, Nolan GP (2005) Causal protein-signaling networks derived from multiparameter single-cell data. Sci 308(5721):523–529
51. Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Phil Trans R Soc A Math Phys Eng Sci 374
52. Park G (2020) Identifiability of additive noise models using conditional variances. J Mach Learn Res 21(75):1–34
53. Stone M (1976) Cross-validatory choice and assessment of statistical predictions. J Royal Stat Soc Ser B-Methodol 36:111–133
54. Simard RJ, L'Ecuyer P (2011) Computing the two-sided kolmogorov-smirnov distribution. J Stat Softw 39:1–18
55. Williams CA (1950) The choice of the number and width of classes for the chi-square test of goodness of fit. J Am Stat Assoc 45:77–86
56. Połap D, Jaszcz A (2024) Sonar digital twin layer via multiattention networks with feature transfer. IEEE Trans Geosci Remote Sens 62:1–10
57. Thomas CK, Saad W, Xiao Y (2023) Causal semantic communication for digital twins: A generalizable imitation learning approach. IEEE J Select Areas Inf Theory 4:698–717
58. Bickel PJ, Doksum KA (2015) Mathematical statistics: Basic ideas and selected topics, volume i, 2nd Ed

59. Grof S, Transpersonal MD (1921) On the mathematical foundations of theoretical statistics. Phil Trans R Soc A 222:309–368

Hristo Petkov earned his First-Class B.Sc. degree from the Department of Computer and Information Sciences at the University of Strathclyde. He is currently working toward his Doctorate degree in the same department. His research focuses on the application of causal structure learning in medicine and healthcare through the use of deep learning theory and neural network models.

Calum MacLellan received an M.Eng. in Biomedical Engineering from the University of Glasgow, UK. He is currently pursuing an Eng.D. in Medical Devices at the University of Strathclyde, UK. His research interests include deep learning, optimisation, explainable AI, and computer vision, with a focus on healthcare applications.



Feng Dong is a Professor of Computer Science, Head of the Human Centric AI research group at the University of Strathclyde, UK. He was awarded a PhD in Zhejiang University, China. His recent research has addressed a range of issues in human centric AI to support knowledge discovery, visual data analytics, image analysis, pattern recognition and parallel computing (GPU). In particular, he is interested in causality learning from data to support the generation of synthetic data for healthcare and virtual clinical trial emulations.