Modeling Causal Mechanisms with Diffusion Models for Interventional and Counterfactual Queries

Anonymous authors
Paper under double-blind review

Abstract

We consider the problem of answering observational, interventional, and counterfactual queries in a causally sufficient setting where only observational data and the causal graph are available. Utilizing the recent developments in diffusion models, we introduce diffusion-based causal models (DCM) to learn causal mechanisms, that generate unique latent encodings. These encodings enable us to directly sample under interventions and perform abduction for counterfactuals. Diffusion models are a natural fit here, since they can encode each node to a latent representation that acts as a proxy for exogenous noise. Our empirical evaluations demonstrate significant improvements over existing state-of-the-art methods for answering causal queries. Furthermore, we provide theoretical results that offer a methodology for analyzing counterfactual estimation in general encoder-decoder models, which could be useful in settings beyond our proposed approach.

1 Introduction

Understanding the causal relationships in complex problems is crucial for making analyses, conclusions, and generalized predictions. To achieve this, we require causal models and queries. Structural Causal Models (SCMs) are generative models describing the causal relationships between variables, allowing for observational, interventional, and counterfactual queries (Pearl, 2009a). An SCM specifies how a set of endogenous (observed) random variables is generated from a set of exogenous (unobserved) random variables with prior distribution via a set of structural equations.

In this work, we focus on approximating SCM given observational data and the underlying causal DAG. Our aim is to provide a mechanism for answering all the three types of causal (observational, interventional, and counterfactual) queries. For this, we assume *causal sufficiency*, i.e., absence of hidden confounders. Note that causal sufficiency is the minimal necessary set of assumptions for causal reasoning from observational data alone.

In the SCM framework, causal queries can be answered by learning a proxy for the unobserved exogenous noise and the structural equations. This suggests that learning (conditional) *latent variable models* could be an attractive choice for the modeling procedure, as the latent serves as the proxy for the exogenous noise. In these models, the encoding process extracts the latent from an observation, and the decoding process generates the sample from the latent, approximating the structural equations.

Our Contributions. In this work, we propose and analyze the effectiveness of using a diffusion model for modeling SCMs. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have gained popularity recently due to their high expressivity and exceptional performance in generative tasks (Saharia et al., 2022; Ramesh et al., 2022; Kong et al., 2021). The primary contribution of our work is in determining how to apply diffusion models to the causal setting. This is non-trivial because diffusion models are typically used to learn a stochastic process between a Gaussian and a density. Our main idea is to model each node in the causal graph as a diffusion model and cascade generated samples in topological order to answer causal queries. For each node, the corresponding diffusion model takes as input the node and parent values to encode and decode a latent representation. To implement the diffusion model, we utilize the recently proposed *Denoising Diffusion Implicit Models* (DDIMs) (Song et al., 2021), which may be interpreted as a deterministic autoencoder model with latent variables. We leverage the deterministic forward and backward diffusion processes of DDIMs to use diffusion models as an encoder-decoder model. We refer to the resulting model as *diffusion-based causal model* (DCM) and show that this model mimics the necessary properties of an SCM. Our key contributions include:

(1) [Section 3] We propose diffusion-based causal model (DCM), a new model class for modeling structural causal models, that provides a flexible and practical framework for approximating both interventions (do-operator) and counterfactuals (abduction-action-prediction steps). We present a procedure for training a DCM given just the causal graph and observational data, and show that the resulting trained model enables sampling from the observational and interventional distributions, and facilitates answering counterfactual queries.

(2) [Section 4] Our theoretical analysis examines the accuracy of counterfactual estimates generated by the DCM, and we demonstrate that they can be bounded given some reasonable assumptions. Importantly, our analysis is not limited to diffusion models, but also applies to other encoder-decoder settings. To the best of our knowledge, these are first error bounds that explain the observed performance improvements in using latent variable models, like diffusion models, to address counterfactual queries. Another feature of this result, is that it also extends, under an additional assumption, to the more challenging multivariate case.

(3) [Section 5] We evaluate the performance of DCM on a range of synthetic datasets generated with various structural equation types for all three forms of causal queries. We find that DCM consistently outperforms existing state-of-the-art methods (Sánchez-Martin et al., 2022; Khemakhem et al., 2021). In fact, for certain interventional and counterfactual queries such as those arising with *nonadditive noise* models, DCM is better by an order of magnitude or more than these existing approaches. Additionally, we demonstrate the favorable performance of DCM on an interventional query experiment conducted on fMRI data.

Related Work. Over the years, a variety of methods have been developed in the causal inference literature for answering interventional and/or counterfactual queries including non-parametric methods (Shalit et al., 2017; Alaa & Van Der Schaar, 2017; Muandet et al., 2021) and probabilistic modeling methods (Zečević et al., 2021a). More relevant to our approach is a recent series of work, including (Moraffah et al., 2020; Pawlowski et al., 2020; Kocaoglu et al., 2018; Parafita & Vitrià, 2020; Zečević et al., 2021b; Garrido et al., 2021; Karimi et al., 2020; Sánchez-Martin et al., 2022; Khemakhem et al., 2021; Sanchez & Tsaftaris, 2022) that have demonstrated the success of using deep (conditional) generative models for this task.

Karimi et al. (2020) propose an approach for answering interventional queries by fitting a *conditional variational autoencoder* to each conditional in the Markov factorization implied by the causal graph. Also using the ideas of variational inference and normalizing flows, Pawlowski et al. (2020) propose schemes for counterfactual inference.

In Khemakhem et al. (2021), the authors propose an autoregressive normalizing flow for causal discovery and queries, referred to as *CAREFL*. While the authors focus on bivariate graphs between observed and unobserved variables, their approach can be extended to more general DAGs. CAREFL is also applicable even with only knowledge of the causal ordering rather than the full causal graph as we require. However, as also noted by Sánchez-Martin et al. (2022), when the causal graph is present, CAREFL is unable to exploit the absence of edges fully as it reduces a causal graph to its causal ordering (which may not be unique). Using normalizing flows for answering causal queries with causal DAG was also explored by (Balgi et al., 2022), however their approach does not have any theoretical guarantees on their counterfactual estimates, and their experimental evaluation is quite limited.

Sánchez-Martin et al. (2022) propose *VACA*, which uses graph neural networks (GNNs) in the form of a *variational graph autoencoder* to sample from the observational, interventional, and counterfactual distribution. VACA can utilize the inherent graph structure through the GNN, however, suffers in empirical performance (see Section 5). Furthermore, the usage of the GNN leads to undesirable design constraints, e.g., the encoder GNN cannot have hidden layers (Sánchez-Martin et al., 2022).

A very recent work by Javaloy et al. (2023) combines the idea of using autoregressive normalizing flows (as in CAREFL) and modeling the entire causal graph as one model (as in VACA), to reduce possible error propagation in the graph. The authors further generalize the theoretical results from Khemakhem et al. (2021) beyond affine autoregressive normalizing flows and establish a clearer connection to SCMs by providing a more direct way of applying the do-operator. In contrast to our work, while Javaloy et al. (2023) focuses on modeling the whole causal graph as one model. Modeling on a per-node basis has several advantages over a single model in terms of flexibility and computational efficiency because it permits individual node models to be trained in parallel. For example, our DCM approach trains approximately seven times faster than CAREFL and nine times faster than VACA (see Appendix D.1). In practice, it also makes it easier to utilize training data even if some values are missing.

Sanchez & Tsaftaris (2022) use diffusion models for counterfactual estimation, focusing on the bivariate graph case with an image class causing an image. The authors train a diffusion model to generate images and use the abduction-

action-prediction procedure from Pearl et al. (2016) as well as classifier guidance (Dhariwal & Nichol, 2021) to generate counterfactual images. However, this is solely for bivariate models and requires training a separate classifier for intermediate diffusion images, and exhibits poor performance for more complex images e.g., ImageNet (Deng et al., 2009). Our approach distinguishes itself from Sanchez & Tsaftaris (2022) as it can handle general causal graphs (beyond the simpler two node setting) and operates on continuous variables (beyond the simpler case of a discrete label and image). Our experimental evaluations are more general as it also covers interventional queries not addressed by Sanchez & Tsaftaris (2022). Finally, in terms of theoretical contribution, we provide rigorous conditions on the latent model and structural equations under which we can estimate counterfactuals. Even for two variable setting considered by Sanchez & Tsaftaris (2022) such a theoretical understanding was previously missing.

Finally, a diffusion model based approach has also been recently proposed for the different task of causal discovery under additive noise models, where the diffusion model serves as an approximation to the Hessian functions (Sanchez et al., 2022). This raises the interesting question of whether the diffusion models can be adapted for solving the end-to-end problem from discovery to inference.

2 Preliminaries

Notation. To distinguish random variables from their instantiation, we represent the former with capital letters and the latter with the corresponding lowercase letters. To distinguish between the nodes in the causal graph and diffusion random variables, we use subscripts to denote graph nodes. Let $[n] := \{1, \dots, n\}$.

Structural Causal Models. Consider a directed acyclic graph (DAG) $\mathcal{G}$ with nodes $\{1, \dots, K\}$ in a topologically sorted order, where a node i is represented by a (random) variable X_i in some generic space $\mathcal{X}_i \subset \mathbb{R}^{d_i}$. Let pa_i be the parents of node i in $\mathcal{G}$ and let $X_{\mathrm{pa}_i} := \{X_j\}_{j \in \mathrm{pa}_i}$ be the variables of the parents of node i. A structural causal model $\mathcal{M}$ describes the relationship between an observed/endogenous node i and its causal parents. Formally, an SCM $\mathcal{M} := (\mathbf{F}, p(\mathbf{U}))$ determines how a set of K endogenous random variables $\mathbf{X} := \{X_1, \dots, X_K\}$ is generated from a set of exogenous random variables $\mathbf{U} := \{U_1, \dots, U_K\}$ with prior distribution $p(\mathbf{U})$ via a set of structural equations, $\mathbf{F} := (f_1, \dots, f_K)$ where $X_i := f_i(X_{\mathrm{pa}_i}, U_i)$ for $i \in [K]$. Throughout this paper, we assume that the unobserved random variables are jointly independent (Markovian SCM), and the DAG $\mathcal{G}$ is the graph induced by $\mathcal{M}$. Every SCM $\mathcal{M}$ entails a unique joint observational distribution satisfying the causal Markov assumption: $p(\mathbf{X}) = \prod_{i=1}^{K} p(X_i \mid X_{\mathrm{pa}_i})$.

Structural causal models address Pearl's causal hierarchy (or "ladder of causation"), which consists of three "layers" of causal queries in increasing complexity (Pearl, 2009a): observational (or associational), interventional, and counterfactual. As an example, an interventional query can be formulated as "What will be the effect on the population $\mathbf{X}$, if a variable X_i is assigned a fixed value γ_i?" The do-operator $\mathrm{do}(X_i := \gamma_i)$ represents the effect of setting variable X_i to γ_i. Note that our proposed framework allows for more general sets of interventions as well, such as interventions on multiple variables denoted as $\mathrm{do}(X_\mathcal{I} := \gamma)$ (where $\mathcal{I} \subseteq [K]$, $X_\mathcal{I} := (X_i)_{i \in \mathcal{I}}$, $\gamma \in \mathbb{R}^{|\mathcal{I}|}$). An intervention operation, $\mathrm{do}(X_\mathcal{I} := \gamma)$, transforms the original joint distribution into an interventional distribution denoted by $p(\mathbf{X} \mid \mathrm{do}(X_\mathcal{I} := \gamma))$. On the other hand, a counterfactual query can be framed as "What would have been the outcome of a particular factual sample $x^{\mathrm{F}} := (x_1^{\mathrm{F}}, \dots, x_K^{\mathrm{F}})$, if the value of $X_\mathcal{I}$ had been set to γ?". Counterfactual estimation may be performed through the three-step procedure of 1) abduction: estimation of the exogenous noise U, 2) action: intervene $\mathrm{do}(X_\mathcal{I} := \gamma)$, and 3) prediction: estimating x^{CF} using the abducted noise and intervention values.

Diffusion Models. Given data from distribution $X^0 \sim Q$, the objective of diffusion models is to construct an efficiently sampleable distribution approximating Q. Denoising diffusion probabilistic models (DDPMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020) accomplish this by introducing a forward noising process that adds isotropic Gaussian noise at each time step and a learned reverse denoising process. A common representation of diffusion models is a fixed Markov chain that adds Gaussian noise with variances $\beta_1, \dots, \beta_T \in (0, 1)$, generating latent variables $X^1, \dots, X^T$, $q(X^t \mid x^{t-1}) = \mathcal{N}(X^t; \sqrt{1-\beta_t} x^{t-1}, \beta_t I)$ and $q(X^t \mid x^0) = \mathcal{N}(X^t; \sqrt{\alpha_t} x^0, (1-\alpha_t) I)$, where $\alpha_t := \prod_{i=1}^{t} (1-\beta_i)$. Here, $T \in \mathbb{Z}^+$, and $t \in \{0, \dots, T\}$ denotes the time index.

By choosing sufficiently large T and α_t that converge to 0, we have X^T is distributed as an isotropic Gaussian distribution. The learned reverse diffusion process attempts to approximate the intractable $q(X^{t-1} \mid x^t)$ using a neural network and is defined as a Markov chain with Gaussian transitions, $p_\theta(X^{t-1} \mid x^t) = \mathcal{N}(X^{t-1}; \mu_\theta(x^t, t), \Sigma_\theta(x^t, t))$. Rather than predicting μ_θ directly, the network could instead predict the Gaussian noise ε from $x^t = \sqrt{\alpha_t} x^0 + \sqrt{1-\alpha_t}\varepsilon$. Ho et al. (2020) found that modeling ε instead of μ_θ, fixing Σ_θ, and using the following reweighted loss

function

$$\mathbb{E}_{\substack{t\sim\mathrm{Unif}\{[T]\} \\ X^0\sim Q \\ \varepsilon\sim\mathcal{N}(0,I)}}[\|\varepsilon - \varepsilon_\theta(\sqrt{\alpha_t}X^0 + \sqrt{1-\alpha_t}\varepsilon, t)\|^2], \tag{1}$$

works well empirically. We also utilize this loss function in our training.

Song et al. (2021) demonstrate that it is possible to take a pretrained standard denoising diffusion probabilistic model (DDPM) and generalize the generation to non-Markovian processes. In particular, it is possible to use a pretrained DDPM model to obtain a deterministic sample given noise X^T, known as the denoising diffusion implicit model (DDIM), with *reverse implicit diffusion process*

$$X^{t-1} := \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}X^t - \varepsilon_\theta(X^t, t)\left(\sqrt{\alpha_{t-1}(1-\alpha_t)/\alpha_t} - \sqrt{1-\alpha_{t-1}}\right). \tag{2}$$

Note that the X^t here is deterministic. We also use a *forward implicit diffusion process* introduced by Song et al. (2021), derived from rewriting the DDIM process Eq. 2 as an ordinary differential equation (ODE) and considering the Euler method approximation in the forward direction to obtain

$$X^{t+1} := \sqrt{\frac{\alpha_{t+1}}{\alpha_t}}X^t + \varepsilon_\theta(X^t, t)\left(\sqrt{1-\alpha_{t+1}} - \sqrt{\alpha_{t+1}(1-\alpha_t)/\alpha_t}\right). \tag{3}$$

We utilize the DDIM framework in this work, in particular Eqs. 3 and 2 will define the encoding (forward) and decoding (reverse) processes. Note that this ensures deterministic encoding and decoding. This construction produces a unique latent variable per observation, as well as a unique decoding, and also ensures we obtain the same output for repeated counterfactual queries.

3 DCMs: Diffusion-based Causal Models

In this section, we present our DCM approach for modeling the SCMs and to answer causal queries. The DCM approach falls in a general class of techniques that try to model a structural causal model by using an encoder-decoder pair. Consider a data generating process $X = f(X_{pa}, U)$. The goal will to construct an encoding function g and a decoding function h. The encoding function g attempts to represent the information in U: for a pair (X, X_{pa}), $Z := g(X, X_{pa})$ is the latent variable. The decoder takes the input Z and X_{pa} as input to attempt to reconstruct X: $\hat{X} = h(Z, X_{pa})$, where under perfect reconstruction, $\hat{X} = X$. The decoding function h mimics the true structural equation f, although it does not need to be exactly equal. For example, there are infinitely many encodings that satisfy $Z = r(U)$ for all U for an invertible function r.

We first explain the construction and the training process of a DCM, and then explain how the model can be used for answering various causal queries. We start with some notations.

- Define Z_i^t to be the ith endogenous node value at diffusion step t of the *forward* implicit diffusion process (Eq. 3), and let $Z_i := Z_i^T$.
- Define $\hat{X}_i^t$ to be ith endogenous node value at diffusion step t of the *reverse* implicit diffusion process (Eq. 2), and let $\hat{X}_i := \hat{X}_i^0$.

Training a DCM. We train a diffusion model for *each node*, taking denoised parent values as input. The parent values can be interpreted as additional covariates to the model, where one may choose to use classifier free guidance to incorporate the covariates (Ho & Salimans, 2021). Empirically, we find that simply concatenating the covariates results in better performance than classifier free guidance.

We use the ε_θ parametrization for the diffusion model from Ho et al. (2020), representing the diffusion model for node i as $\varepsilon_\theta^i(X, X_{\mathrm{pa}_i}, t)$. The complete training procedure presented in Algorithm 1 is only slightly modified from the usual training procedure, with the additions of the parents as covariates and training a diffusion model for each node. Since the generative models learned for generation of different endogenous nodes do not affect training of each other, these models may be trained in parallel. For each node in the graph, we can train a model in parallel as each diffusion

model only requires the current node and parent values. Our final DCM model is just a combination of these K trained diffusion models $\varepsilon_\theta^1, \ldots, \varepsilon_\theta^K$.

Algorithm 1 DCM Training

Input: Distribution Q, scale factors $\{\alpha_t\}_{t=1}^T$, causal DAG $\mathcal{G}$ with node i represented by X_i

1: **while** not converged **do**
2: Sample $X^0 \sim Q$
3: **for** $i = 1, \ldots, K$ **do**
4: $t \sim \text{Unif}[\{1, \ldots, T\}]$
5: $\varepsilon \sim \mathcal{N}(0, I_{d_i})$ $\{d_i$ is the dimension of $X_i\}$
6: Update parameters of node i's diffusion model ε_θ^i, by minimizing the following loss: $\|\varepsilon - \varepsilon_\theta^i(\sqrt{\alpha_t} X_i^0 + \sqrt{1-\alpha_t}\varepsilon, X_{\text{pa}_i}^0, t)\|_2^2$ (based on Eq. 1)
7: **end for**
8: **end while**

We use all the variables $(X_1, \ldots, X_K)$ for the training procedure, because a priori, we do not assume anything on the possible causal queries, i.e., we allow for all possible target variables, intervened variables, etc. However, if we are only interested in some pre-defined set of queries, then the graph could be reduced accordingly. For example, if we are only interested in counterfactual estimate of a particular node with respect to an intervention of a predecessor, one can simply reduce it to a subgraph containing the target node, the intervened node and a backdoor adjustment set (e.g., the ancestors of the intervened node). This then reduces to only learning a single diffusion model.

One major advantage of our proposed DCM approach is the ability to generalize to larger graphs. Since each diffusion model only uses the parents as input, modeling each node depends only on the incoming degree of the node (the number of causal parents). While the number of diffusion models scales with the number of non-root nodes, each model is generally small in terms of its parameter size and can be trained in parallel. Additionally, we may apply the proposed DCM approach to any setting where diffusion models are applicable: continuous variables, high dimensional settings, categorical data, images, etc.

Encoding and Decoding Steps with DCM. With a DCM, the encoding (resp. decoding) process is identical to the DDIM encoding (resp. decoding) process except we include the parent values as additional covariates. Note that, given the model ϵ_θ, DDIM is a deterministic process as laid out in Eqs. 2 and 3. Let us focus on a node $i \in [K]$ (same process is repeated for each node i). The encoding process takes X_i and its parent values X_{pa_i} as input and maps it to a latent variable Z_i. The decoding process takes Z_i and X_{pa_i} as input to construct $\hat{X}_i$ (an approximation of X_i). Formally, using the forward implicit diffusion process in Eq. 3, given a sample X_i, we encode a unique latent variable $Z_i := Z_i^T$, using the recursive formula

$$Z_i^{t+1} := \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} Z_i^t + \varepsilon_\theta^i(Z_i^t, X_{\text{pa}_i}, t) \left(\sqrt{1-\alpha_{t+1}} - \sqrt{\frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t}} \right), \forall t = 0, .., T-1, \tag{4}$$

where $Z_i^0 := X_i$. The latent variable Z_i acts as a proxy for the exogenous noise U_i. Using the reverse implicit diffusion process from DDIM in Eq. 2, given a latent vector Z_i we obtain a deterministic decoding $\hat{X}_i := \hat{X}_i^0$, using the recursive formula

$$\hat{X}_i^{t-1} := \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \hat{X}_i^t - \varepsilon_\theta^i(\hat{X}_i^t, X_{\text{pa}_i}, t) \left(\sqrt{\frac{\alpha_{t-1}(1-\alpha_t)}{\alpha_t}} - \sqrt{1-\alpha_{t-1}} \right), \text{ for all } t = T, \ldots, 1, \tag{5}$$

where $\hat{X}^T := Z_i$. In the following, we use $\mathsf{Enc}_i(X_i, X_{\text{pa}_i})$ and $\mathsf{Dec}_i(Z_i, X_{\text{pa}_i})$ to denote the encoding and decoding functions for node i defined in Eqns. 4 and 5 respectively. See Algorithms 2 and 3 for detailed pseudocodes.

Answering Causal Queries with a Trained DCM. We now describe how a trained DCM model can be used for (approximately) answering causal queries. Answering observational and interventional queries require sampling from the observational and the interventional distribution respectively. With counterfactuals, a query is at the unit level, where the structural assignments are changed, but the exogenous noise is identical to that of the observed datum.

Algorithm 2 $\mathsf{Enc}_i(X_i, X_{\mathrm{pa}_i})$

Input: X_i, X_{pa_i}

$Z_i^0 \leftarrow X_i$
for $t = 0, \ldots, T-1$ **do**
  $Z_i^{t+1} \leftarrow \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} Z_i^t + \varepsilon_\theta^i(Z_i^t, X_{\mathrm{pa}_i}, t)\left(\sqrt{1-\alpha_{t+1}} - \sqrt{\frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t}}\right)$
end for
Return $Z_i := Z_i^T$

Algorithm 3 $\mathsf{Dec}_i(Z_i, X_{\mathrm{pa}_i})$

Input: Z_i, X_{pa_i}

$\hat{X}^T \leftarrow Z_i$
for $t = T, \ldots, 1$ **do**
  $\hat{X}_i^{t-1} \leftarrow \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \hat{X}_i^t - \varepsilon_\theta^i(\hat{X}_i^t, X_{\mathrm{pa}_i}, t)\left(\sqrt{\frac{\alpha_{t-1}(1-\alpha_t)}{\alpha_t}} - \sqrt{1-\alpha_{t-1}}\right)$
end for
Return $\hat{X}_i := \hat{X}_i^0$

(a) Generating Samples for Observational/Interventional Queries. Samples from a DCM model that approximates the interventional distribution $p(\mathbf{X} \mid \mathrm{do}(X_{\mathcal{I}} := \gamma))$ can be generated as follows. For an intervened node i with intervention γ_i, the sampled value is always the intervention value, therefore we generate $\hat{X}_i := \gamma_i$. For a non-intervened node i, assume by induction we have the generated parent values $\hat{X}_{\mathrm{pa}_i}$. To generate $\hat{X}_i$, we first sample the latent vector $Z_i \sim \mathcal{N}(0, I_{d_i})$ where d_i is the dimension of X_i. Then taking Z_i as the noise for node i, we compute $\hat{X}_i := \mathsf{Dec}_i(Z_i, \hat{X}_{\mathrm{pa}_i})$ as the generated sample value for node i. This value $\hat{X}_i$ is then used as the parent value for the children of node i. Samples from a DCM model that approximates the observational distribution $p(\mathbf{X})$ can be generated by setting $\mathcal{I} = \emptyset$. See Algorithm 4 for the pseudocode.

Algorithm 4 Observational/Interventional Sampling

Input: Intervention set $\mathcal{I}$ with values γ ($I = \emptyset$ for observational sampling)

for $i = 1, \ldots, K$ **do** {in topological order}
  $Z_i \sim \mathcal{N}(0, I_{d_i})$
  if $i \in \mathcal{I}$ **then**
    $\hat{X}_i \leftarrow \gamma_i$
  else
    $\hat{X}_i \leftarrow \mathsf{Dec}_i(Z_i, \hat{X}_{\mathrm{pa}_i})$
  end if
end for
Return $\hat{X} := (\hat{X}_1, \ldots, \hat{X}_K)$

(b) Counterfactual Queries. Consider a factual observation $x^{\mathrm{F}} := (x_1^{\mathrm{F}}, \ldots, x_K^{\mathrm{F}})$ and interventions on a set of nodes $\mathcal{I}$ with values γ. We use a DCM model to construct a counterfactual estimate $\hat{x}^{\mathrm{CF}}$ as follows. The counterfactual estimate only differs from the factual value on intervened nodes or descendants of an intervened node. Similarly to interventional queries, for each intervened node $i \in \mathcal{I}$, $\hat{x}_i^{\mathrm{CF}} := \gamma_i$. For each non-intervened node i that is a descendant of any intervened node, assume by induction that we have the generated counterfactual estimates $\hat{x}_{\mathrm{pa}_i}^{\mathrm{CF}}$. To obtain $\hat{x}_i^{\mathrm{CF}}$, we first define the estimated factual noise as $\hat{z}_i^{\mathrm{F}} := \mathsf{Enc}_i(x_i^{\mathrm{F}}, x_{\mathrm{pa}_i}^{\mathrm{F}})$. Then we generate our counterfactual estimate by using $\hat{z}_i^{\mathrm{F}}$ as the noise for node i, by decoding, $\hat{x}_i^{\mathrm{CF}} := \mathsf{Dec}_i(\hat{z}_i^{\mathrm{F}}, \hat{x}_{\mathrm{pa}_i}^{\mathrm{CF}})$. See Algorithm 5 for the pseudocode.

Note that with x^{F}, we assumed full observability,[1] since because Algorithm 5 produces a counterfactual estimate for each node. However, when intervening on $X_{\mathcal{I}}$ and if the only quantity of interest is counterfactual on some $X^\star$, then

[1]This is a common assumption in literature also made in all the related work e.g., (Sánchez-Martin et al., 2022; Khemakhem et al., 2021; Pawlowski et al., 2020).

Algorithm 5 Counterfactual Estimation

Input: Intervention set $\mathcal{I}$ with values γ, factual sample $x^{\mathrm{F}} := (x_1^{\mathrm{F}}, \ldots, x_K^{\mathrm{F}})$

for $i = 1, \ldots, K$ **do** {in topological order}
  if $i \in \mathcal{I}$ **then**
    $\hat{x}_i^{\mathrm{CF}} \leftarrow \gamma_i$
  else if i is not a descendant of any intervened node in $\mathcal{I}$ **then**
    $\hat{x}_i^{\mathrm{CF}} \leftarrow x_i^{\mathrm{F}}$
  else
    $z_i^{\mathrm{F}} \leftarrow \mathsf{Enc}_i(x_i^{\mathrm{F}}, x_{\mathrm{pa}_i}^{\mathrm{F}})$ {abduction step}
    $\hat{x}_i^{\mathrm{CF}} \leftarrow \mathsf{Dec}_i(z_i^{\mathrm{F}}, \hat{x}_{\mathrm{pa}_i})$ {action and prediction steps}
  end if
end for
Return $\hat{x}^{\mathrm{CF}} := (\hat{x}_1^{\mathrm{CF}}, \ldots, \hat{x}_K^{\mathrm{CF}})$

you only need factual samples from $\{X_i : X_i \text{ is on a path from } X_{\mathcal{I}} \to X^\star\}$ (Saha & Garain, 2022). In practice, this could be further relaxed by imputing for missing data, which is beyond the scope of this work.

4 Bounding Counterfactual Error

We now establish *sufficient* conditions under which the counterfactual estimation error can be bounded. In fact, the results in this section not only hold for diffusion models, but to a more general setting of conditional latent variable models satisfying certain properties. Another feature of this result, is that it also extends, under an additional assumption, to the more challenging higher-dimensional case. All proofs from this section are collected in Appendix A.

We focus on learning a single conditional latent variable model for an endogenous node X_i, given its parents X_{pa_i}, as the models learned for different endogenous nodes do not affect each other. Since the choice of node i plays no role, we drop the subscript i in the following and refer to the node of interest as X, its causal parents as X_{pa}, its corresponding exogenous variables as U, and its structural equation as $X := f(X_{\mathrm{pa}}, U)$. Let the encoding function $g : \mathcal{X} \times \mathcal{X}_{\mathrm{pa}} \to \mathcal{Z}$ and the decoding function $h : \mathcal{Z} \times \mathcal{X}_{\mathrm{pa}} \to \mathcal{X}$, where $\mathcal{Z}$ is the latent space. In the DCM context, the functions g and h correspond to Enc and Dec functions, respectively.

It is well-known that certain counterfactual queries are not identifiable from observational data without making assumptions on the functional relationships, even under causal sufficiency (Pearl, 2009b). Consequently, recent research has been directed towards understanding the conditions under which identifiability results can be obtained (Lu et al., 2020; Nasr-Esfahany & Kiciman, 2023; Nasr-Esfahany et al., 2023). Assumption 2 of our Theorem 1, ensures that the true counterfactual outcome is identifiable, see e.g., (Lu et al., 2020, Theorem 1) or (Nasr-Esfahany & Kiciman, 2023, Theorem 5). In the context of learned structural causal models to determine whether a given counterfactual query can be answered with sufficient accuracy, requires also assumptions on the learned SCM, e.g., encoder and decoder in this case.

Our first result presents sufficient conditions on the latent variable encoding function and the structural equation under which we can recover latent exogenous variable up to a (possibly nonlinear) invertible function. We start with a one-dimensional exogenous noise U and variable $X \in \mathcal{X} \subset \mathbb{R}$. In Section 4.1, we provide a similar theorem for the higher-dimensional case where $X \in \mathbb{R}^m$ for $m \geq 3$ in Theorem 2 with a stronger assumption on the Jacobian of f and g.

Theorem 1. *Assume for $X \in \mathcal{X} \subset \mathbb{R}$ and exogenous noise $U \sim \mathrm{Unif}[0, 1]$, X satisfies the structural equation: $X := f(X_{\mathrm{pa}}, U)$, where $X_{\mathrm{pa}} \in \mathcal{X}_{\mathrm{pa}} \subset \mathbb{R}^d$ are the parents of node X and $U \perp\!\!\!\perp X_{\mathrm{pa}}$. Consider an encoder-decoder model with encoding function $g : \mathcal{X} \times \mathcal{X}_{\mathrm{pa}} \to \mathcal{Z}$ and decoding function $h : \mathcal{Z} \times \mathcal{X}_{\mathrm{pa}} \to \mathcal{X}$, $Z := g(X, X_{\mathrm{pa}})$, $\hat{X} := h(Z, X_{\mathrm{pa}})$. Assume the following conditions:*

1. *The encoding is independent of the parent values, $g(X, X_{\mathrm{pa}}) \perp\!\!\!\perp X_{\mathrm{pa}}$.*
2. *The structural equation f is differentiable and strictly increasing with respect to U.*
3. *The encoding g is invertible and differentiable with respect to X.*

Then, $g(X, X_{\text{pa}}) = \tilde{q}(U)$ *for an invertible function* $\tilde{q}$*.*

Discussion on Assumptions Underlying Theorem 1.

(1) Assumption 1 of independence between the encoding and the parent values may appear strong, but is in fact often valid. For example, in the additive noise setting with $X := f(X_{\text{pa}}) + U$ where X_{pa} and U are independent, if the fitted model $\hat{f} \equiv f$, then the encoder $g(X, X_{\text{pa}}) = X - \hat{f}(X_{\text{pa}}) = U$ and by definition U is independent of X_{pa}.[2] The same assumption also appears in other related results on counterfactual identifiability in bijective SCMs, see, e.g., (Nasr-Esfahany et al., 2023, Theorem 5.3) and proof of Theorem 5 in (Nasr-Esfahany & Kiciman, 2023). We conduct empirical tests to further confirm this assumption by examining the dependence between the parents and encoding values. Our experimental results show that DCMs consistently fail to reject the null hypothesis of independence. This implies that independent encodings can be found in practice. We provide the details of these experiments in Appendix B.[3].

(2) Assumption 2 is always satisfied under the additive noise model (i.e., $X := f(X_{\text{pa}}) + U$) and post non-linear models (Zhang & Hyvarinen, 2012).[4] Again, the recent results about counterfactual identifiability, e.g., (Nasr-Esfahany & Kiciman, 2023, Theorem 5) and (Lu et al., 2020, Theorem 1), also utilize the same assumption. The strictly increasing assumption is somewhat necessary for any identifiability result as it obviates trivial cases, e.g., distinguishing between $X := X_{\text{pa}} + U$ and $X := X_{\text{pa}} - U$.

(3) We may consider transformations of the uniform noise U to obtain other settings, for example additive Gaussian noise. For a continuous random variable U' with invertible CDF F and the structural equation $f(\cdot, F(\cdot))$, we have $U' \overset{d}{=} F^{-1}(U)$ and the results similarly hold.

We now discuss some consequences of Theorem 1 for estimating counterfactual outcomes.

1. Perfect Estimation. Using Theorem 1, we now look at a condition under which the counterfactual estimate produced by the encoder-decoder model matches the true counterfactual outcome.[5]. The idea here been, if no information is lost in the encoding and decoding steps, i.e., $h(g(X, X_{\text{pa}}), X_{\text{pa}}) = X$, and assuming Theorem 1 ($g(X, X_{\text{pa}}) = \tilde{q}(U)$), we have $h(\tilde{q}(U), X_{\text{pa}}) = X = f(X_{\text{pa}}, \tilde{q}^{-1}(U))$. This means that in the abduction step, the encoder-decoder model could recover $\tilde{q}(U)$, but in the prediction step, it first applies the inverse of $\tilde{q}$ to the recovered exogenous variable, and then f. Thus, the counterfactual estimate equals the true counterfactual outcome. We formalize this in Corollary 1.

Corollary 1. *Assume the conditions of Theorem 1. Furthermore, assume the encoder-decoder model pair* (g, h) *satisfies:* $h(g(X, X_{\text{pa}}), X_{\text{pa}}) = X$*. Consider a factual sample pair* $x^{\text{F}} := (x, x_{\text{pa}})$ *where* $x := f(x_{\text{pa}}, u)$ *and an intervention* $\text{do}(X_{\text{pa}} := \gamma)$*. Then, the counterfactual estimate, given by* $h(g(x, x_{\text{pa}}), \gamma)$ *matches the true counterfactual outcome* $x^{\text{CF}} := f(\gamma, u)$*.*

Comparison with Recent Related Work. Recent studies by Nasr-Esfahany & Kiciman (2023) and Nasr-Esfahany et al. (2023) have explored the problem of estimating counterfactual outcomes with learned SCMs. In particular, Nasr-Esfahany & Kiciman (2023, Theorem 5) consider a setting where the SCM $X := f(X_{\text{pa}}, U)$ is learned with a *bijective* (deep conditional generative) model $\hat{f}(X_{\text{pa}}, \hat{U})$. Nasr-Esfahany et al. (2023, Theorem 5.3) considered a closely related problem of learning a ground-truth bijective SCM. The conditions underlying ours and these results are not directly comparable because, unlike our setup, they do not explicitly consider an encoder-decoder model. Our results provide precise conditions on the encoder g and decoder h for recovering the correct counterfactual outcome, and we can extend these results to obtain counterfactual estimation error bounds under relaxed assumptions, a problem that has not been addressed previously.

Counterfactual identifiability in a different context of reinforcement learning was also established by (Lu et al., 2020). Their result relies on incomparable assumptions on state-action pairs. Furthermore, our proof techniques are quite

[2]In general, if we have a good approximation f by some $\hat{f}$, then the encoding $g(X, X_{\text{pa}}) = X - \hat{f}(X_{\text{pa}})$ would be close to U, as also noted by (Hoyer et al., 2008).

[3]To encourage independence, one could also modify the original diffusion model training objective to add a Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2007) regularization term. Our experiments did not show a clear benefit of using this modified objective, and we leave further investigation here for future work.

[4]It is also satisfied by heteroscedastic noise models (HNMs) which are generally defined as $X = f(X_{pa}) + g(X_{pa}) \cdot U$ where the function g is assumed to be strictly positive (see Definition 1 in (Strobl & Lasko, 2023), which makes it compatible with our Assumption 2 of Theorem 1.

[5]Note that identifiability of the counterfactual outcomes, does not require identifiability of the SCM.

different from Lu et al. (2020) who rely on a technique based on analyzing conditional quantile, unlike an algebraic technique employed here.

2. Estimation Error. Another consequence of Theorem 1 is that it can bound the counterfactual error in terms of the reconstruction error of the encoder-decoder model. Informally, the following corollary shows that if the reconstruction $h(g(X, X_{\text{pa}}), X_{\text{pa}})$ is "close" to X (measured under some metric $d(\cdot, \cdot)$), then such encoder-decoder models can provide "good" counterfactual estimates. To the best of our knowledge, this is the first result that establishes a bound on the counterfactual error in relation to the reconstruction error of these encoder-decoder models.

Corollary 2. *Let $\gamma \geq 0$. Assume the conditions of Theorem 1. Furthermore, assume the encoder-decoder model pair (g, h) under some metric d (e.g., $\|\cdot\|_2$), has reconstruction error less than τ: $d(h(g(X, X_{\text{pa}}), X_{\text{pa}}), X) \leq \tau$. Consider a factual sample pair $x^{\text{F}} := (x, x_{\text{pa}})$ where $x := f(x_{\text{pa}}, u)$ and an intervention $\text{do}(X_{\text{pa}} := \gamma)$. Then, the error between the true counterfactual $x^{\text{CF}} := f(\gamma, u)$ and counterfactual estimate given by $h(g(x, x_{\text{pa}}), \gamma)$ is at most τ. In other words, $d(h(g(x, x_{\text{pa}}), \gamma), x^{\text{CF}}) \leq \tau$.*

The above result suggests that the reconstruction error can serve as an estimate for the counterfactual error. While the true value of τ is unknown, we may compute a reasonable bound by computing the reconstruction error over the dataset.

4.1 Extension of Theorem 1 to Higher-Dimensional Setting

In this section, we present an extension of Theorem 1 to a higher dimensional setting and use it to provide counterfactual identifiability and estimation error results.

Theorem 2. *Assume for $X \in \mathcal{X} \subset \mathbb{R}^m$ and continuous exogenous noise $U \sim \text{Unif}[0, 1]^m$ for $m \geq 3$, and X satisfies the structural equation*

$$X = f(X_{\text{pa}}, U) \tag{6}$$

where $X_{\text{pa}} \in \mathcal{X}_{\text{pa}} \subset \mathbb{R}^d$ are the parents of node X and $U \perp\!\!\!\perp X_{\text{pa}}$. Consider an encoder-decoder model with encoding function $g : \mathcal{X} \times \mathcal{X}_{\text{pa}} \to \mathcal{Z}$ and decoding function $h : \mathcal{Z} \times \mathcal{X}_{\text{pa}} \to \mathcal{X}$,

$$Z := g(X, X_{\text{pa}}), \quad \hat{X} := h(Z, X_{\text{pa}}). \tag{7}$$

Assume the following conditions:

1. *The encoding is independent of the parent values, $g(X, X_{\text{pa}}) \perp\!\!\!\perp X_{\text{pa}}$.*
2. *The structural equation f is invertible and differentiable with respect to U, and $Jf_{x_{\text{pa}}}$ is p.d. for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$.*
3. *The encoding g is invertible and differentiable with respect to X, and $Jg_{x_{\text{pa}}}$ is p.d. for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$.*
4. *The encoding $q_{x_{\text{pa}}}(U) := g(f(x_{\text{pa}}, U), x_{\text{pa}})$ satisfies $Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = c(x_{\text{pa}})A$ for all $z \in \mathcal{Z}$ and $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$, where c is a scalar function and A is an orthogonal matrix.*

Then, $g(f(X_{\text{pa}}, U), X_{\text{pa}}) = \tilde{q}(U)$ for an invertible function $\tilde{q}$.

In Corollaries 3 and 4 (Appendix A), we restate Corollaries 1 and 2 to this higher-dimensional setting. The proofs are identical to that of Corollaries 1 and 2, with the only change being that the role of Theorem 1 in those proofs is now replaced by Theorem 2.

On Negative Result of Nasr-Esfahany & Kiciman (2023). Nasr-Esfahany & Kiciman (2023) presented a general counterfactual impossibility identification result under multidimensional exogenous noise. The construction in Nasr-Esfahany & Kiciman (2023) considers two structural equations f, f' that are indistinguishable in distribution. Formally, let $R \in \mathbb{R}^{m \times m}$ be a rotation matrix, and $U \in \mathbb{R}^m$ be a standard (isotropic) Gaussian random vector. Define,

$$f'(X_{\text{pa}}, U) = \begin{cases} f(X_{\text{pa}}, U) & \text{for } X_{\text{pa}} \in A \\ f(X_{\text{pa}}, R \cdot U) & \text{for } X_{\text{pa}} \in B \end{cases}, \tag{8}$$

where the domain $\mathcal{X}_{\text{pa}}$ is split into disjoint A and B. Now, f and f' generate different counterfactual outcomes, for counterfactual queries with evidence in A and intervention in B (or the other way around).

In Theorem 2, we avoid this impossibility result by assuming that we can construct an encoding of a "special" kind captured through our Assumption 4. In particular, consider the encoding $q_{x_{\text{pa}}}$ at a specific parent value x_{pa} as a function of the exogenous noise U. The assumption states that the Jacobian of the encoding is equal to $c(x_{\text{pa}})A$ for a scalar function c and orthogonal matrix A. However, it is important to acknowledge that this assumption is highly restrictive and difficult to verify, not to mention challenging to construct in practice with just observational data. Our intention is that these initial ideas can serve as a starting point for addressing the impossibility result, with the expectation that subsequent results will further refine and expand upon these ideas.

5 Experimental Evaluation

In this section, we evaluate the empirical performance of DCM for answering causal queries on both synthetic and real world data. Additional semi-synthetic experimental evaluations are presented in Appendix D.2. For the semi-synthetic experiments, we leverage the *Sachs* dataset (Sachs et al., 2005).

Diffusion Model Implementation and Training. For our implementation of the ε_θ model in DCM, we use a simple fully connected neural network with three hidden layers of size $[128, 256, 256]$ and SiLU activation (Elfwing et al., 2018). We fit the model using Adam with a learning rate of 1e-4, batch size of 64, and train for 500 epochs. Since the root nodes lack parents, the only form of counterfactual reasoning involves directly intervening on the root node, which may be done trivially. Therefore, for root nodes, we do not train diffusion models and instead sample from the empirical distribution of the training data. Additional details about the diffusion model parameters are in Appendix C.1.

Compared Approaches. For a fair comparison, our evaluation centers on methodologies that allow for both interventional and counterfactual estimation. With this criteria, we primarily compare DCM to two recently proposed state-of-the-art schemes VACA (Sánchez-Martin et al., 2022) and CAREFL (Khemakhem et al., 2021), and a general regression model that assumes an additive noise model which we refer to as ANM.[6] For VACA and CAREFL, we use the code provided by their respective authors. The ANM approach performs model selection over a variety of models, including linear and gradient boosted regressor, and we use the implementation from the popular *DoWhy* causal inference package (Sharma et al., 2019; Blöbaum et al., 2022). Additional details on how ANM answers causal queries are provided in Appendix C.2 and implementation details for VACA, CAREFL, and ANM are in Appendix C.1.

5.1 Synthetic Data Experiments

For generating quantitative results, we use synthetic experiments since we know the exact structural equations, and hence we have access to the ground-truth observational, interventional, and counterfactual distributions. We provide two sets of synthetic experiments, a set of two larger graphs provided here and a set of four smaller graphs provided in Appendix D.3. The two larger graphs, include a *ladder* graph structure (see Figure 9) and a randomly generated graph structure. Both graphs are comprised of 10 nodes of three dimensions each, and the random graph is a randomly sampled directed acyclic graph (see Appendix C.3 for more details). Since each diffusion model only uses the parents as input, modeling each node depends only on the incoming degree of the node (the number of causal parents).

Following Sánchez-Martin et al. (2022), for the observational and interventional distributions, we report the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between the true and estimated distributions. For counterfactual estimation, we report the mean squared error (MSE) of the true and estimated counterfactual values. Again following Sánchez-Martin et al. (2022), we consider two broad classes of structural equations:

1. Additive Noise Model (NLIN): $f_i(X_{\text{pa}_i}, U_i) = f'(X_{\text{pa}_i}) + U_i$. In particular, we will be interested in the case where f_i's are non-linear.
2. Nonadditive Noise Model (NADD): $f_i(X_{\text{pa}_i}, U_i)$ is an arbitrary function of X_{pa_i} and U_i.

To prevent overfitting of hyperparameters, we randomly generate these structural equations for each initialization. Each structural equation is comprised a neural network with a single hidden layer of 16 units and SiLU activation (Elfwing et al., 2018) with random weights from sampled from $[-1, 1]$.

[6] In spite of our best efforts, we were unable to run a proper comparison against the very recent approach proposed by Javaloy et al. (2023) due to challenges in adapting their code into our settings.

SCM		Metric	DCM ($\times 10^{-2}$)	ANM ($\times 10^{-2}$)	VACA ($\times 10^{-2}$)	CAREFL ($\times 10^{-2}$)
Ladder	NLIN	Obs. MMD	**0.44±0.14**	0.63±0.21	2.82±0.83	13.41±1.14
		Int. MMD	**1.63±0.20**	1.80±0.17	4.48±0.78	15.01±1.23
		CF. MSE	**3.42±1.67**	10.65±2.48	41.03±19.00	17.46±6.04
	NADD	Obs. MMD	**0.32±0.11**	0.40±0.16	3.22±1.05	14.60±1.34
		Int. MMD	**1.54±0.17**	1.57±0.15	5.13±1.16	16.87±1.85
		CF. MSE	**4.28±2.39**	10.71±5.47	27.42±12.34	22.26±13.75
Random	NLIN	Obs. MMD	**0.28±0.12**	0.47±0.15	1.82±0.73	12.11±1.21
		Int. MMD	**1.45±0.07**	1.88±0.22	3.52±1.03	14.15±2.34
		CF. MSE	**9.51±13.12**	23.68±28.49	82.10±78.49	52.57±82.03
	NADD	Obs. MMD	**0.19±0.05**	0.31±0.14	2.09±0.60	12.63±1.10
		Int. MMD	**1.42±0.25**	1.73±0.44	4.24±1.40	14.65±1.76
		CF. MSE	**20.13±57.52**	44.76±86.02	124.82±275.09	54.29±83.68

Table 1: Mean±standard deviation of observational, interventional, and counterfactual queries of the ladder and random SCMs in nonlinear and nonadditive settings over 20 random initializations of the model and training data. The values are multiplied by 100 for clarity.

Each simulation generates $n = 5000$ samples as training data. Let $\hat{M}$ be a fitted causal model and $M^\star$ be the true causal model, both capable of generating observational and interventional samples, and answering counterfactual queries. Each pair of graphs and structural equation type is evaluated for 20 different initialization, and we report the mean value. We provide additional details about our observational, interventional, and counterfactual evaluation frameworks in Appendix C.4.

Synthetic Experiments Results. In Table 1, we provide the performance of all evaluated models for observational, interventional, and counterfactual queries, averaged over 20 separate initializations of models and training data, with the lowest value in each row bolded. The values are multiplied by 100 for clarity. We also provide boxplots of the performances in Figure 1 (Appendix D).

We see DCM and ANM are the most competitive approaches, with similar performance on observational and interventional queries. If the ANM is the correctly specified model, then the ANM encoding should be close to the true encoding, assuming the regression model fit the data well. We see this in the nonlinear setting, ANM performs well but struggles to outperform DCM, perhaps due to the complexity of fitting a neural network using classical models. Note thats since observational and interventional queries are inherently easier than counterfactual queries, it is natural that we observe smaller improvements over other baselines.

Our proposed DCM method exhibits superior performance compared to VACA and CAREFL, often by as much as an order of magnitude. The better performance of DCM over CAREFL may be attributed to the fact that DCM uses the causal graph, while CAREFL only relies on a non-unique causal ordering. In the case of VACA, the limited expressiveness of the GNN encoder-decoder might be the reason behind its inferior performance, especially when dealing with multivariable, multidimensional complex structural equations as considered here, a shortcoming that has also been acknowledged by the authors of VACA (Sánchez-Martin et al., 2022). Furthermore, VACA performs *approximate* inference, e.g. when performing a counterfactual query with $\mathrm{do}(X_1 = 2)$, the predicted counterfactual value for X_1 is not exactly 2 due to imperfections in the reconstruction. This design choice may result in downstream compounding errors, possibly explaining discrepancies in performance. To avoid penalizing this feature, all metrics are computed using downstream nodes from intervened nodes. In terms of speed, DCM is much more compute efficient than VACA and CAREFL, typically training 7 times faster, see Appendix D.1 for more details. Lastly, the standard deviation of DCM is small relative to the other models, demonstrating relative consistency, which points to the robustness of our proposed approach.

Algorithm	Median Abs. Error	Mean Abs. Error
DCM	**0.5981** $\pm$ **9.5e-3**	**0.5779** $\pm$ **1.9e-3**
CAREFL	$0.5983 \pm$ 3.0e-2	$0.6004 \pm$ 2.2e-2
ANM	$0.6746 \pm$ 4.8e-8	$0.6498 \pm$ 1.4e-6
Linear SCM	0.6045 ± 0	0.6042 ± 0

Table 2: Performances for interventional predictions on the fMRI dataset, of the form median/mean absolute error$\pm$standard deviation, using the mean over 10 random seeds. We do not include VACA due to implementation difficulties. The results for CAREFL, ANM, and Linear SCM are consistent with those observed by (Khemakhem et al., 2021). We note that the Linear SCM has zero standard deviation, as the ridge regression model does not vary with the random seed.

5.2 Real Data Experiments

We evaluate DCM on interventional real world data by evaluating our model on the electrical stimulation interventional fMRI data from (Thompson et al., 2020), using the experimental setup from (Khemakhem et al., 2021). The fMRI data comprises samples from 14 patients with medically refractory epilepsy, with time series of the Cingulate Gyrus (CG) and Heschl's Gyrus (HG). The assumed underlying causal structure is the bivarate graph CG $\rightarrow$ HG. Our interventional ground truth data comprises an intervened value of CG and an observed sample of HG. We defer the reader to (Thompson et al., 2020; Khemakhem et al., 2021) for a more thorough discussion of the dataset.

In Table 2, we note that the difference in performance is more minor than our synthetic results. We believe this is due to two reasons. Firstly, the data seems inherently close to linear, as exhibited by the relatively similar performance with the standard ridge regression model (Linear SCM). Secondly, we only have a single ground truth interventional value instead of multiple samples from the interventional distribution. As a result, we can only compute the absolute error based on this single value, rather than evaluating the maximum mean discrepancy (MMD) between the true and predicted interventional distributions. Specifically, in the above table, we compute the absolute error between the model prediction and the interventional sample for each of the 14 patients and report the mean/median. The availability of a single interventional introduces a possibly large amount of irreducible error, artificially inflating the error values. For more details on the error inflation, see Appendix C.5.

6 Concluding Remarks

We demonstrate that diffusion models, in particular, the DDIM formulation (which allows for unique encoding and decoding) provide a flexible and practical framework for approximating interventions (do-operator) and counterfactual (abduction-action-prediction) steps. Our approach, DCM, is applicable independent of the DAG structure. We find that empirically DCM outperforms competing methods in all three causal settings, observational, interventional, and counterfactual queries, across various classes of structural equations and graphs.

While not in scope of this paper, the proposed DCM approach can also be naturally extended to any setting where diffusion models are applicable like categorical data, images, etc. For higher dimensional spaces, we believe DCMs should scale nicely, as diffusion models are typically deployed for high-dimensional image settings and exhibit SOTA performance. Furthermore, we may leverage many of the optimization and implementation tricks, as diffusion models are a very active field of research.

The proposed method does come with certain limitations. For example, as with all the previously mentioned related approaches, DCM precludes unobserved confounding. The theoretical analyses require assumptions, not all of which are easy to test. However, our practical results suggest that DCM provides competitive empirical performance, even when some assumptions needed for our theoretical guarantees are violated.

References

Ahmed M Alaa and Mihaela Van Der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *Advances in neural information processing systems*, 30, 2017.

Sourabh Balgi, Jose M Pena, and Adel Daoud. Personalized public policy analysis in social sciences using causal-graphical normalizing flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11810–11818, 2022.

David E Blair. *Inversion theory and conformal mapping*, volume 9. American Mathematical Soc., 2000.

Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A. Mastakouri, and Dominik Janzing. Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models. *arXiv preprint arXiv:2206.06821*, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2017.12.012. URL https://www.sciencedirect.com/science/article/pii/S0893608017302976. Special issue on deep reinforcement learning.

Sergio Garrido, Stanislav Borysov, Jeppe Rich, and Francisco Pereira. Estimating causal effects with the neural autoregressive density estimator. *Journal of Causal Inference*, 9(1):211–218, 2021.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/d5cfead94f5350c12c322b5b664544c1-Paper.pdf.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. Causal normalizing flows: from theory to practice, 2023.

Amir-Hossein Karimi, Julius Von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in neural information processing systems*, 33: 265–277, 2020.

Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3520–3528. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/khemakhem21a.html.

Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Chaochao Lu, Biwei Huang, Ke Wang, José Miguel Hernández-Lobato, Kun Zhang, and Bernhard Schölkopf. Sample-efficient reinforcement learning via counterfactual-based data augmentation. *arXiv preprint arXiv:2012.09092*, 2020.

Raha Moraffah, Bahman Moraffah, Mansooreh Karami, Adrienne Raglin, and Huan Liu. Can: A causal adversarial network for learning observational and interventional distributions. *arXiv preprint arXiv:2008.11376*, 2020.

Krikamol Muandet, Motonobu Kanagawa, Sorawit Saengkyongam, and Sanparith Marukatat. Counterfactual mean embeddings. *J. Mach. Learn. Res.*, 22:162–1, 2021.

Arash Nasr-Esfahany and Emre Kiciman. Counterfactual (non-) identifiability of learned structural causal models. *arXiv preprint arXiv:2301.09031*, 2023.

Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual identifiability of bijective causal models. *arXiv preprint arXiv:2302.02228*, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Álvaro Parafita and Jordi Vitrià. Causal inference with deep causal graphs. *arXiv preprint arXiv:2006.08380*, 2020.

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. ISBN 9781119186847. URL https://books.google.com/books?id=L3G-CgAAQBAJ.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146, 2009a. doi: 10.1214/09-SS057. URL https://doi.org/10.1214/09-SS057.

Judea Pearl. Causal inference in statistics: An overview, 2009b.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. doi: 10.1126/science.1105809. URL https://www.science.org/doi/abs/10.1126/science.1105809.

Saptarshi Saha and Utpal Garain. On noise abduction for answering counterfactual queries: A practical outlook. *Transactions on Machine Learning Research*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487.

Pedro Sanchez and Sotirios A. Tsaftaris. Diffusion causal models for counterfactual estimation. In *CLeaR*, 2022.

Pedro Sanchez, Xiao Liu, Alison Q O'Neil, and Sotirios A Tsaftaris. Diffusion models for causal discovery via topological ordering. In *The Eleventh International Conference on Learning Representations*, 2022.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

Amit Sharma, Emre Kiciman, et al. DoWhy: A Python package for causal inference. https://github.com/microsoft/dowhy, 2019.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Eric V Strobl and Thomas A Lasko. Identifying patient-specific root causes with the heteroscedastic noise model. *Journal of Computational Science*, 72:102099, 2023.

Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Designing variational graph autoencoders for causal queries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):8159–8168, Jun. 2022. doi: 10.1609/aaai.v36i7.20789. URL https://ojs.aaai.org/index.php/AAAI/article/view/20789.

W. H. Thompson, R. Nair, H. Oya, O. Esteban, J. M. Shine, C. I. Petkov, R. A. Poldrack, M. Howard, and R. Adolphs. A data resource from concurrent intracranial stimulation and functional MRI of the human brain. *Scientific Data*, 7(1):258, August 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-00595-y. URL https://doi.org/10.1038/s41597-020-00595-y.

Matej Zečević, Devendra Dhami, Athresh Karanam, Sriraam Natarajan, and Kristian Kersting. Interventional sum-product networks: Causal inference with tractable probabilistic models. *Advances in Neural Information Processing Systems*, 34:15019–15031, 2021a.

Matej Zečević, Devendra Singh Dhami, Petar Veličković, and Kristian Kersting. Relating graph neural networks to structural causal models. *arXiv preprint arXiv:2109.04173*, 2021b.

Kun Zhang and Aapo Hyvarinen. On the identifiability of the post-nonlinear causal model. *arXiv preprint arXiv:1205.2599*, 2012.

A Missing Details from Section 4

Notation. For two sets $\mathcal{X}, \mathcal{Y}$ a map $f : \mathcal{X} \mapsto \mathcal{Y}$, and a set $S \subset \mathcal{X}$, we define $f(S) = \{f(x) : x \in S\}$. For $x \in \mathcal{X}$, we define $x + S = \{x + x' : x' \in \mathcal{X}\}$. For a random variable X, define $p_X(x)$ as the probability density function (PDF) at x. We use p.d. to denote positive definite matrices and $Jf|_x$ to denote the Jacobian of f evaluated at x. For a function with two inputs $f(\cdot, \cdot)$, we define $f_x(Y) := f(x, Y)$ and $f_y(X) := f(X, y)$.

Lemma 1. *For $\mathcal{U}, \mathcal{Z} \subset \mathbb{R}$, consider a family of invertible functions $q_{x_{\text{pa}}} : \mathcal{U} \to \mathcal{Z}$ for $x_{\text{pa}} \in \mathcal{X}_{\text{pa}} \subset \mathbb{R}^d$, then $\frac{dq_{x_{\text{pa}}}}{du}(q^{-1}_{x_{\text{pa}}}(z)) = c(z)$ for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$ if and only if $q_{x_{\text{pa}}}$ can be expressed as*

$$q_{x_{\text{pa}}}(u) = q(u + r(x_{\text{pa}}))$$

for some function r and invertible q.

Proof. First for the reverse direction, we may assume $q_{x_{\text{pa}}}(u) = q(u + r(x_{\text{pa}}))$. Then

$$\frac{dq_{x_{\text{pa}}}}{du}(u) = \frac{dq}{du}(u + r(x_{\text{pa}})).$$

Now plugging in $u = q_{x_{\text{pa}}}^{-1}(z) = q^{-1}(z) - r(x_{\text{pa}})$,

$$\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z)) = \frac{dq}{du}(q^{-1}(z) - r(x_{\text{pa}}) + r(x_{\text{pa}})) = \frac{dq}{du}(q^{-1}(z)) = c(z).$$

Therefore $\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z))$ is a function of z.

For the forward direction, assume $\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z)) = c(z)$. Define $s_{x_{\text{pa}}} : \mathcal{Z} \to \mathcal{U}$ to be the inverse of $q_{x_{\text{pa}}}$. By the inverse function theorem and by assumption.

$$\frac{ds_{x_{\text{pa}}}}{dz}(z) = \frac{dq_{x_{\text{pa}}}^{-1}}{dz}(z) = \frac{1}{\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z))} = \frac{1}{c(z)}$$

for all x_{pa}. Since the derivatives of $s_{x_{\text{pa}}}$ are equal for all x_{pa}, by the mean value theorem, all $s_{x_{\text{pa}}}$ are additive shifts of each other. Without loss of generality, we may consider an arbitrary fixed $x_{\text{pa}_0} \in \mathcal{X}_{\text{pa}}$ and reparametrize $s_{x_{\text{pa}}}$ as

$$s_{x_{\text{pa}}}(z) = s_{x_{\text{pa}_0}}(z) - r(x_{\text{pa}}).$$

Let $u = s_{x_{\text{pa}}}(z)$. Then we have

$$s_{x_{\text{pa}_0}}(z) = u + r(x_{\text{pa}})$$
$$q_{x_{\text{pa}}}(u) = z = q_{x_{\text{pa}_0}}(u + r(x_{\text{pa}})),$$

and we have the desired representation by choosing $q = q_{x_{\text{pa}_0}}$. □

Theorem 1. *Assume for $X \in \mathcal{X} \subset \mathbb{R}$ and exogenous noise $U \sim \text{Unif}[0,1]$, X satisfies the structural equation: $X := f(X_{\text{pa}}, U)$, where $X_{\text{pa}} \in \mathcal{X}_{\text{pa}} \subset \mathbb{R}^d$ are the parents of node X and $U \perp\!\!\!\perp X_{\text{pa}}$. Consider an encoder-decoder model with encoding function $g : \mathcal{X} \times \mathcal{X}_{\text{pa}} \to \mathcal{Z}$ and decoding function $h : \mathcal{Z} \times \mathcal{X}_{\text{pa}} \to \mathcal{X}$, $Z := g(X, X_{\text{pa}})$, $\hat{X} := h(Z, X_{\text{pa}})$. Assume the following conditions:*

1. *The encoding is independent of the parent values, $g(X, X_{\text{pa}}) \perp\!\!\!\perp X_{\text{pa}}$.*
2. *The structural equation f is differentiable and strictly increasing with respect to U.*
3. *The encoding g is invertible and differentiable with respect to X.*

Then, $g(X, X_{\text{pa}}) = \tilde{q}(U)$ for an invertible function $\tilde{q}$.

Proof. First, we show that $g(X, X_{\text{pa}}) = g(f(X_{\text{pa}}, U), X_{\text{pa}})$ is solely a function of U.

Since continuity and invertibility imply strict monotonicity, without loss of generality, assume g is an strictly increasing function (if not, we may replace g with $-g$ and use $h(-Z, X_{\text{pa}})$). By properties of the composition of functions, $q_{x_{\text{pa}}}(U) := g(f(x_{\text{pa}}, U), x_{\text{pa}})$ is also differentiable and strictly increasing with respect to U. Also, because of strict monotonicity it is also invertible.

By the assumption that the encoding Z is independent of X_{pa},

$$Z = q_{x_{\text{pa}}}(U) \perp\!\!\!\perp X_{\text{pa}}. \tag{9}$$

Therefore the conditional distribution of Z does not depend on X_{pa}. Using the assumption that $U \perp\!\!\!\perp X_{\text{pa}}$, for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$ and z in the support of Z, by the change of density formula,

$$p_Z(z) = \frac{p_U(q_{x_{\text{pa}}}^{-1}(z))}{\left|\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z))\right|} = \frac{\mathbb{1}\{q_{x_{\text{pa}}}^{-1}(z) \in [0,1]\}}{\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z))} = c_1(z). \tag{10}$$

The numerator follows from the fact that the noise is uniformly distributed. The term $\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z))$ is nonnegative since $q_{x_{\text{pa}}}$ is increasing. Furthermore, since $p_Z(z) > 0$, the numerator in Eq. 10 is always equal to 1 and the

denominator must not depend on X_{pa},

$$\frac{dq_{x_{\text{pa}}}}{du}(q_{x_{\text{pa}}}^{-1}(z)) = c_2(z)$$

for some function c_2. From Lemma 1 (by replacing a by x_{pa}), we may express

$$q_{x_{\text{pa}}}(u) = q(u + r(x_{\text{pa}})) \tag{11}$$

for an invertible function q.

Next, since $Z \perp\!\!\!\perp X_{\text{pa}}$, the support of Z does not depend on X_{pa}, equivalently the ranges of q_{x_1} and q_{x_2} are equal for all $x_1, x_2 \in \mathcal{X}_{\text{pa}}$,

$$q_{x_1}([0,1]) = q_{x_2}([0,1]). \tag{12}$$

Applying Eq. 11 and the invertibility of q,

$$\begin{aligned} q([0,1] + r(x_1)) &= q([0,1] + r(x_2)) \\ [0,1] + r(x_1) &= [0,1] + r(x_2) \\ [r(x_1), r(x_1)+1] &= [r(x_2), r(x_2)+1] \end{aligned}$$

Since this holds for all $x_1, x_2 \in \mathcal{X}_{\text{pa}}$, we have $r(x_{\text{pa}})$ is a constant function, or $r(x_{\text{pa}}) \equiv r$. Thus let $\tilde{q}$ be $\tilde{q}(u) = q(u + r) = q_{x_{\text{pa}}}(u)$, which is solely a function of U for all x_{pa}. For all x_{pa},

$$g(f(x_{\text{pa}}, U), x_{\text{pa}})) = q_{x_{\text{pa}}}(U) = \tilde{q}(U) \implies g(f(X_{\text{pa}}, U), X_{\text{pa}}) = \tilde{q}(U), \tag{13}$$

for an invertible function $\tilde{q}$. This completes the proof. □

Corollary 1. *Assume the conditions of Theorem 1. Furthermore, assume the encoder-decoder model pair (g, h) satisfies: $h(g(X, X_{\text{pa}}), X_{\text{pa}}) = X$. Consider a factual sample pair $x^{\text{F}} := (x, x_{\text{pa}})$ where $x := f(x_{\text{pa}}, u)$ and an intervention $\text{do}(X_{\text{pa}} := \gamma)$. Then, the counterfactual estimate, given by $h(g(x, x_{\text{pa}}), \gamma)$ matches the true counterfactual outcome $x^{\text{CF}} := f(\gamma, u)$.*

Proof. For the intervention $\text{do}(X_{\text{pa}} := \gamma)$, the true counterfactual outcome is $x^{\text{CF}} := f(\gamma, u)$. By assumption, $h(g(x^{\text{CF}}, x_{\text{pa}}), x_{\text{pa}}) = x^{\text{CF}}$. Now since Eq. 13 holds true for all X_{pa} and U, it also holds for the factual and counterfactual samples. We have,

$$g(x, x_{\text{pa}}) = g(f(x_{\text{pa}}, u), x_{\text{pa}}) = \tilde{q}(u) = g(f(\gamma, u), \gamma) = g(x^{\text{CF}}, \gamma).$$

Therefore, the counterfactual estimate produced by the encoder-decoder model

$$h(g(x, x_{\text{pa}}), \gamma) = h(g(x^{\text{CF}}, \gamma), \gamma) = x^{\text{CF}}.$$

This completes the proof. □

Corollary 2. *Let $\gamma \geq 0$. Assume the conditions of Theorem 1. Furthermore, assume the encoder-decoder model pair (g, h) under some metric d (e.g., $\|\cdot\|_2$), has reconstruction error less than τ: $d(h(g(X, X_{\text{pa}}), X_{\text{pa}}), X) \leq \tau$. Consider a factual sample pair $x^{\text{F}} := (x, x_{\text{pa}})$ where $x := f(x_{\text{pa}}, u)$ and an intervention $\text{do}(X_{\text{pa}} := \gamma)$. Then, the error between the true counterfactual $x^{\text{CF}} := f(\gamma, u)$ and counterfactual estimate given by $h(g(x, x_{\text{pa}}), \gamma)$ is at most τ. In other words, $d(h(g(x, x_{\text{pa}}), \gamma), x^{\text{CF}}) \leq \tau$.*

Proof. For the intervention $\text{do}(X_{\text{pa}} := \gamma)$, the counterfactual outcome is $x^{\text{CF}} := f(\gamma, u)$. Since Eq. 13 holds true for all X_{pa} and U, it also holds for the factual and counterfactual samples. We have,

$$g(x, x_{\text{pa}}) = g(f(x_{\text{pa}}, u), x_{\text{pa}}) = \tilde{q}(u) = g(f(\gamma, u), \gamma) = g(x^{\text{CF}}, \gamma). \tag{14}$$

By the assumption on the reconstruction error of the encoder-decoder and 14,

$$d(h(g(x^{\mathrm{CF}}, \gamma), \gamma), x^{\mathrm{CF}}) \leq \tau \tag{15}$$

$$d(h(g(x, x_{\mathrm{pa}}), \gamma), x^{\mathrm{CF}}) \leq \tau. \tag{16}$$

□

We now discuss a lemma that is an extension of Lemma 1 to higher dimensions.

Lemma 2. *For $\mathcal{U}, \mathcal{Z} \subset \mathbb{R}^m$, consider a family of invertible functions $q_{x_{\mathrm{pa}}} : \mathcal{U} \to \mathcal{Z}$ for $x_{\mathrm{pa}} \in \mathcal{X}_{\mathrm{pa}} \subset \mathbb{R}^d$, if $Jq_{x_{\mathrm{pa}}}(q_{x_{\mathrm{pa}}}^{-1}(z)) = cA$ for all $x_{\mathrm{pa}} \in \mathcal{X}_{\mathrm{pa}}$ then $q_{x_{\mathrm{pa}}}$ can be expressed as*

$$q_{x_{\mathrm{pa}}}(u) = q(u + r(x_{\mathrm{pa}}))$$

for some function r and invertible q.

Proof. Assume $Jq_{x_{\mathrm{pa}}}(q_{x_{\mathrm{pa}}}^{-1}(z)) = cA$. By the inverse function theorem,

$$Jq_{x_{\mathrm{pa}}}^{-1}|_z = \left(Jq_{x_{\mathrm{pa}}}|_{q_{x_{\mathrm{pa}}}^{-1}(z)}\right)^{-1}. \tag{17}$$

Define $s_{x_{\mathrm{pa}}} : \mathcal{Z} \to \mathcal{U}$ to be the inverse of $q_{x_{\mathrm{pa}}}$. By assumption

$$\det Jq_{x_{\mathrm{pa}}}|_{q_{x_{\mathrm{pa}}}^{-1}(z)} = \det Jq_{x_{\mathrm{pa}}}^{-1}|_z = \det Js_{x_{\mathrm{pa}}}|_z = cA$$

for all x_{pa} and a constant c and orthgonal matrix A. Since the Jacobian of $s_{x_{\mathrm{pa}}}$ is a scaled orthogonal matrix, $s_{x_{\mathrm{pa}}}$ is a conformal function. Therefore by Liouville's theorem, $s_{x_{\mathrm{pa}}}$ is a Möbius function (Blair, 2000), which implies that

$$s_{x_{\mathrm{pa}}}(z) = b_{x_{\mathrm{pa}}} + \alpha_{x_{\mathrm{pa}}} A_{x_{\mathrm{pa}}}(z - a_{x_{\mathrm{pa}}})/\|z - a_{x_{\mathrm{pa}}}\|^{\varepsilon}, \tag{18}$$

where $A_{x_{\mathrm{pa}}}$ is an orthogonal matrix, $\varepsilon \in \{0, 2\}$, $a_{x_{\mathrm{pa}}} \in \mathbb{R}^m$, and $\alpha_{x_{\mathrm{pa}}} \in \mathbb{R}$. The Jacobian of $s_{x_{\mathrm{pa}}}$ is equal to cA by assumption

$$Js_{x_{\mathrm{pa}}}\big|_z = \frac{\alpha_{x_{\mathrm{pa}}} A_{x_{\mathrm{pa}}}}{\|z - a_{x_{\mathrm{pa}}}\|^{\varepsilon}} \left(I - \varepsilon \frac{(z - a_{x_{\mathrm{pa}}})(z - a_{x_{\mathrm{pa}}})^T}{\|z - a_{x_{\mathrm{pa}}}\|^2}\right) = cA.$$

This imposes constraints on variables α, a, and ε. Choose z such that $z - a_{x_{\mathrm{pa}}} = kv$ for a unit vector v and multiply by $A_{x_{\mathrm{pa}}}^{-1}$,

$$cAA_{x_{\mathrm{pa}}}^{-1} = \frac{\alpha_{x_{\mathrm{pa}}}}{\|kv\|^{\varepsilon}} \left(I - \varepsilon \frac{k^2 vv^T}{k^2 \|v\|^2}\right)$$

$$I = \varepsilon vv^T + \left(\frac{ck^{\varepsilon}}{\alpha_{x_{\mathrm{pa}}}}\right) AA_{x_{\mathrm{pa}}}^{-1}.$$

If $\varepsilon = 2$, choosing different values of k, implying different values of z, results in varying values of on the right hand side, which should be the constant identity matrix. Therefore we must have $\varepsilon = 0$. This also implies that $\alpha_{x_{\mathrm{pa}}} = c$ and $A = A_{x_{\mathrm{pa}}}$. This gives the further parametrization

$$s_{x_{\mathrm{pa}}}(z) = b_{x_{\mathrm{pa}}} - cA(z - a_{x_{\mathrm{pa}}}) = b'(x_{\mathrm{pa}}) + cAz$$

where $b'(x_{\mathrm{pa}}) = b_{x_{\mathrm{pa}}} - cAa_{x_{\mathrm{pa}}}$.

Without of loss of generality, we may consider an arbitrary fixed $x_{\mathrm{pa}_0} \in \mathcal{X}_{\mathrm{pa}}$,

$$s_{x_{\mathrm{pa}_0}}(z) - s_{x_{\mathrm{pa}}}(z) = r(x_{\mathrm{pa}}) \coloneqq b'(x_{\mathrm{pa}_0}) - b'(x_{\mathrm{pa}}).$$

Let $u = s_{x_{\text{pa}}}(z)$. Then we have

$$s_{x_{\text{pa}_0}}(z) = u + r(x_{\text{pa}})$$
$$q_{x_{\text{pa}}}(u) = z = q_{x_{\text{pa}_0}}(u + r(x_{\text{pa}})),$$

and we have the desired representation by choosing $q = q_{x_{\text{pa}_0}}$. □

Theorem 2. *Assume for $X \in \mathcal{X} \subset \mathbb{R}^m$ and continuous exogenous noise $U \sim \text{Unif}[0,1]^m$ for $m \geq 3$, and X satisfies the structural equation*

$$X = f(X_{\text{pa}}, U) \tag{6}$$

where $X_{\text{pa}} \in \mathcal{X}_{\text{pa}} \subset \mathbb{R}^d$ are the parents of node X and $U \perp\!\!\!\perp X_{\text{pa}}$. Consider an encoder-decoder model with encoding function $g : \mathcal{X} \times \mathcal{X}_{\text{pa}} \to \mathcal{Z}$ and decoding function $h : \mathcal{Z} \times \mathcal{X}_{\text{pa}} \to \mathcal{X}$,

$$Z := g(X, X_{\text{pa}}), \quad \hat{X} := h(Z, X_{\text{pa}}). \tag{7}$$

Assume the following conditions:

1. *The encoding is independent of the parent values, $g(X, X_{\text{pa}}) \perp\!\!\!\perp X_{\text{pa}}$.*
2. *The structural equation f is invertible and differentiable with respect to U, and $Jf_{x_{\text{pa}}}$ is p.d. for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$.*
3. *The encoding g is invertible and differentiable with respect to X, and $Jg_{x_{\text{pa}}}$ is p.d. for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$.*
4. *The encoding $q_{x_{\text{pa}}}(U) := g(f(x_{\text{pa}}, U), x_{\text{pa}})$ satisfies $Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = c(x_{\text{pa}})A$ for all $z \in \mathcal{Z}$ and $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$, where c is a scalar function and A is an orthogonal matrix.*

Then, $g(f(X_{\text{pa}}, U), X_{\text{pa}}) = \tilde{q}(U)$ for an invertible function $\tilde{q}$.

Proof. We may show that $g(X, X_{\text{pa}}) = g(f(X_{\text{pa}}, U), X_{\text{pa}})$ is solely a function of U.

By properties of composition of functions, $q_{x_{\text{pa}}}(U) := g(f(x_{\text{pa}}, U), x_{\text{pa}})$ is also invertible, differentiable. Since $Jf_{x_{\text{pa}}}$ and $Jg_{x_{\text{pa}}}$ are p.d. and $Jq_{x_{\text{pa}}} = Jf_{x_{\text{pa}}} Jg_{x_{\text{pa}}}$, then $Jq_{x_{\text{pa}}}$ is p.d. for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$ as well.

By the assumption that the encoding Z is independent of X_{pa},

$$Z = q_{X_{\text{pa}}}(U) \perp\!\!\!\perp X_{\text{pa}}. \tag{19}$$

Therefore the conditional distribution of Z does not depend on X_{pa}. Using the assumption that $U \perp\!\!\!\perp X_{\text{pa}}$, for all $x_{\text{pa}} \in \mathcal{X}_{\text{pa}}$ and z in the support of Z, by the change of density formula,

$$p_Z(z) = \frac{p_U(q^{-1}_{x_{\text{pa}}}(z))}{\left|\det Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)}\right|} = \frac{2^{-m}\mathbb{1}\{q^{-1}_{x_{\text{pa}}}(z) \in [0,1]^m\}}{\det Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)}} = c_1(z). \tag{20}$$

The numerator follows from the fact that the noise is uniformly distributed. The determinant of the Jacobian term $Jq_{x_{\text{pa}}}(q^{-1}_{x_{\text{pa}}}(z))$ is nonnegative since $Jq_{x_{\text{pa}}}$ is p.d. Furthermore, since $p_Z(z) > 0$, the numerator in Eq. 20 is always equal to 2^{-m} and the denominator must not depend on X_{pa},

$$\det Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = c_2(z) \tag{21}$$

for some function c_2. From our assumption, $Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = c(x_{\text{pa}})A$ for an orthogonal matrix A for all z. Applying this to Eq. 21,

$$\det Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = \det c(x_{\text{pa}})A = c(x_{\text{pa}}) = c_2(z),$$

which implies $c(x_{\text{pa}}) \equiv c$ is a constant function, or $Jq_{x_{\text{pa}}}|_{q^{-1}_{x_{\text{pa}}}(z)} = cA$. By Lemma 2, we may express $q_{x_{\text{pa}}}(u)$ as

$$q_{x_{\text{pa}}}(u) = q(u + r(x_{\text{pa}})) \tag{22}$$

for an invertible function q.

Next, since $Z \perp\!\!\!\perp X_{\text{pa}}$, the support of Z does not depend on X_{pa}, equivalently the ranges of $q_{x_{\text{pa}_1}}$ and $q_{x_{\text{pa}_2}}$ are equal for all $x_1, x_2 \in \mathcal{X}_{\text{pa}}$,

$$q_{x_1}([0,1]^m) = q_{x_2}([0,1]^m). \tag{23}$$

Applying Eq. 22 and the invertibility of q,

$$\begin{aligned} q([0,1]^m + r(x_1)) &= q([0,1]^m + r(x_2)) \\ [0,1]^m + r(x_1) &= [0,1]^m + r(x_2) \\ [r(x_1), r(x_1)+1]^m &= [r(x_2), r(x_2)+1]^m. \end{aligned}$$

Since this holds for all $x_1, x_2 \in \mathcal{X}_{\text{pa}}$, we have $r(x)$ is a constant, or $r(x) \equiv r$. Thus let $\tilde{q}$ be $\tilde{q}(u) = q(u+r) = q_{x_{\text{pa}}}(u)$, which is solely a function of U for all x_{pa}. For all x_{pa},

$$g(f(x_{\text{pa}}, U, x_{\text{pa}})) = q_{x_{\text{pa}}}(U) = \tilde{q}(U) \implies g(f(X_{\text{pa}}, U), X_{\text{pa}}) = \tilde{q}(U). \tag{24}$$

This completes the proof. □

Corollary 3. *Assume the conditions of Theorem 2. Furthermore, assume the encoder-decoder model pair (g, h) satisfies*

$$h(g(X, X_{\text{pa}}), X_{\text{pa}}) = X. \tag{25}$$

Consider a factual sample pair (x, x_{pa}) where $x := f(x_{\text{pa}}, u)$ and an intervention $\text{do}(X_{\text{pa}} := \gamma)$. Then, the given by $h(g(x, x_{\text{pa}}), \gamma)$ matches the true counterfactual outcome $x^{\text{CF}} := f(\gamma, u)$.

Corollary 4. *Let $\gamma \geq 0$. Assume the conditions of Theorem 2. Furthermore, assume the encoder-decoder model pair (g, h) under some metric d (e.g., $\|\cdot\|_2$), has reconstruction error less than τ,*

$$d(h(g(X, X_{\text{pa}}), X_{\text{pa}}), X) \leq \tau. \tag{26}$$

Consider a factual sample pair (x, x_{pa}) where $x := f(x_{\text{pa}}, u)$ and an intervention $\text{do}(X_{\text{pa}} := \gamma)$. Then, the error between the true counterfactual $x^{\text{CF}} := f(\gamma, u)$ and counterfactual estimate $h(g(x, x_{\text{pa}}), \gamma)$ is at most τ, i.e., $d(h(g(x, x_{\text{pa}}), \gamma), x^{\text{CF}}) \leq \tau$.

B Testing Independence between Parents and Encodings

We empirically evaluate the dependence between the encoding and parent values. We consider a bivariate nonlinear SCM $X_1 \to X_2$ where $X_2 = f(X_1, U_2) = X_1^2 + U_2$ and X_1 and U_2 are independently sampled from a standard normal distribution. We evaluate the HSIC between X_1 and the encoding of X_2. We fit our model on $n = 5000$ samples and evaluate the HSIC score on 1000 test samples from the same distribution. We compute a p-value using a kernel based independence test and compare our performance to ANM, a correctly specified model in this setting (Gretton et al., 2007). We perform this experiment 100 times. Given true independence, we expect the p-values to follow a uniform distribution. In the table below, we show some summary statistics of the p-values from the 100 trials, with the last row representing the expected values with true uniform p-values (which happens under the null hypothesis).

We provide p-values from the correctly specified ANM approach which are close to a uniform distribution, demonstrating that it is possible to have encodings that are close to independent. Although the p-values produced by our DCM approach are not completely uniform, the encodings do not to consistently reject the null hypothesis of independence.

	Mean	Std. Dev	10% Quantile	90% Quantile	Min	Max
DCM	0.196	0.207	0.004	0.515	6e-6	0.947
ANM	0.419	0.255	0.092	0.774	3e-5	0.894
True Uniform (null)	0.500	0.288	0.100	0.900	1e-2	0.990

Table 3: Table of p-value descriptions.

These results demonstrate that it is empirically possible to obtain encodings independent of the parent variables. We further note that the ANM is correctly specified in this setting and DCM is relatively competitive despite being far more general.

C Missing Experimental Details

In this section, we provide missing details from Section 5.

C.1 Model Hyperparameters

For all experiments in our evaluation, we hold the model hyperparameters constant. For DCM, we use $T = 100$ total time steps with a linear β_t schedule interpolating between 1e-4 and 0.1, or $\beta_t = \left(0.1 - 10^{-4}\right) \frac{t-1}{T-1} + 10^{-4}$ for $t \in [T]$. To incorporate the parents' values and time step t, we simply concatenate the parent values and t/T as input to the ε_θ model. We found that using the popular cosine schedule (Nichol & Dhariwal, 2021) resulted in worse empirical performance, as well as using a positional encoding for the time t. We believe the drop in performance from the positional encoding is due to the low dimensionality of the problem since the dimension of the positional encoder would dominate the dimension of the other inputs.

We also evaluated using classifier-free guidance (CFG) (Ho & Salimans, 2021) to improve the reliance on the parent values, however, we found this also decreased performance. We provide a plausible explanation that can be explained through Theorem 1. With Theorem 1, we would like our encoding $g(Y, X)$ to be independent of X, however using a CFG encoding $(1 + w)g(Y, X) - wg(Y, 0)$ would only serve to increase the dependence of $g(Y, X)$ on X, which is counterproductive to our objective.

For VACA, we use the default implementation[7], training for 500 epochs, with a learning rate of 0.005, and the encoder and decoder have hidden dimensions of size $[8, 8]$ and $[16]$ respectively, a latent vector dimension of 4, and a parent dropout rate of 0.2.

For CAREFL, we also use the default implementation[8] with the neural spline autoregressive flows (Durkan et al., 2019), training for 500 epochs with a learning rate of 0.005, four flows, and ten hidden units.

For ANM, we also use the default implementation to select a regression model. Given a set of fitted regression models, the ANM chooses the model with the lowest root mean squared error averaged over splits of the data. The ANM considers the following regressor models: linear, ridge, LASSO, elastic net, random forest, histogram gradient boosting, support vector, extra trees, k-NN, and AdaBoost.

C.2 Details about the Additive Noise Model (ANM)

For a given node X_i with parents X_{pa_i}, consider fitting a regression model $\hat{f}_i$ where $\hat{f}_i(X_{\mathrm{pa}_i}) \approx X_i$. Using this regression model and the training dataset is sufficient for generating samples from the observational and interventional distribution, as well as computing counterfactuals.

Observational/Interventional Samples. Samples are constructed in topological order. For intervened nodes, the sampled value is always the intervened value. Non-intervened root nodes in the SCM are sampled from the empirical distribution of the training set. A new sample for X_i is generated by sampling the parent value X_{pa_i} inductively and sampling $\hat{U}$ from the empirical residual distribution, and outputting $\hat{X}_i := \hat{f}_i(X_{\mathrm{pa}_i}) + \hat{U}$.

[7] https://github.com/psanch21/VACA
[8] https://github.com/piomonti/carefl/

Counterfactual Estimation. For a factual observation x^{F} and interventions on nodes $\mathcal{I}$ with values γ, the counterfactual estimate only differs from the factual estimate for all nodes that are intervened or downstream from an intervened node. We proceed in topological order. For each intervened node i, $\hat{x}_i^{\mathrm{CF}} := \gamma_i$. For each non-intervened node i downstream from an intervened node, define $\hat{u}_i^{\mathrm{F}} := x_i^{\mathrm{F}} - \hat{f}_i(x_{\mathrm{pa}_i}^{\mathrm{F}})$, the residual and estimated noise for the factual sample. Let $\hat{x}_{\mathrm{pa}_i}^{\mathrm{CF}}$ be counterfactual estimates of the parents of X_i. Then $\hat{x}_i^{\mathrm{CF}} := \hat{f}_i(\hat{x}_{\mathrm{pa}_i}^{\mathrm{CF}}) + \hat{u}_i^{\mathrm{F}}$.

Therefore, for counterfactual queries, if the true functional equation f_i is an additive noise model, then if $\hat{f}_i \approx f_i$, the regression model will have low counterfactual error. In fact, if $\hat{f}_i \equiv f_i$, then the regression model will have perfect counterfactual performance.

C.3 Details about Random Graph Generation

The random graph is comprised of ten nodes. We randomly sample this graph by generating a random upper triangular adjacency matrix where each entry in the upper triangular half is each to 1 with probability 30%. We then check that this graph is comprised of a single connected component (if not, we resample the graph). For a graphical representation, we provide an example in Figure 4.

C.4 Query Evaluation Frameworks for Synthetic Data Experiments

Observational Evaluation. We generate 1000 samples from both the fitted and true observational distribution and report the MMD between the two. Since DCM and the ANM use the empirical distribution for root nodes, we only take the MMD between nonroot nodes.

Interventional Evaluation. We consider interventions of individual nodes. For an intervention node i, we choose 20 intervention values $\gamma_1, \ldots, \gamma_{20}$, linearly interpolating between the 10% and 90% quantiles of the marginal distribution of node i to represent realistic interventions. Then for each intervention $\mathrm{do}(X_i := \gamma_j)$, we generate 100 values from the fitted model and true causal model, $\hat{X}$ and $X^\star$ for the samples from the fitted model and true model respectively. Since the intervention only affects the descendants of node i, we subset $\hat{X}$ and $X^\star$ to include only the descendants of node i, and compute the MMD on $\hat{X}$ and $X^\star$ to obtain a distance $\delta_{i,j}$ between the interventional distribution for the specific node and interventional value. Lastly, we report the mean MMD over all 20 intervention values and all intervened nodes. For the ladder graph, we choose to intervene on X_2 and X_3 as these are the farthest nodes upstream and capture the maximum difficulty of the intervention. For the random graph, we randomly select three non-sink nodes to intervene on. A formal description of our interventional evaluation framework is given in Algorithm 6.

Algorithm 6 Evaluation of Interventional Queries

```
for each intervention node i do
    γ_1, ..., γ_20 linearly interpolate 10% to 90% quantiles of node i
    for each intervention γ_j generated above do
        Intervene do(X_i := γ_j)
        Generate 100 samples X̂ from M̂ and X* from M* of descendants of i
        δ_{i,j} ← MM̂D(X̂, X*)
    end for
end for
Output mean of all δ_{i,j}
```

Counterfactual Evaluation. Similarly to interventional evaluation, we consider interventions of individual nodes and for node i, we choose 20 intervention values $\gamma_1, \ldots, \gamma_{20}$, linearly interpolating between the 10% and 90% quantiles of the marginal distribution of node i to represent realistic interventions. Then for each intervention $\mathrm{do}(X_i := \gamma_j)$, we generate 100 nonintervened factual samples x^{F}, and query for the estimated and true counterfactual values $\hat{x}^{\mathrm{CF}}$ and x^{CF} respectively. Similarly to before, $\hat{x}^{\mathrm{CF}}$ and x^{CF} only differ on the descendants of node i, therefore we only consider the subset of the descendants of node i. We compute the MSE $\delta_{i,j}$, since the counterfactual estimate and ground truth are point values, giving us an error for a specific node and interventional value. Lastly, we report the mean MSE over all 20 intervention values and all intervened nodes. We use the same intervention nodes as in the interventional evaluation mentioned above. A formal description of our counterfactual evaluation framework is given in Algorithm 7 (Appendix C).

Algorithm 7 Evaluation of Counterfactual Queries

for each intervention node i **do**
  $\gamma_1, \ldots, \gamma_{20}$ linearly interpolate 10% to 90% quantiles of node i
  for each intervention γ_j generated above **do**
    Generate 100 factual samples $x_1^{\mathrm{F}}, \ldots, x_{100}^{\mathrm{F}}$
    Intervene $\mathrm{do}(X_i := \gamma_j)$
    Using $\{x^{\mathrm{F}}\}_{k=1}^{100}$, compute counterfactual estimates $\{\hat{x}_k^{\mathrm{CF}}\}_{k=1}^{100}$ and true counterfactuals $\{x_k^{\mathrm{CF}}\}_{k=1}^{100}$ for all descendants of i
    $\delta_{i,j} \leftarrow \mathrm{MSE}(\{\hat{x}_k^{\mathrm{CF}}\}_{k=1}^{100}, \{x_k^{\mathrm{CF}}\}_{k=1}^{100})$
  end for
end for
Output mean of all $\delta_{i,j}$

C.5 Explanation of Error Inflation in fMRI Experiments

In our fMRI experiments, we compute the absolute error on a single interventional sample and compute the absolute value. As an intuition for why we cannot hope to have errors close to zero and for why the errors are relatively much closer together, consider the following toy problem.

Assume $X_1, \ldots, X_n, Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} \mathcal{N}(\theta, 1)$ and we observe $X_1, \ldots, X_n$ as data. Consider the two following statistical problems:

1. Learn a distribution $\hat{D}$ such that samples from $\hat{D}$ and samples $Y_1, \ldots, Y_n$ achieves low MMD.

2. Learn an estimator that estimates Y_1 well under squared error.

In the first statistical problem, if $\hat{D}$ is a reasonable estimator, we should expect that more data leads to lower a MMD, for example the error may decay at a $1/n$ rate. We should expect MMD values close to zero, and the magnitude of the performance is directly interpretable.

In the second statistical problem, under squared error, the problem is equivalent to

$$\min_c \mathbb{E}_{Y_1 \sim \mathcal{N}(\theta,1)} (Y_1 - c)^2 = \min_c \theta^2 + 1 - 2c\theta + c^2.$$

The optimal estimator is the mean θ and achieves a squared error of 1. Of course in practice we do not know θ, if we were to use the sample mean of $X_1, \ldots, X_n$, we would have an error of the order $1 + 1/n$. While we may still compare various estimators, e.g. sample mean, sample median, deep neural network, all the losses will be inflated by 1, causing the difference in performances to seem much more minute.

Our synthetic experiments hope to directly estimate the interventional distribution and computes the MMD between samples from the true and model's distributions, implying they are of the first statistical problem. The fMRI real data experiments aim to estimate a single intervention value from a distribution, implying they are of the second statistical problem. We should not expect these results to be very small, and the metric values should all be shifted by an intrinsic irreducible error.

D Additional Experiments

D.1 Running Time Experiments

We present the training times in minutes for one seed on the ladder graph using the default implementation and parameters. For a fair comparison, these are all evaluated on a CPU. Note that ANM is the fastest as it uses standard regression models, and our proposed DCM approach is about 7-9x faster than CAREFL and VACA. The generation (inference) times for all the methods are in the order of 1 second. For VACA and CAREFL, we use the implementation provided by the respective authors.

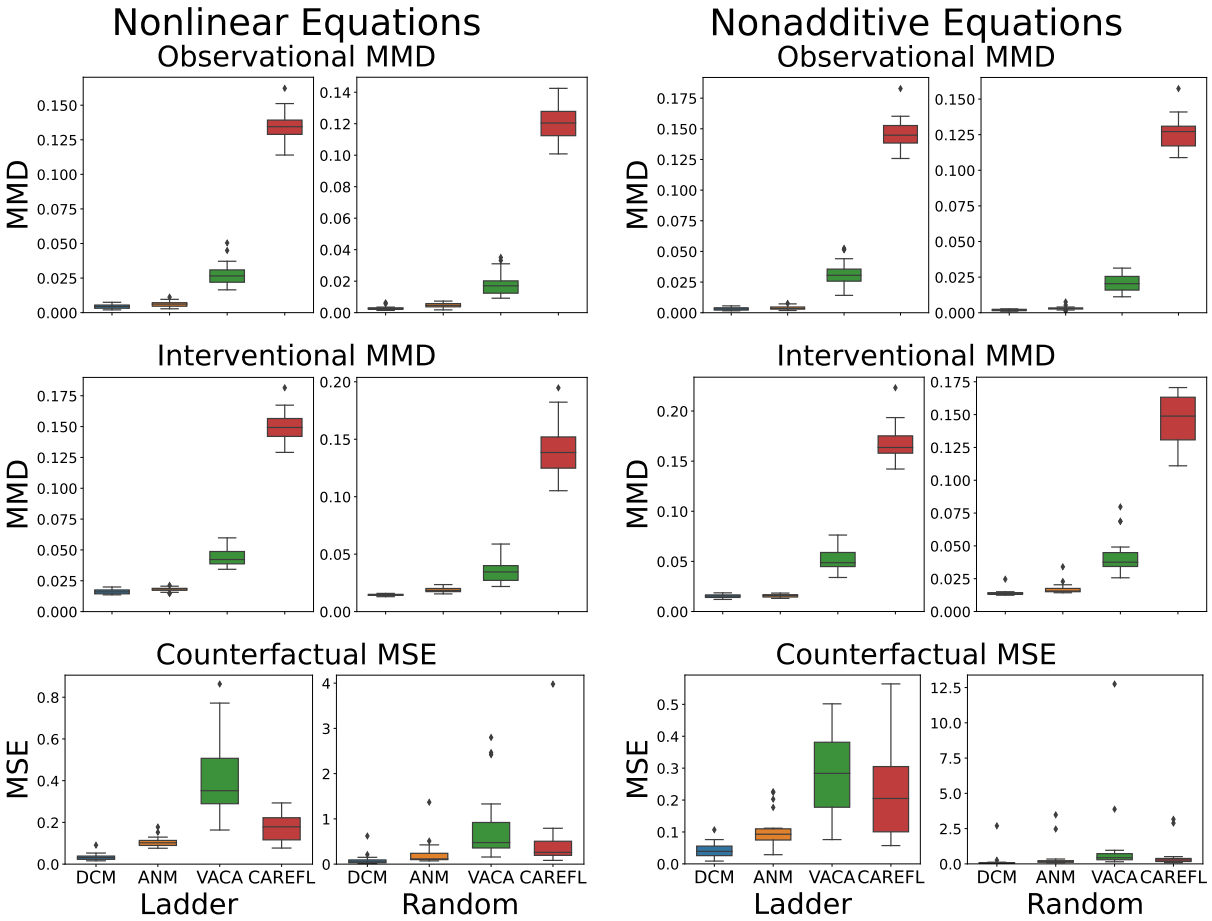

Figure 1: Left: Nonlinear setting (NLIN), Right: Nonadditive setting (NADD). Box plots of observational, interventional, and counterfactual queries of the ladder and random SCMs over 20 random initializations of the model and training data.

	DCM	ANM	VACA	CAREFL
Training Time (in minutes)	15.3	4.3	142.8	110.5

Table 4: Table of training times for a single run.

D.2 Semi-Synthetic Experiments

To further evaluate the effectiveness of DCM, we explore a semi-synthetic experiment based on the Sachs dataset (Sachs et al., 2005). We use the real world graph from comprised of 11 nodes[9]. The graph represents an intricate network of signaling pathways within human T cells. The 11 nodes within this graph each correspond to one of the phosphorylated proteins or phospholipids that were examined in their study.

For our experiment, we sample data in a semi-synthetic manner. For the root nodes, we sample from the empirical marginal distribution. For non-root nodes, since the ground truth structural equations are unknown, we use a random neural network as the structural equation, as was done in Section 5. We report the performances in Table 5. We see that the performance improvements with our DCM approach corroborate our prior findings.

[9] https://www.bnlearn.com/bnrepository/discrete-small.html#sachs

SCM		Metric	DCM ($\times 10^{-2}$)	ANM ($\times 10^{-2}$)	VACA ($\times 10^{-2}$)	CAREFL ($\times 10^{-2}$)
Sachs	NLIN	Obs. MMD	0.31±0.15	**0.21±0.04**	0.53±0.24	7.30±0.95
		Int. MMD	**1.25±0.11**	1.37±0.12	2.03±0.36	5.77±0.85
		CF. MSE	**0.72±0.19**	4.72±1.71	17.71±9.23	9.59±2.52
	NADD	Obs. MMD	**0.18±0.07**	0.18±0.06	0.39±0.24	6.10±1.14
		Int. MMD	**1.42±0.26**	1.86±0.51	2.21±0.98	5.29±2.07
		CF. MSE	**1.99±2.49**	4.68±5.77	7.30±11.57	8.30±8.34

Table 5: Mean±standard deviation of observational, interventional, and counterfactual queries of the Sachs SCM in nonlinear (NLIN) and nonadditive (NADD) settings over 10 random initializations of the model and training data. The values are multiplied by 100 for clarity.

D.3 Additional Synthetic Experiments

In this section, we present additional experimental evidence showcasing the superior performance of DCM in addressing various types of causal queries. We consider four additional smaller graph structures, which we call the *chain*, *triangle*, *diamond*, and *Y* graphs (see Figure 9).

The exact equations are presented in Table 6. These functional equations were chosen to balance the signal-to-noise ratio of the covariates and noise to represent realistic settings. Furthermore, these structural equations were chosen *after* hyperparameter selection, meaning we did not tune DCM's parameters nor tune the structural equations after observing the performance of the models.

Consider a node with value X with parents X_{pa} and exogenous noise U where $X_{\text{pa}} \perp\!\!\!\perp U$, and corresponding functional equation f such that

$$X = f(X_{\text{pa}}, U).$$

Further assuming the additive noise model, $X := f_1(X_{\text{pa}}) + f_2(U)$. In this additive setting, since $X_{\text{pa}} \perp\!\!\!\perp U$, we have

$$\text{Var}\left[X\right] = \text{Var}\left[f_1(X_{\text{pa}}\right] + \text{Var}\left[f_2(U)\right].$$

We choose f_1 and f_2 such that

$$0.05 \leq \frac{\text{Var}\left[f_2(U)\right]}{\text{Var}\left[f_1(X_{\text{pa}}\right]} \leq 0.5,$$

representing the fact that the ratio of the effect of the noise to the parents is roughly approximate or smaller by an order of magnitude.

For the nonadditive case, we decompose the variance using the law of total variance,

$$\text{Var}\left[X\right] = \mathbb{E}\,\text{Var}\left[f(X_{\text{pa}}, U) \mid U\right] + \text{Var}\,\mathbb{E}[f(X_{\text{pa}}, U) \mid U].$$

Similarly, we choose the functional equation f such that f satisfies

$$0.05 \leq \frac{\text{Var}\,\mathbb{E}[f(X_{\text{pa}}, U) \mid U]}{\mathbb{E}\,\text{Var}\left[f(X_{\text{pa}}, U) \mid U\right]} \leq 0.5$$

For all graphs, $U_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$, and we choose f such that $X_i = U_i$ if X_i is a root node, i.e. f is the identity function. Lastly, we normalize every node X_i such that $\text{Var}\left(X_i\right) \approx 1$. For the sake of clarity, we omit all normalizing terms in the formulas and omit functional equations for root nodes below.

Results. For these 4 graph structures (*chain*, *triangle*, *diamond*, and *Y*), in Table 7, we provide the performance of all evaluated models for observational, interventional, and counterfactual queries, averaged over 10 separate initializations

SCM		Nonlinear Case	Nonadditive Case
Chain	$f_2(U_2, X_1)$	$\exp(X_1/2) + U_2/4$	$1/((U_2 + X_1)^2 + 0.5)$
	$f_3(U_3, X_2)$	$(X_2 - 5)^3/15 + U_3$	$\sqrt{X_2 + \lvert U_3 \rvert}/(0.1 + X_2)$
Triangle	$f_2(U_2, X_1)$	$2X_1^2 + U_2$	$X_1/((U_2 + X_1)^2 + 1) + U_2/4$
	$f_3(U_3, X_1, X_2)$	$20/(1 + \exp(-X_2^2 + X_1)) + U_3$	$(\lvert U_3 \rvert + 0.3)(-X_1 + X_2/2 + \lvert U_3 \rvert/5)^2$
Diamond	$f_2(U_2, X_1)$	$X_1^2 + U_2/2$	$\sqrt{\lvert X_1 \rvert}(\lvert U_2 \rvert + 0.1)/2 + \lvert X_1 \rvert + U_2/5$
	$f_3(U_3, X_1, X_2)$	$X_2^2 - 2/(1 + \exp(-X_1)) + U_3/2$	$1/(1 + (\lvert U_3 \rvert + 0.5)\exp(-X_2 + X_1))$
	$f_4(U_4, X_2, X_3)$	$X_3/(\lvert X_2 + 2 \rvert + X_3 + 0.5) + U_4/10$	$(X_3 + X_2 + U_4/4 - 7)^2 - 20$
Y	$f_3(U_3, X_1, X_2)$	$4/(1 + \exp(-X_1 - X_2)) - X_2^2 + U_3/2$	$(X_1 - 2X_2 - 2)(\lvert U_3 \rvert + 0.2)$
	$f_4(U_4, X_3)$	$20/(1 + \exp(X_3^2/2 - X_3)) + U_4$	$(\cos(X_3) + U_4/2)^2$

Table 6: The equations defining the data generating process in the NLIN and NADD cases.

of models and training data, with the lowest value in each row bolded. The values are multiplied by 100 for clarity. In Figure 2, we show the box plots for the same set of experiments.

The results here are similar to those observed with the larger graph structures in Table 1. DCM has the lowest error in 7 out of 12 of the nonlinear settings, with the correctly specified ANM having the lowest error in the remaining 5. Furthermore, DCM and ANM both typically have a lower standard deviation compared to the other competing methods. For the nonadditive settings, DCM demonstrates the lowest values for all 12 causal queries.

SCM		Metric	DCM ($\times 10^{-2}$)	ANM ($\times 10^{-2}$)	VACA ($\times 10^{-2}$)	CAREFL ($\times 10^{-2}$)
Chain	NLIN	Obs. MMD	0.27±0.11	**0.19±0.27**	1.63±0.42	4.25±1.12
		Int. MMD	1.71±0.27	**1.70±0.47**	10.10±1.81	8.63±0.52
		CF. MSE	**0.33±0.16**	2.43±2.49	25.99±6.47	19.62±4.01
	NADD	Obs. MMD	**0.22±0.16**	1.51±0.43	1.53±0.41	3.48±1.06
		Int. MMD	**2.85±0.45**	7.34±0.62	10.42±2.77	11.02±1.32
		CF. MSE	**75.84±1.65**	88.56±2.63	98.82±4.16	105.80±11.04
Triangle	NLIN	Obs. MMD	0.16±0.11	**0.12±0.07**	3.12±0.93	4.64±1.03
		Int. MMD	**1.50±0.30**	3.28±0.79	18.43±1.72	7.08±0.82
		CF. MSE	**1.12±0.26**	9.80±1.70	178.69±16.45	41.85±19.58
	NADD	Obs. MMD	**0.25±0.12**	0.51±0.08	2.42±0.48	5.12±1.10
		Int. MMD	**2.81±0.21**	5.54±0.60	11.09±0.85	4.17±0.44
		CF. MSE	**26.28±6.68**	97.25±16.45	173.67±16.28	121.99±31.34
Y	NLIN	Obs. MMD	**0.11±0.05**	0.14±0.08	2.29±0.69	6.82±0.85
		Int. MMD	**1.23±0.08**	1.40±0.14	9.50±0.96	14.97±1.29
		CF. MSE	**0.28±0.25**	1.22±0.27	28.79±4.02	27.85±5.33
	NADD	Obs. MMD	**0.21±0.15**	1.00±0.19	1.51±0.44	3.39±0.58
		Int. MMD	**1.54±0.23**	5.62±0.31	5.37±0.72	6.51±0.40
		CF. MSE	**33.45±2.06**	47.55±2.56	60.67±3.24	52.41±6.69
Diamond	NLIN	Obs. MMD	0.22±0.10	**0.13±0.07**	2.77±0.45	8.28±1.29
		Int. MMD	3.21±0.62	**2.56±0.31**	25.30±1.39	18.23±3.01
		CF. MSE	**14.74±6.09**	32.02±37.74	138.65±12.33	607.62±241.21
	NADD	Obs. MMD	**0.25±0.18**	0.28±0.09	2.36±0.44	5.50±0.81
		Int. MMD	**1.88±0.23**	4.40±0.52	12.54±0.89	16.34±1.72
		CF. MSE	**1.36±0.14**	8.58±0.77	57.40±4.23	24.61±7.05

Table 7: Mean±standard deviation of observational, interventional, and counterfactual queries of four different SCMs in nonlinear and nonadditive settings over 10 random initializations of the model and training data. The values are multiplied by 100 for clarity.

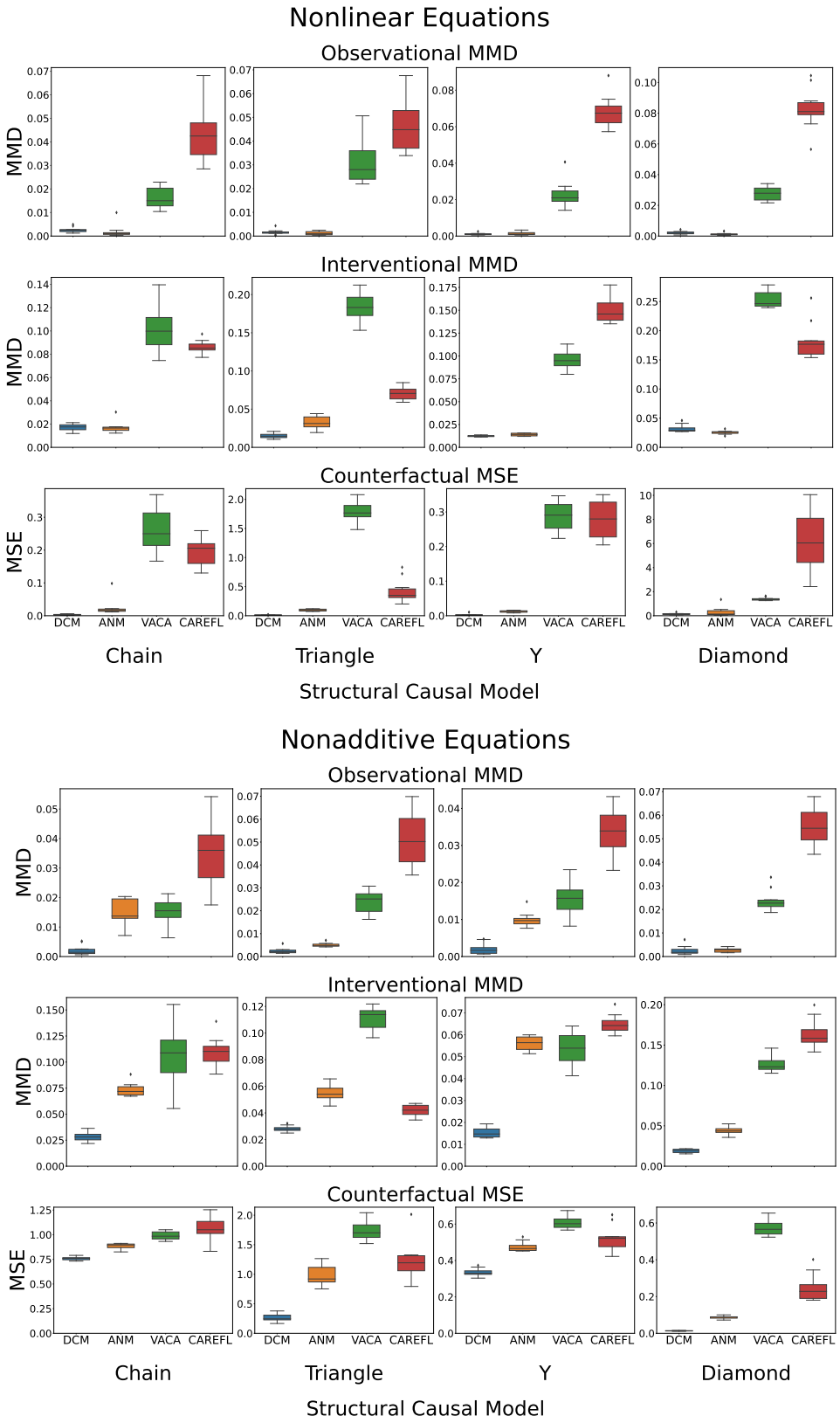

Figure 2: Top: Nonlinear setting (NLIN), Bottom: Nonadditive setting (NADD). Box plots of observational, interventional, and counterfactual queries of four different SCMs over 10 random initializations of the model and training data.

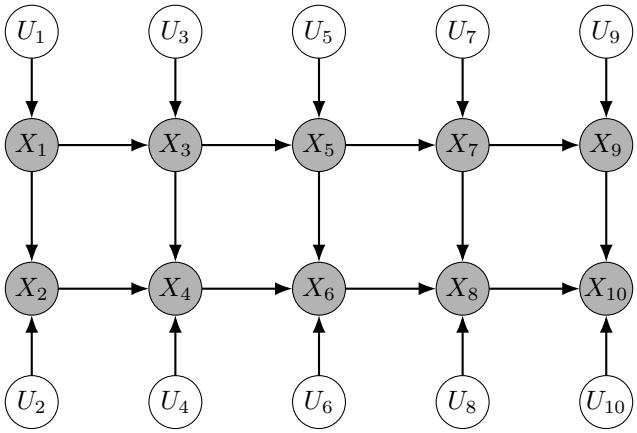

Figure 3: Ladder graph used in Section 5.

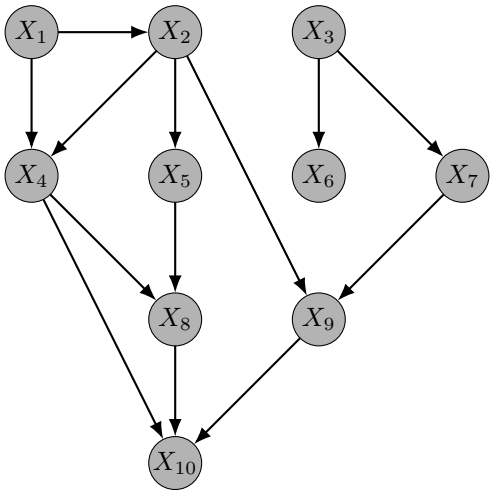

Figure 4: Example of a random graph used in Section 5, with exogenous noise nodes omitted for clarity.

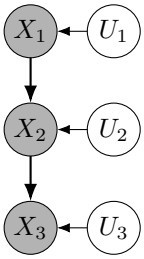

Figure 5: Chain graph.

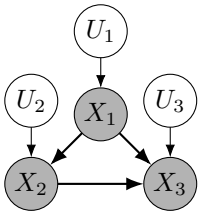

Figure 6: Triangle graph.

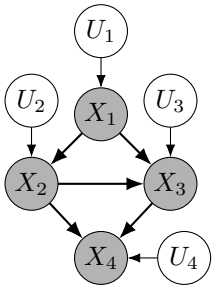

Figure 7: Diamond graph.

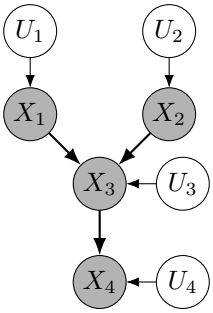

Figure 8: Y graph.

Figure 9: Causal graphs used in our experiments in Appendix D.3.