AMORTIZED COUNTERFACTUAL PREDICTION WITH GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual inference asks how an observed outcome might have differed under different circumstances, such as whether a patient would have recovered had they received a different treatment. Such questions arise whenever decisions depend on the outcomes of alternatives that were not observed. Answering counterfactual queries generally requires knowledge of the underlying causal structure, which is often unavailable in practice. A common approach is therefore to first estimate the causal graph from observational data and then perform counterfactual inference. However, errors in the estimated causal graph can propagate to downstream counterfactual predictions. To address this challenge, we introduce the **Counterfactual Graph Neural Process (CFGNP)**, an end-to-end approach that answers counterfactual queries directly from observational data without first estimating a causal graph. CFGNP uses Graph Neural Networks (GNNs) to exploit the dual-graph structure by restricting cross-world message passing to corresponding factual–counterfactual variables. Across synthetic and image-based benchmarks, CFGNP achieves stronger predictive performance than existing approaches, including on the German Loan benchmark, where it reduces MSE from 4.51×10^{-2} to 0.19×10^{-2}. Furthermore, on synthetic causal graphs, a model trained only on 10-node graphs maintains strong predictive performance on graphs with up to 30 nodes, without retraining.

1 INTRODUCTION

Understanding cause-and-effect relationships is a central goal in many scientific disciplines. In medicine, researchers may wish to determine whether a treatment improves patient outcomes (Stovitz & Shrier, 2019; Burotto et al., 2014). In economics, policymakers often aim to understand how changes in policy affect employment or inflation (Heckman & Pinto, 2022). Similar questions arise in biology, social sciences, and machine learning (Morgan & Winship, 2007; Sanderson et al., 2022; Schölkopf et al., 2021). Especially with the increased deployment of machine learning systems in settings where interpretability is crucial, understanding cause-and-effect relationships becomes more important (Li & Han, 2023). In all of these settings, the objective is not merely to observe statistical associations, but to understand how outcomes would change if different actions or conditions were imposed.

Causal inference (Pearl, 2009) provides a framework for studying such questions. Unlike conventional statistical learning, causal inference seeks to uncover cause-effect relationships between variables, rather than statistical associations. The framework formalises this with the structural causal model (SCM), which specifies how variables are generated, and how they causally affect one another. Within the SCM framework, causal questions can be distinguished by the type of reasoning they require. Pearl (Pearl, 2009) differentiates between three levels: associational, interventional, and counterfactual queries. Associational queries concern relationships observed in data, while interventional queries ask how outcomes change when a variable is actively manipulated (Pearl, 1996). Counterfactual queries go one step further by asking what would have happened in an alternative scenario, given what was observed in the factual world (Shpitser & Pearl, 2012). Counterfactual queries are particularly important because they allow reasoning about individualized outcomes and explanations, rather than only population-level effects (Nguyen et al., 2020). This makes them harder to solve than the other two query types as we additionally have to take the alternative hypothesis into account.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

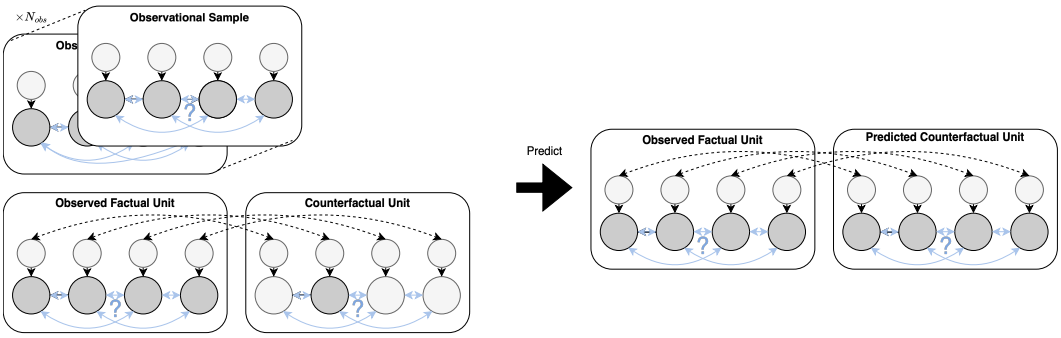

Figure 1: Overview of the task solved by CFGNP. Unknown variables are in light grey and known variables in dark grey. Blue edges represent missing knowledge about the causal graph. The counterfactual is predicted based on the counterfactual query and additional observational samples.

Traditional approaches to counterfactual inference typically rely on substantial prior knowledge about the underlying cause-effect relationships between variables (Xia et al., 2022; Rasal et al., 2025; Ribeiro et al., 2025; Nasr-Esfahany et al., 2023). In practice, this often means assuming access to the *causal graph*, which describes these relationships. Sometimes, less information, such as the topological node ordering of the causal graph, is required (Le et al., 2024). However, in many real-world settings, such information is unavailable or highly uncertain. This is particularly the case in domains such as medicine, biology, and the social sciences, where causal relationships may be obscured by latent confounders, incomplete observations, and complex dependencies among variables (Weberpals et al., 2024; Ruiz-Perez et al., 2024; Loh, 2025).

Hence, if there is no prior knowledge about the causal graph, we must first use observational samples to estimate the causal graph (Spirtes et al., 2001) and then use the resulting causal graph with the existing methods for counterfactual inference. However, this has two drawbacks; first, the search space of causal graphs scales super-exponentially in the number of variables (Zheng et al., 2018) and second, choosing the wrong causal graph amplifies errors and may cause the counterfactual to be incorrect (Karimi et al., 2020b). To avoid explicitly estimating the causal graph, we instead condition the counterfactual prediction directly on observational samples. This allows the model to extract information about the underlying causal structure that is relevant to the counterfactual query, without committing to a single estimated graph. We approach this problem through meta-learning using Neural Processes (NPs) (Garnelo et al., 2018), which learn to map a set of observations to a predictive distribution. Building on Graph Neural Processes (Carr & Wingate, 2019), we further incorporate the structural constraints imposed by the dual graph into the prediction.

Based on this approach, we introduce the *Counterfactual Graph Neural Process* (CFGNP), an end-to-end framework for counterfactual inference under unknown causal structure. An overview of our setting is shown in Figure 1.

Our contributions.

- We introduce the *Counterfactual Graph Neural Process* (CFGNP), an end-to-end framework for learning counterfactual distributions without prior knowledge of the underlying causal graph (Section 4).
- We develop a Graph Neural Process architecture that conditions counterfactual predictions on observational samples (Section 4.1).
- We demonstrate that CFGNP outperforms existing counterfactual inference methods on both synthetic and real-world data (Sections 5.1 and 5.2).
- We evaluate CFGNP on counterfactual queries involving features that are unseen during training and observe strong performance in this setting (Section 5.2).

2 Related Work

Causal Discovery. For causal inference tasks such as evaluating counterfactual queries, most existing methods rely on explicit knowledge of the causal graph. Estimating this graph from associational data is known as causal discovery (Verma & Pearl, 2022). The primary goal of causal discovery is to uncover true cause-and-effect relationships by learning the underlying structural dependencies rather than relying on statistical correlations alone (Pearl, 2009; Peters et al., 2017). Classical methodologies encompass constraint-based approaches, such as the PC algorithm (Spirtes et al., 2001), and score-based frameworks, such as Greedy Equivalence Search (Chickering, 2002). More recently, modern techniques including LiNGAM (Shimizu et al., 2006), NOTEARS (Zheng et al., 2018), and DAG-GNN (Yu et al., 2019) have advanced the field by introducing functional form assumptions, continuous optimization, and deep learning architectures. Nevertheless, key challenges remain, such as unobserved confounding (Kaltenpoth & Vreeken, 2023; Diepen et al., 2023), complex nonlinearities (Li et al., 2024), and computational scalability (Amin & Wilson, 2024; Cheng et al., 2024). Addressing these challenges continues to make causal inference an active area of research.

Counterfactual Inference. Existing approaches attempt to learn counterfactual distributions under varying degrees of prior structural knowledge (Khemakhem et al., 2021; Karimi et al., 2020a; Caron et al., 2022). For instance, Le et al. (2024) and Javaloy et al. (2023) use normalizing flows that require only a topological ordering of the graph nodes, whereas most alternatives assume the full causal graph. Within deep learning for counterfactual inference, Pawlowski et al. (2020) use normalizing flows to parameterize structural causal models, while recent flow-matching approaches similarly require full graph specification (Ribeiro et al., 2025). Other methods explicitly exploit relational structure through Graph Neural Networks (Zečević et al., 2021; Sanchez-Martin et al., 2021). This reliance on a known, estimated, or otherwise fixed causal structure can be restrictive (Wu et al., 2025; Caron et al., 2022): the space of DAGs grows super-exponentially with the number of variables (Zheng et al., 2018), and errors in the assumed causal structure can propagate to downstream counterfactual predictions (Karimi et al., 2020b). While end-to-end learning under uncertain causal structure has been explored for interventional inference (Geffner et al., 2022; Dhir et al., 2025), an analogous approach has not yet been established for counterfactual inference, where additional structural constraints from the dual graph can be exploited (von Kügelgen et al., 2022).

Image Counterfactuals. Several works focus specifically on image-based counterfactual generation and analysis (Pawlowski et al., 2020; Rasal et al., 2025; Luedtke & Fukumizu, 2025; Ma et al., 2026). Among these, DeepBC (Kladny et al., 2024) performs counterfactual inference through latent exogenous variables under a known causal graph. Other recent approaches based on diffusion models (Rasal et al., 2025) and optimal transport (Ribeiro et al., 2025) similarly assume that the causal graph is known. In contrast, VCI (Wu et al., 2025) does not take an explicit causal graph as input, but instead assumes a prescribed treatment–covariate–outcome structure and uses a variational formulation to generate counterfactuals without requiring counterfactual training pairs.

Neural Processes. To address these limitations, our approach builds on Neural Processes (NPs) (Garnelo et al., 2018), which have been successfully applied across a range of domains including Bayesian optimization and other scientific applications (Kim et al., 2019; Foong et al., 2020; Bruinsma et al., 2023). NPs have recently been explored in causal settings, including treatment-effect estimation and Bayesian causal inference (Bynum et al., 2026; Dhir et al., 2025). However, these works address related causal inference tasks rather than counterfactual inference. In contrast, our approach extends this framework to counterfactual inference, for which we additional leveraging GNNs to capture the structural relationships between factual and counterfactual variables (von Kügelgen et al., 2022). Graph Neural Processes extend NPs to relational and graph-structured data (Carr & Wingate, 2019; Liang & Gao, 2021; Hu et al., 2023). By combining these ideas, we directly map observational data to the target counterfactual distribution using GNNs over the dual-graph structure, without first estimating a causal graph.

3 Background

We now introduce notions of causal inference, neural processes, and graph neural networks.

Notation. Unstarred and starred variables denote the factual and counterfactual worlds, respectively. Let $z \in \mathbb{R}^{n \times d_{\text{data}}}$ be the factual observed unit, $T^* = t^*$ a counterfactual condition, and $Y^* \subseteq V$ the

target variables. The observational context is $O = \{O_i\}_{i=1}^{N_{\text{obs}}}$, consisting of i.i.d. samples from the same SCM $\mathcal{M}$, where n and N_{obs} denote the number of endogenous variables and context samples, respectively. We write $p(Y^* \mid O, z, T^* = t^*)$ when $\mathcal{M}$ is informed by the observational context O, and $p(Y^* \mid \mathcal{M}, z, T^* = t^*)$ when emphasizing the model. If $\mathcal{M}$ is unspecified, we assume the true model.

3.1 Structural Causal Models and Causal Inference

Structural Causal Model. A structural causal model (SCM) describes the causal relationships among a set of endogenous variables $V = \{V_1, \ldots, V_n\}$ through a collection of structural equations $\mathbf{f} = \{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$. Each variable V_k is determined by its direct causes $Pa_k \subseteq V \setminus V_k$ and an exogenous noise variable U_k according to

$$V_k = \mathbf{f}_k(Pa_k, U_k).$$

The exogenous variables $U = U_1, \ldots, U_n$ capture factors outside the model and are assumed to be mutually independent. The causal relationships encoded by the structural equations can be represented by a directed and acyclic causal graph $\mathcal{G}$, where an edge $V_j \to V_k$ indicates that $V_j \in Pa_k$. In particular, with U coming from a probability distribution we can define associational queries ($\mathcal{L}_1$) to concern distributions $p(Y)$ with $Y \subset V$. Observational samples O_i are sampled from $p(V)$. Interventional queries ($\mathcal{L}_2$) concern questions $p(Y \mid \text{do}(T = t))$ where we actively set nodes $T \subset V$ to a specific value and consider the outcome $Y \subset V$. (Backtracking) counterfactual queries ($\mathcal{L}_3$) are probability distributions, with $T^*, Y^*, Z \subset V$ and $T^* \cap Y^* = \emptyset$, of the form

$$P(Y^* = y^* \mid T^* = t^*, Z = z),$$

concerning questions such as: given that $Z = z$ happened, what would have happened to variables Y^* if alternatively $T^* = t^*$. In particular, we use notation $X^* = x^*$ instead of the do-operator to make clear that we are using backtracking counterfactuals (von Kügelgen et al., 2022) and not interventionist counterfactuals (Pearl, 2009). While interventionist counterfactuals artificially modify the structural equations in the counterfactual world, they thereby locally violate the laws of nature. In contrast, backtracking counterfactuals keep the structural equations intact and instead consider alternative exogenous conditions that could have produced $X^* = x^*$.

Dual Graph. To answer backtracking counterfactuals, we need the dual graph. Figure 2 shows an example of the dual graph for backtracking counterfactuals. The dual graph consists of two copies of the same SCM. The left part, the observed factual unit, represents what actually happened, whereas the copy on the right, the counterfactual unit, represents the alternative circumstances. Both worlds are connected via the exogenous nodes, which share a probability distribution $p(U, U^*)$.

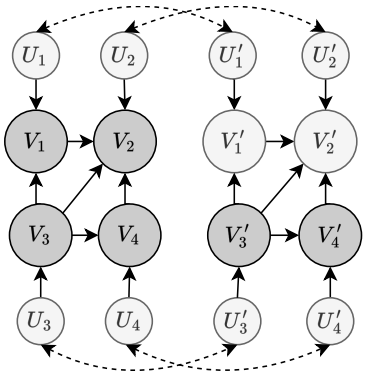

Figure 2: Example of the dual graph for a counterfactual query with a causal graph of four nodes. Light grey nodes are unknown.

We can solve the counterfactual query via the abduction–marginalisation–prediction framework (von Kügelgen et al., 2022), consisting of the following steps: **1.) Abduction:** We infer the joint posterior over the exogenous nodes $P(U, U^* \mid Z = z, T^* = t^*)$. **2.) Marginalisation:** We marginalise out U to obtain $P(U^* \mid Z = z, T^* = t^*)$. **3.) Prediction:** We use the SCM to predict $P(Y^* \mid Z = z, T^* = t^*)$.

Identifiability. When the underlying causal model is only partially known, a causal query may not be uniquely determined by the available information. An $\mathcal{L}_i$ query is identifiable from $\mathcal{L}_j$ information if all models in the considered model class that agree on the available $\mathcal{L}_j$ information yield the same answer (Pearl, 2009). This is relevant to our setting because counterfactual queries are not necessarily identifiable from observational data alone.

3.2 Neural Processes and Graph Neural Networks

Neural Processes. Neural Processes (NPs) (Garnelo et al., 2018) are probabilistic models that learn distributions over functions from a collection of related tasks. Each task is represented by a context

set $C = \{(x_i, y_i)\}_{i=1}^{N_C}$ of observed input-output pairs and a target set $G = \{(x_j, y_j)\}_{j=1}^{N_G}$ at which predictions are made. NPs learn the conditional predictive distribution $p(y_G \mid x_G, C)$, thereby amortizing inference across tasks by mapping context observations directly to predictive distributions. Extensions of NPs to graph-structured domains replace neural architectures with graph-based models to capture relational structure (Carr & Wingate, 2019; Ashman et al., 2024).

Graph Neural Networks. Graph Neural Networks (GNNs) learn representations of graph-structured data by iteratively propagating information between connected nodes. Let $h_i^{(k)}$ denote the representation of node v_i at layer k, with $h_i^{(0)} = x_i$. Following the message-passing framework (Gilmer et al., 2017), a GNN layer aggregates messages from the neighbourhood $\mathcal{N}(i)$ of each node,

$$m_i^{(k)} = \mathrm{AGG}^{(k)}\left(\left\{\phi^{(k)}(h_i^{(k)}, h_j^{(k)}, e_{ji}) \,\middle|\, v_j \in \mathcal{N}(i)\right\}\right), \qquad h_i^{(k+1)} = \psi^{(k)}(h_i^{(k)}, m_i^{(k)}),$$

where $\phi^{(k)}$ and $\psi^{(k)}$ are learnable message and update functions, respectively, and $\mathrm{AGG}^{(k)}$ is permutation invariant. For directed graphs, the neighbourhood can be partitioned into incoming and outgoing neighbours, $\mathcal{N}^-(i)$ and $\mathcal{N}^+(i)$, allowing messages to be propagated separately along each direction. This distinction is particularly relevant for causal graphs, where incoming and outgoing edges represent causal relations in opposite directions.

4 A GNN Model for Counterfactual Inference without Causal Graph Knowledge

This work focuses on predicting counterfactual distributions directly from observational data without prior knowledge about the causal graph. For each task, we observe a context set $O = \{z_i\}_{i=1}^{N_{\mathrm{obs}}}$ of i.i.d. observational samples from an unknown SCM $\mathcal{M} = (\mathcal{G}, \mathbf{f})$. Given a factual observation z and a counterfactual condition $T^* = t^*$, our goal is to predict the counterfactual distribution $p(Y^* \mid O, z, T^* = t^*)$ over a set of query variables Y^*. The observational context O provides evidence about $\mathcal{M}$. We aim to directly learn the model-averaged counterfactual predictive distribution

$$p(Y^* \mid O, z, T^* = t^*) = \mathbb{E}_{\mathcal{M}\sim p(\mathcal{M}\mid O,z)}\left[p(Y^* \mid \mathcal{M}, z, T^* = t^*)\right], \tag{1}$$

thereby accounting for uncertainty over causal models without explicitly inferring or committing to a particular causal graph. Existing methods on the other hand try to estimate the causal graph first.

Since we intend to approximate the counterfactual probability distribution, we use the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) as the minimization objective. Suppose we have a list L of samples $(\tau, y^*) \sim q$, where $\tau = ((z, T^*, t^*, Y^*), O)$ contains the factual representation z, counterfactual query, and observation pattern O, while y^* contains the corresponding counterfactual target values. Let $p_\theta(y^* \mid \tau)$ denote the distribution predicted by the model with weights θ, where y^* may contain multiple target variables.

We then want to obtain the weights that minimize the KL divergence, hence

$$\begin{aligned}\theta^* &= \arg\min_\theta \mathbb{E}_{\tau\sim q}\left[\mathrm{KL}(p(Y^* = y^*|z, T^* = t^*) \,||\, p_\theta(Y^* = y^*|z, T^* = t^*, O))\right] \\ &= \arg\max_\theta \mathbb{E}_{\tau\sim q}\Big[\mathbb{E}_{y^*|\tau}[\log p_\theta(Y^* = y^*|z, T^* = t^*, O)]\Big] + C\end{aligned}$$

for some constant C independent of the model parameters. During training we assume access to counterfactual pairs, as we sample $y^*| \tau \sim p(Y^* = y^*|z, T^* = t^*, O)$.

Recovery of the Counterfactual Distribution. In the limit of infinite data and model capacity, minimizing the KL divergence recovers the conditional predictive distribution (Bruinsma, 2022) $p(Y^* \mid O, z, T^* = t^*)$, with latent quantities, such as the exogenous variables, implicitly marginalized out (Gordon et al., 2019). If the counterfactual distribution is identifiable from observational data, this predictive distribution coincides with the counterfactual distribution of the ground-truth causal model. Otherwise, the model instead recovers the model-averaged counterfactual distribution over SCMs compatible with the observations.

4.1 Approach

Figure 3 illustrates the architecture of the Counterfactual Graph Neural Process (CFGNP). Given observational samples O, a factual observation z, and a counterfactual intervention x^*, CFGNP

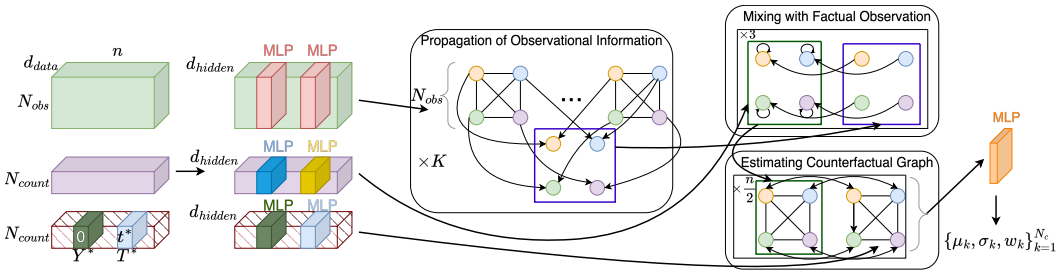

Figure 3: Overview of CFGNP. Observational samples are first aggregated into node-wise context representations, which are combined with the observed factual unit. Message passing between factual and counterfactual unit then produces parameters of the counterfactual predictive distribution.

predicts the corresponding counterfactual outcome distribution. The architecture consists of three stages: summarizing the observational context (middle), combining it with the factual observation (top right), and propagating the resulting representation to the counterfactual unit (bottom right). We describe each component below.

Embedding Input. We project all nodes to the hidden dimension d_{hidden} using node-wise MLPs. For O, we apply the same MLP h_O across all nodes. For z and x^*, we distinguish intervention nodes T^* from the remaining nodes using separate MLPs: $h_{z,\text{int}}$ and $h_{z,\text{common}}$ for z, and $h_{x,\text{int}}$ and $h_{x,\text{common}}$ for x^*. In x^*, all entries outside T^* are masked.

Observational Context Encoding. We next incorporate the observational samples O by allowing nodes within each sample O_j to communicate and compressing their information into q. For each variable i, we introduce an aggregation node q_i, initialized to zero, and connect the corresponding node i from every observational sample to q_i. Since we have no prior knowledge of the causal graph, we additionally fully connect the nodes within each observational sample.

The resulting graph consists of N_{obs} disjoint fully connected n-node subgraphs and n aggregation nodes. A K-layer GAT (Brody et al., 2022) performs message passing, $[O', q'] = m_\theta^{(1)}([O, q])$, such that q' captures the compressed observational context.

We then integrate this context with the factual unit z. To keep the graph compact, we connect each q_i' only to its corresponding z_i and add a self-loop to each z_i. A 3-layer GAT propagates the contextual information to the factual unit, $z' = m_\theta^{(2)}(z, q')$.

Predicting Counterfactual Distribution. The final stage aims to approximate the abduction–marginalisation–prediction steps using the context-informed factual representation z' and counterfactual condition x^*. The dual graph enables these steps by coupling the factual and counterfactual worlds through connected exogenous nodes. Approximating these steps generally requires bidirectional message passing.

Because we do not have prior knowledge of the causal graph, factual and counterfactual units are fully connected. Thus, each endogenous node has direct access to the information needed to reproduce its corresponding exogenous state, allowing us to remove explicit exogenous nodes and connect only corresponding factual and counterfactual variables across worlds. In this case, a unidirectional GNN can approximate the corresponding bidirectional GNN arbitrarily well (App. A.1).

As a design choice, we apply $\min\{\lfloor n/2 \rfloor, 5\}$ GAT layers, scaling depth with graph size while capping it at five to limit over-smoothing (Li et al., 2018). This yields $[z'', h^*] = m_\theta^{(3)}(z', x^*)$, where x^* encodes t^* at conditioned nodes and masks the rest.

The resulting h^* represents the counterfactual unit. To obtain the counterfactual predictive distribution, a node-wise MLP maps h^* to its distributional parameters. We use a mixture of N_c Gaussians, allowing CFGNP to represent uncertainty arising from ambiguity over the underlying SCM.

Training Objective. During training CFGNP minimizes the negative-log-likelihood objective, as this is equivalent to minimizing the KL divergence (Murphy, 2022):

$$\mathcal{L}_\theta = -\sum_{i=1}^{|L|}\sum_{j=1}^{d} \log\left(\sum_{k=1}^{N_c} w_k(\tau_i)_j \mathcal{N}\left(y_{i,j}^* | \mu_k(\tau_i)_j, \sigma_k(\tau_i)_j\right)\right).$$

Model Properties. The structure of CFGNP reflects two desirable properties. First, predictions should be independent of the ordering of variables. Permuting target nodes Y^* should induce the same permutation in the corresponding predictions, while permutations of the remaining nodes $(Y^*)^c$ should leave predictions for Y^* unchanged. This is provided by design of GNNs (App. A.3).

Second, a trained model should be applicable to graphs of different sizes and arbitrary numbers N_{obs} of observational samples (App. A.4). This enables CFGNP to generalize beyond the graph sizes and context sizes encountered during training.

We note for completeness that, for n nodes, data/hidden dimensions $d_{\text{data}}/d_{\text{hidden}}$, N_{obs} observations, K GNN iterations, and N_C mixture components, one forward pass has complexity (A.2):

$$\Theta\Big(n\, d_{\text{data}} d_{\text{hidden}} N_{\text{obs}} + K N_{\text{obs}} n d_{\text{hidden}}(d_{\text{hidden}} + n) + n d_{\text{hidden}}^2 + n^2 d_{\text{hidden}} + n d_{\text{hidden}} N_C\Big).$$

5 RESULTS

We evaluate CFGNP in terms of counterfactual prediction performance, generalization across causal graph sizes and unseen features, and computational efficiency, with additional experiments assessing its ability to recover analytically available counterfactual distributions with increasing context size (App. D.4). We also include an ablation study (App. D.5) where a normal transformer is run on the concatenated inputs. The results show that CFGNP performs much better than this.

Baselines. We compare against the following baselines:

- **VCI** (Wu et al., 2025): uses a conditional VAE with a counterfactual variational objective under a restricted causal structure.
- **DeepBC** (Kladny et al., 2024): performs counterfactual inference through exogenous variables in a learned SCM, assuming a known causal graph (CG).
- **VACA** (Sanchez-Martin et al., 2021): uses a variational graph autoencoder on a known CG.
- **BGM** (Nasr-Esfahany et al., 2023): represents structural equations as invertible mappings between exogenous and observed variables given the causal structure.
- **CFM** (Le et al., 2024): learns flow-based causal mechanisms assuming a given causal ordering.

Unlike the baseline methods, CFGNP uses counterfactual pairs during training, but no causal graph.

Metrics. We evaluate counterfactual predictions using the following metrics:

- **MSE/MAE:** Mean squared/absolute error (MSE/MAE) for numerical variables.
- **Accuracy/ROC-AUC:** Accuracy and ROC-AUC for categorical variables.

For datasets containing both numerical and categorical variables, we report metrics separately for each variable. For synthetic experiments with multiple variables, we follow Sanchez-Martin et al. (2021) and report MSE aggregated across all variables and samples.

Synthetic Datasets. For the synthetic experiments, we use the Triangle, M-shape, and German Loan benchmarks from Sanchez-Martin et al. (2021), together with randomly generated SCMs for the graph-size experiments. Triangle and M-shape have 3 and 5 nodes respectively and are evaluated with linear (LIN), nonlinear additive (NLIN), and nonlinear non-additive (NADD) mechanisms, while German Loan is a 7-node semi-synthetic seven-variable SCM inspired by the German Credit domain. Detailed definitions can be found in C.3.

Further, we conduct experiments on causal graphs of increasing size where each sample comes from its own SCM. We follow Soleymani et al. (2020). We generate the SCMs by randomly permuting the nodes into a topological order and adding each admissible edge $X_i \to X_j$ with probability $\frac{1}{2}$. Each variable X is generated from its direct causes using one of seven mechanisms f per sample, followed by posterior additive noise (App. C.4). We consider causal graphs of sizes $\mathcal{S} = 10, 15, 20, 25, 30$.

Real-world Datasets. For the real-world experiments, we use the CheXpert dataset (Irvin et al., 2019), consisting of chest X-rays (App. C.2). We augment each image with four explicit variables: age, sex, disease, and AP/PA, where AP/PA denotes the radiographic viewing direction. Following Ribeiro et al. (2025), we supply the baselines with a causal structure in which age affects disease, while age, sex, disease, and AP/PA determine the image. Since ground-truth counterfactuals are

unavailable (Ma et al., 2026), we evaluate whether generated images exhibit the intended counterfactual attributes using classifiers. Details on the classifiers and image-generating CFGNP variant are provided in App. C.2. We assess whether images resemble real-world images qualitatively.

5.1 Synthetic Experiments

On synthetic experiments, we compare against VACA (Sanchez-Martin et al., 2021), BGM (Nasr-Esfahany et al., 2023), and CFM (Le et al., 2024), which have demonstrated strong performance in prior work. VACA and BGM require the full causal graph, whereas CFM only requires its topological ordering, allowing comparison across different levels of causal knowledge. For the graph-size experiments, we omit CFM as it was not computationally viable on larger causal graphs.

Counterfactual Prediction. Table 1 shows the results for the synthetic datasets. CFGNP improves across almost all variants of the datasets and in many by a substantial amount. For instance, for German loan, CFGNP has an MSE of 0.19×10^{-2} compared to 4.51×10^{-2} of CFM. However, on the LIN variant of M-Shape, the MSE of 3.60×10^{-5} is higher than BGM's with 1.46×10^{-5}.

Dataset	Variant	BGM	VACA	CFM	CFGNP (Ours)
			MSE ↓		
Triangle	LIN	$(3.71 \pm 0.08) \times 10^{-3}$	$(23.52 \pm 1.83) \times 10^{-2}$	$(3.32 \pm 0.05) \times 10^{-3}$	$\mathbf{(2.59 \pm 2.13) \times 10^{-7}}$
	NLIN	$(4.78 \pm 0.97) \times 10^{-6}$	$(13.24 \pm 1.93) \times 10^{-2}$	$(1.35 \pm 0.24) \times 10^{-5}$	$\mathbf{(2.14 \pm 0.34) \times 10^{-7}}$
	NADD	$(1.80 \pm 1.80) \times 10^{-6}$	$(22.50 \pm 2.61) \times 10^{-2}$	$(1.54 \pm 0.40) \times 10^{-6}$	$\mathbf{(2.81 \pm 1.11) \times 10^{-7}}$
M-shape	LIN	$\mathbf{(1.46 \pm 0.40) \times 10^{-5}}$	$(12.83 \pm 0.69) \times 10^{-2}$	$(2.73 \pm 0.17) \times 10^{-5}$	$(3.60 \pm 7.00) \times 10^{-5}$
	NLIN	$(1.97 \pm 0.50) \times 10^{-5}$	$(16.26 \pm 0.95) \times 10^{-2}$	$(4.67 \pm 1.14) \times 10^{-5}$	$\mathbf{(2.50 \pm 0.47) \times 10^{-6}}$
	NADD	$(1.97 \pm 0.07) \times 10^{-3}$	$(41.06 \pm 0.31) \times 10^{-2}$	$(1.89 \pm 0.03) \times 10^{-3}$	$\mathbf{(1.08 \pm 0.35) \times 10^{-4}}$
German Loan	–	$(46.81 \pm 4.20) \times 10^{-2}$	$(10.30 \pm 0.40) \times 10^{-2}$	$(4.51 \pm 0.78) \times 10^{-2}$	$\mathbf{(0.19 \pm 0.09) \times 10^{-2}}$

Table 1: Mean MSE $\pm$ standard deviation achieved by BGM, VACA, CFM, and CFGNP (ours) across the Triangle, M-shape, and German Loan datasets. Lower values indicate better performance. Best scores are marked in bold.

Graph-Size Experiments. For experiments with increasing graph size we use two variants of CFGNP. In one variant, CFGNP is retrained for each $s \in \mathcal{S}$, while in the other, CFGNP is trained only on graphs with $s = 10$ nodes and applied to the other graph sizes without training. We refer to this model as 10-CFGNP. Additionally, as the benchmarks require a fixed causal graph, we provide the fullest DAG, connecting each node to all topological successors. This highlights an advantage of CFGNP: when causal graphs vary across the dataset, baselines must still assume a fixed graph.

Figure 4 shows MSE with increasing graph size (note the non-linear y-axis). 10-CFGNP performs similarly to retrained CFGNP for Linear and Sum-sine, but worse for Log-sum-exp and Sqrt-sum. MSE increases for all methods, but both CFGNP variants outperform the baselines and degrade more slowly. Other mechanisms are reported in App. D.2.

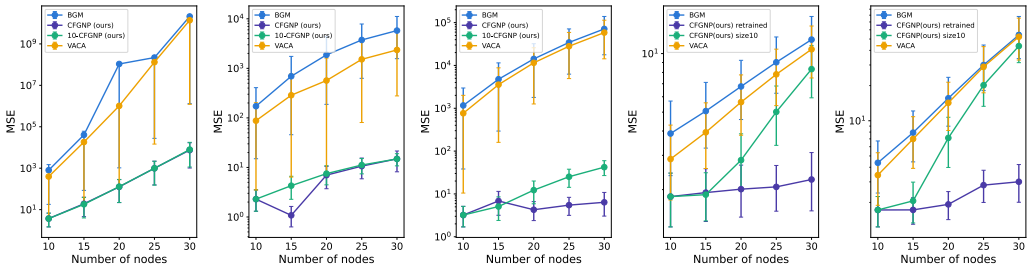

Figure 4: Performance (MSE) of models with regard to size of the causal graph. Models are compared on causal graphs with Linear (1.), Sum-sine (2.), Log-sum-exp (3.), Sqrt-sum (4.), and Sum-sqrt (5.) structural equations. MSE mean over distinct causal graphs. Vertical lines show 25th and 75th quantiles.

5.2 CheXpert Experiments

We compare against VCI (Wu et al., 2025) and DeepBC (Kladny et al., 2024) which both have previously demonstrated strong performance. Again, VCI assumes treatment–covariate–outcome structure while DeepBC needs full graph knowledge.

Counterfactual Prediction. Table 2 shows that CFGNP performs particularly well on age, reducing MAE from 10.88 for DeepBC to 2.71, and consistently achieves high ROC-AUC on categorical variables. For disease, for example, accuracies are around 0.87, while CFGNP achieves a ROC-AUC of 0.77, compared with 0.75 for DeepBC and 0.45 for VCI. Generated counterfactual images are provided in Appendix D.2.

Feature	Metric	CFGNP (Ours)	DeepBC	VCI
Sex	Accuracy ↑	$\mathbf{0.8452 \pm 0.0608}$	0.8222 ± 0.1220	0.5173 ± 0.0296
	ROC-AUC ↑	0.9155 ± 0.0540	$\mathbf{0.9204 \pm 0.0752}$	0.5238 ± 0.0122
Age	MAE ↓	$\mathbf{2.7136 \pm 0.7326}$	10.8750 ± 0.8732	66.8172 ± 62.1769
AP/PA	Accuracy ↑	$\mathbf{0.9664 \pm 0.0140}$	0.9572 ± 0.0552	0.8597 ± 0.0000
	ROC-AUC ↑	$\mathbf{0.9787 \pm 0.0136}$	0.9658 ± 0.0584	0.5741 ± 0.0342
Disease	Accuracy ↑	$\mathbf{0.8830 \pm 0.0368}$	0.8720 ± 0.0240	0.8548 ± 0.0667
	ROC-AUC ↑	$\mathbf{0.7710 \pm 0.0572}$	0.7517 ± 0.0327	0.4516 ± 0.0202

Table 2: Performance of CFGNP, VCI, and DeepBC on the CheXpert dataset. MAE is reported for the continuous feature, while accuracy and ROC-AUC are reported for categorical features. Arrows indicate whether higher or lower values are preferred. Best values marked in bold.

Out-of-Distribution Variable-Set Experiment. We evaluate CFGNP's generalization to unseen features on CheXpert. For each OOD setting, we exclude one feature k_{OOD} during training and evaluate on the full test set, leveraging CFGNP's ability to handle graphs of different sizes. Table 3 shows strong generalization for sex and AP/PA, with ROC-AUC above 0.99, while age and disease are more challenging, with an MAE of up to 11.53 and ROC-AUC between 0.61 and 0.64, respectively.

Feature	Metric	OOD Sex	OOD Age	OOD AP/PA	OOD Disease
Sex	Accuracy ↑	0.9516 ± 0.0108	0.9634 ± 0.0157	$\mathbf{0.9692 \pm 0.0084}$	0.9627 ± 0.0029
	ROC-AUC ↑	0.9927 ± 0.0037	0.9941 ± 0.0039	$\mathbf{0.9965 \pm 0.0017}$	0.9964 ± 0.0005
Age	MAE ↓	10.7037 ± 1.5283	11.5291 ± 0.0950	9.2965 ± 1.9205	$\mathbf{8.1875 \pm 0.4218}$
AP/PA	Accuracy ↑	0.9917 ± 0.0069	0.9904 ± 0.0085	0.9995 ± 0.0008	$\mathbf{0.9996 \pm 0.0005}$
	ROC-AUC ↑	0.9998 ± 0.0004	0.9998 ± 0.0003	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{1.0000 \pm 0.0000}$
Disease	Accuracy ↑	0.6560 ± 0.0176	0.6678 ± 0.0119	$\mathbf{0.6690 \pm 0.0073}$	0.6579 ± 0.0212
	ROC-AUC ↑	$\mathbf{0.6378 \pm 0.0068}$	0.6296 ± 0.0163	0.6058 ± 0.0264	0.6155 ± 0.0197

Table 3: OOD generalization on CheXpert. Each column indicates the feature excluded during training; evaluation is performed on the full test set containing all features. Best results in bold.

6 Conclusion

We propose CFGNP, an end-to-end architecture designed to learn counterfactual distributions directly from observational data without knowledge of the causal graph. CFGNP applies GNNs to enforce structural constraints on the dual graph implied by the notion of backtracking counterfactuals. By leveraging observational context samples alongside counterfactual queries, CFGNP directly approximates the Bayesian model-averaged counterfactual distribution. Evaluations on image and synthetic counterfactual queries have shown strong performance compared to baselines. Further experiments on synthetic datasets show that CFGNP retains strong performance on larger causal graphs compared with retraining from scratch, while outperforming baselines. Finally, we empirically validate that CFGNP performs adequately even on out-of-distribution features. In terms of limitations, the preceding capabilities come with a computational trade-off. The required observational context increases both training and inference costs. Moreover, if the counterfactual distribution is non-identifiable, CFGNP cannot recover the true distribution. However, this also applies to the baseline methods. In the future, one could incorporate priors about the causal graph. While the entire causal graph may not be known, an expert may have knowledge about the existence and absence of certain causal relationships. Taking this even further, one could provide simple distributions over edges, for instance obtained by amortized causal structure learning methods (Lorch et al., 2022).

AI USE STATEMENT

In this work, we used generative AI tools for refining writing, formulating mathematical claims, assisting in the writing of proofs, and feedback on research methodology or experiments. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, providing critical ingredients for proving mathematical claims, implementing methods, assisting with translation, cleaning and reformatting datasets, and interpreting results. Support for qualitative and thematic data analysis and proposing or refining hypotheses are not applicable to this work. We have reviewed all AI-assisted work. We checked that the generated formulation of the claim matches the intended claim. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses existing datasets and collects no new human-subject data. While counterfactual models may have applications in healthcare, their predictions can reflect biases and limitations in the data and should not be used for clinical decision-making without appropriate validation.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the source code and experimental setup at a later stage to ensure anonymity. Model architectures and hyperparameters are detailed in the appendix.

REFERENCES

Alan Nawzad Amin and Andrew Gordon Wilson. Scalable and flexible causal discovery with an efficient test for adjacency. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1331–1358. PMLR, 21–27 Jul 2024.

Matthew Ashman, Cristiana Diaconu, Junhyuck Kim, Lakee Sivaraya, Stratis Markou, James Requeima, Wessel P Bruinsma, and Richard E Turner. Translation equivariant transformer neural processes. *arXiv preprint arXiv:2406.12409*, 2024.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2022. URL https://arxiv.org/abs/2105.14491.

Wessel Bruinsma. *Convolutional Conditional Neural Processes*. PhD thesis, 2022. URL https://www.repository.cam.ac.uk/handle/1810/354383.

Wessel Bruinsma, Stratis Markou, James Requeima, Andrew Y. K. Foong, Tom Andersson, Anna Vaughan, Anthony Buonomo, Scott Hosking, and Richard E Turner. Autoregressive Conditional Neural Processes. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/bruinsma2023iclr-autoregressive/.

Mauricio Burotto, Victoria L. Chiou, Jung-Min Lee, and Elise C. Kohn. The mapk pathway across different malignancies: A new perspective. *Cancer*, 120(22):3446–3456, 2014. doi: https://doi.org/10.1002/cncr.28864. URL https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.1002/cncr.28864.

Lucius E. J. Bynum, Rajesh Ranganath, and Kyunghyun Cho. Computational identifiability, 2026. URL https://arxiv.org/abs/2606.19361.

Alberto Caron, Gianluca Baio, and Ioanna Manolopoulou. Counterfactual Learning with Multioutput Deep Kernels, November 2022. URL http://arxiv.org/abs/2211.11119. arXiv:2211.11119 [cs].

Andrew Carr and David Wingate. Graph neural processes: Towards bayesian graph neural networks, 2019. URL https://arxiv.org/abs/1902.10042.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11525–11533, 2024.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Anish Dhir, Cristiana Diaconu, Valentinian Mihai Lungu, James Requeima, Richard E. Turner, and Mark van der Wilk. Estimating Interventional Distributions with Uncertain Causal Graphs through Meta-Learning, July 2025. URL http://arxiv.org/abs/2507.05526. arXiv:2507.05526 [cs].

Mirthe Maria Van Diepen, Ioan Gabriel Bucur, Tom Heskes, and Tom Claassen. Beyond the markov equivalence class: Extending causal discovery under latent confounding. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing (eds.), *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pp. 707–725. PMLR, 11–14 Apr 2023.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes, July 2018. URL http://arxiv.org/abs/1807.01622. arXiv:1807.01622 [cs].

Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, Miltiadis Allamanis, and Cheng Zhang. Deep end-to-end causal inference, 2022. URL https://arxiv.org/abs/2202.02195.

Judy Wawira Gichoya, Imon Banerjee, Ananth Reddy Bhimireddy, John L Burns, Leo Anthony Celi, Li-Ching Chen, Ramon Correa, Natalie Dullerud, Marzyeh Ghassemi, Shih-Cheng Huang, et al. Ai recognition of patient race in medical imaging: a modelling study. *The Lancet Digital Health*, 4(6):e406–e414, 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard E. Turner. Meta-learning probabilistic inference for prediction, 2019. URL https://arxiv.org/abs/1805.09921.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

James J Heckman and Rodrigo Pinto. Causality and econometrics. Working Paper 29787, National Bureau of Economic Research, February 2022. URL http://www.nber.org/papers/w29787.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Junfeng Hu, Yuxuan Liang, Zhencheng Fan, Hongyang Chen, Yu Zheng, and Roger Zimmermann. Graph neural processes for spatio-temporal extrapolation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 752–763. ACM, August 2023. doi: 10.1145/3580305.3599372. URL http://dx.doi.org/10.1145/3580305.3599372.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019. URL https://arxiv.org/abs/1901.07031.

Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. Causal normalizing flows: from theory to practice, December 2023. URL http://arxiv.org/abs/2306.05415. arXiv:2306.05415 [cs].

Mehran Javanmardi, Mehdi Sajjadi, Ting Liu, and Tolga Tasdizen. Unsupervised total variation loss for semi-supervised deep learning of semantic segmentation, 2018. URL https://arxiv.org/abs/1605.01368.

David Kaltenpoth and Jilles Vreeken. Nonlinear causal discovery with latent confounders. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15639–15654. PMLR, 23–29 Jul 2023.

Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach, October 2020a. URL http://arxiv.org/abs/2006.06831. arXiv:2006.06831 [cs].

Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach, 2020b. URL https://arxiv.org/abs/2006.06831.

Ilyes Khemakhem, Ricardo Pio Monti, Robert Leech, and Aapo Hyvärinen. Causal Autoregressive Flows, February 2021. URL http://arxiv.org/abs/2011.02268. arXiv:2011.02268 [stat].

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Klaus-Rudolf Kladny, Julius von Kügelgen, Bernhard Schölkopf, and Michael Muehlebach. Deep Backtracking Counterfactuals for Causally Compliant Explanations, August 2024. URL http://arxiv.org/abs/2310.07665. arXiv:2310.07665 [cs].

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Minh Khoa Le, Kien Do, and Truyen Tran. Learning Structural Causal Models from Ordering: Identifiable Flow Models, December 2024. URL http://arxiv.org/abs/2412.09843. arXiv:2412.09843 [cs].

Chunlin Li, Xiaotong Shen, and Wei Pan. Nonlinear causal discovery with confounders. *Journal of the American Statistical Association*, 119(546):1205–1214, 2024.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Tong Li and Lu Han. Dealing with explainability requirements for machine learning systems. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1203–1208, 2023. doi: 10.1109/COMPSAC57700.2023.00182.

Huidong Liang and Junbin Gao. How neural processes improve graph link prediction, 2021. URL https://arxiv.org/abs/2109.14894.

Wen Wei Loh. Causal inference with unobserved confounding: leveraging negative control outcomes using lavaan. *Multivariate Behavioral Research*, 60(5):1029–1041, 2025.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning, 2022. URL https://arxiv.org/abs/2205.12934.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Alex Luedtke and Kenji Fukumizu. DoubleGen: Debiased Generative Modeling of Counterfactuals, September 2025. URL http://arxiv.org/abs/2509.16842. arXiv:2509.16842 [stat].

Haroui Ma, Francesco Quinzan, Theresa Willem, and Stefan Bauer. Ai alignment in medical imaging: Unveiling hidden biases through counterfactual analysis, 2026. URL https://arxiv.org/abs/2504.19621.

Stephen L Morgan and Christopher Winship. Counterfactuals and causal inference. *New York: Cambridge*, 2007.

Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual Identifiability of Bijective Causal Models, June 2023. URL http://arxiv.org/abs/2302.02228. arXiv:2302.02228 [stat].

Tri-Long Nguyen, Gary S Collins, Paul Landais, and Yannick Le Manach. Counterfactual clinical prediction models could help to infer individualized treatment effects in randomized controlled trials—an illustration with the international stroke trial. *Journal of clinical epidemiology*, 125: 47–56, 2020.

Nick Pawlowski, Daniel C. Castro, and Ben Glocker. Deep Structural Causal Models for Tractable Counterfactual Inference, October 2020. URL http://arxiv.org/abs/2006.06485. arXiv:2006.06485 [stat].

Judea Pearl. A causal calculus for statistical research. In *Learning from Data: Artificial Intelligence and Statistics V*, pp. 23–33. Springer, 1996.

Judea Pearl. *Causality*. Cambridge University Press, 2 edition, 2009.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Rajat Rasal, Avinash Kori, Fabio De Sousa Ribeiro, Tian Xia, and Ben Glocker. Diffusion Counterfactual Generation with Semantic Abduction, June 2025. URL http://arxiv.org/abs/2506.07883. arXiv:2506.07883 [cs].

Fabio De Sousa Ribeiro, Ainkaran Santhirasekaram, and Ben Glocker. Counterfactual Identifiability via Dynamic Optimal Transport, October 2025. URL http://arxiv.org/abs/2510.08294. arXiv:2510.08294 [cs].

Daniel Ruiz-Perez, Gimon Isabella, Sazal Musfiqur, Mathee Kalai, and Narasimhan Giri. Unfolding and de-confounding: biologically meaningful causal inference from longitudinal multi-omic networks using metalica. *Msystems*, 9(10), 2024.

Pablo Sanchez-Martin, Miriam Rateike, and Isabel Valera. VACA: Design of Variational Graph Autoencoders for Interventional and Counterfactual Queries, October 2021. URL http://arxiv.org/abs/2110.14690. arXiv:2110.14690 [stat].

Eleanor Sanderson, M Maria Glymour, Michael V Holmes, Hyunseung Kang, Jean Morrison, Marcus R Munafò, Tom Palmer, C Mary Schooling, Chris Wallace, Qingyuan Zhao, et al. Mendelian randomization. *Nature reviews Methods primers*, 2(1):6, 2022.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvarinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006. URL http://jmlr.org/papers/v7/shimizu06a.html.

Ilya Shpitser and Judea Pearl. What counterfactuals can be tested. *arXiv preprint arXiv:1206.5294*, 2012.

Ashkan Soleymani, Anant Raj, Stefan Bauer, Bernhard Schölkopf, and Michel Besserve. Causal feature selection via orthogonal search. *arXiv preprint arXiv:2007.02938*, 2020.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. The MIT press, 2001.

Steven D Stovitz and Ian Shrier. Causal inference for clinicians. *BMJ Evidence-Based Medicine*, 24(3):109–112, 2019. ISSN 2515-446X. doi: 10.1136/bmjebm-2018-111069. URL https://ebm.bmj.com/content/24/3/109.

Thomas S Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Probabilistic and causal inference: The works of Judea Pearl*, pp. 221–236. 2022.

Julius von Kügelgen, Abdirisak Mohamed, and Sander Beckers. Backtracking Counterfactuals, 2022. URL https://arxiv.org/abs/2211.00472. Version Number: 3.

Janick Weberpals, Sudha R Raman, Pamela A Shaw, Hana Lee, Massimiliano Russo, Bradley G Hammill, Sengwee Toh, John G Connolly, Kimberly J Dandreo, Fang Tian, et al. A principled approach to characterize and analyze partially observed confounder data from electronic health records. *Clinical Epidemiology*, pp. 329–343, 2024.

Yulun Wu, Louie McConnell, and Claudia Iriondo. Counterfactual Generative Modeling with Variational Causal Inference, March 2025. URL http://arxiv.org/abs/2410.12730. arXiv:2410.12730 [cs].

Kevin Xia, Yushu Pan, and Elias Bareinboim. Neural Causal Models for Counterfactual Identification and Estimation, September 2022. URL http://arxiv.org/abs/2210.00035. arXiv:2210.00035 [cs].

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7154–7163. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yu19a.html.

Matej Zečević, Devendra Singh Dhami, Petar Veličković, and Kristian Kersting. Relating Graph Neural Networks to Structural Causal Models, October 2021. URL http://arxiv.org/abs/2109.04173. arXiv:2109.04173 [cs].

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning, 2018. URL https://arxiv.org/abs/1803.01422.

A Appendix

A.1 Equivalence of a Direct MPNN to a Non-directional MPNN on the Dual Graph

This section provides an expressivity result justifying the removal of explicit exogenous nodes from the dual-graph architecture.

Proposition A.1 (Expressive sufficiency after eliminating exogenous nodes)**.** *Let G_{exo} denote the dual graph containing explicit exogenous nodes, and let G_{dir} denote the reduced graph obtained by removing the exogenous nodes and directly connecting each factual–counterfactual pair (x_i, x_i^*).*

Assume that the endogenous subgraph of G_{dir} is fully connected. Then, for every function of the endogenous nodes that can be approximated arbitrarily well by a bidirectional MPNN on G_{exo}, there exists a direct MPNN on G_{dir}, with sufficiently large hidden dimension, that approximates the same function arbitrarily well.

Proof. A bidirectional MPNN layer on G_{exo} has the form

$$h_v^{(\ell)} = \Phi_\ell\left(h_v^{(\ell-1)}, \sum_{(w,v)\in E} M_\ell^{\rightarrow}(h_v^{(\ell-1)}, h_w^{(\ell-1)}), \sum_{(v,w)\in E} M_\ell^{\leftarrow}(h_v^{(\ell-1)}, h_w^{(\ell-1)})\right)$$

where $M_\ell^{\rightarrow}$ and $M_\ell^{\leftarrow}$ are neural networks handling incoming and outcoming edges respectively.

Since the values of the exogenous variables are not observed, the initial exogenous embeddings are fixed constants,

$$h_{u_i}^{(0)} = c_U, \qquad h_{u_i^*}^{(0)} = c_{U^*}.$$

Consequently, the hidden state of every exogenous node is completely determined by the hidden states of neighbouring endogenous and exogenous nodes. G_{dir} does not have exogenous nodes, so we model them by doubling the hidden dimension of the endogenous nodes. Instead of explicitly representing u_i and u_i^*, the direct MPNN stores their hidden states inside the corresponding endogenous representations,

$$\widetilde{h}_{x_i}^{(\ell)} = \big(h_{x_i}^{(\ell)}, h_{u_i}^{(\ell)}\big), \qquad \widetilde{h}_{x_i^*}^{(\ell)} = \big(h_{x_i^*}^{(\ell)}, h_{u_i^*}^{(\ell)}\big).$$

Assume inductively that these augmented representations reproduce the hidden states of the bidirectional MPNN after layer $\ell - 1$.

The bidirectional update of an exogenous node is a continuous function

$$h_{u_i}^{(\ell)} = F_{\ell,i}\left(h_{u_i}^{(\ell-1)}, h_{u_i^*}^{(\ell-1)}, h_{x_i}^{(\ell-1)}\right),$$

and analogously for u_i^*. Since the endogenous graph is fully connected, node x_i receives the representation of x_i^* through an ordinary directed message. Hence, all arguments of $F_{\ell,i}$ are available locally in the augmented representations. By the universal approximation theorem (Hornik et al., 1989), an MLP can approximate $F_{\ell,i}$ arbitrarily well, allowing the direct MPNN to update the virtual exogenous states.

Next, consider the update of an endogenous node,

$$h_{x_i}^{(\ell)} = G_{\ell,i}\left(h_{x_i}^{(\ell-1)}, \{h_y^{(\ell-1)} : y \in N_{\mathrm{end}}(x_i)\}, h_{u_i}^{(\ell-1)}\right).$$

The virtual state $h_{u_i}^{(\ell-1)}$ is stored locally in $\widetilde{h}_{x_i}^{(\ell-1)}$. Because the endogenous graph is fully connected, node x_i receives the hidden state of every endogenous node. Hence, all arguments of $G_{\ell,i}$ are

available to the direct MPNN. Again, the update function can be approximated arbitrarily well by an MLP.

After each layer,

$$\widetilde{h}_{x_i}^{(\ell)} = \left(h_{x_i}^{(\ell)}, h_{u_i}^{(\ell)}\right),$$

and similarly for x_i^*. By induction, the hidden representations of all endogenous nodes produced by the bidirectional MPNN can be reproduced arbitrarily closely by the direct MPNN.

Since the readout is performed only on endogenous nodes, reproducing the exogenous hidden states explicitly is unnecessary.

□

A.2 CFGNP Complexity Analysis

We calculate the complexity in each part of the module. We first note that the GAT layer has computational complexity $\Theta(|\mathcal{V}|dd' + |\mathcal{E}|d')$ for a graph with $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges and the hidden dimensions go from d to d' (Brody et al., 2021). Let $|A|$ be the number of edges in the graph with adjacency matrix A.

Changing Data Dimension

Applying the MLP to each entry in $O \in \mathbb{R}^{(N_{\text{obs}} \cdot n) \times d_{\text{data}}}$ and $z, x^* \in \mathbb{R}^{(N_{\text{count}} \cdot n) \times d_{\text{data}}}$ takes

$$(N_{\text{obs}} n + 2N_{\text{count}} n)\, d_{\text{data}} d_{\text{hidden}} \in \Theta(n\, d_{\text{data}} d_{\text{hidden}} (N_{\text{count}} + N_{\text{obs}}))$$

operations. Since we use $N_{count} = 1$ throughout this work we get complexity

$$\Theta(n\, d_{\text{data}} d_{\text{hidden}} N_{\text{obs}})$$

Propagation of Observational Information

Each layer of the GAT uses adjacency matrix

$$G_c = \begin{pmatrix} M_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & I_n \\ \mathbf{0}_n & M_n & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_n & \vdots \\ \mathbf{0}_n & \cdots & \mathbf{0}_n & M_n & I_n \\ \mathbf{0}_n & \cdots & \cdots & \mathbf{0}_n & \mathbf{0}_n \end{pmatrix}.$$

With $|M_n| = n(n-1)$ and $|I_n| = n$ we get $|G_c| = N_{obs} n(n-1) + N_{obs} n$. Further, we have $|\mathcal{V}| = (N_{obs} + 1)n$. For one layer we then get computational complexity

$$\Theta((N_{obs} + 1) n d_{hidden}^2 + (N_{obs} n(n-1) + N_{obs} n) d_{hidden}).$$

For all K layers in the block, we have computational complexity

$$\Theta(K(N_{obs} + 1) n d_{hidden}^2 + K(N_{obs} n^2) d_{hidden}) = \Theta(K N_{obs} n d_{hidden} (d_{hidden} + n)).$$

Mixing Factual Information

Each layer in the module for mixing factual information uses adjacency matrix

$$G'_C = \begin{pmatrix} \mathbf{0}_n & I_n \\ \mathbf{0}_n & I_n \end{pmatrix}.$$

With $|G'_C| = 2n$ and $|\mathcal{V}| = 2n$ we get computational complexity $\Theta(n d_{hidden}^2 + n d_{hidden})$. All three layers for this module thus have complexity $\Theta(n d_{hidden}^2 + n d_{hidden})$.

ESTIMATING COUNTERFACTUAL GRAPH

Each layer in this block uses adjacency matrix

$$G_C'' = \begin{pmatrix} M_n & I_n \\ I_n & M_n \end{pmatrix}.$$

We then get $|G_C''| = 2n(n-1) + 2n = 2n^2$ and $|\mathcal{V}| = 2n$. This gives us computational complexity of $\Theta(nd_{hidden}^2 + n^2 d_{hidden})$ in each layer.

All $\min\{\lfloor \frac{n}{2} \rfloor, 5\}$ layers together give us computational complexity $\Theta(nd_{hidden}^2 + n^2 d_{hidden})$ for the entire block.

The MLP layer needs $\Theta(n\, d_{\text{hidden}} N_C)$ operations.

Overall, the computational complexity of CFGNP is

$$\Theta\Big(n\, d_{\text{data}} d_{\text{hidden}} N_{\text{obs}} + K N_{\text{obs}} n d_{\text{hidden}} (d_{\text{hidden}} + n) + n d_{\text{hidden}}^2 + n^2 d_{\text{hidden}} + n d_{\text{hidden}} N_C\Big).$$

A.3 PERMUTATION PROPERTIES OF CFGNP

Proposition A.2. *Let*

$$F(z, T^*, t^*, Y^*, O)$$

denote the output of the CFGNP model restricted to the target nodes Y^*. Then the CFGNP is

1. *permutation equivariant with respect to permutations of the nodes in Y^*; and*
2. *permutation invariant with respect to permutations of the nodes in $(Y^*)^c$.*

Proof. In the following, let M_n be the $n \times n$ matrix that has 0 on the diagonal and everywhere else 1. Let I_n be the $n \times n$ identity matrix. Finally, let $\mathbf{0}_n$ be the $n \times n$ matrix with only 0.

Let $P \in \{0,1\}^{n \times n}$ be the permutation matrix associated with a permutation π of the n variables. We use the convention $Pe_i = e_{\pi(i)}$, where e_i denotes the i-th standard basis vector. Consequently, for every node-indexed matrix $H \in \mathbb{R}^{n \times d}$,

$$(PH)_j = H_{\pi^{-1}(j)}.$$

A permutation represents only a relabelling of the variables and must therefore be applied consistently to every node-indexed input. We define

$$z_P := Pz, \qquad x_P^* := Px^*, \qquad O_{k,P} := PO_k$$

for every observational sample O_k. We further define

$$T_\pi^* := \pi(T^*), \qquad Y_\pi^* := \pi(Y^*).$$

Let $\tau_{T^*} \in \{0,1\}^n$ be the indicator vector of the intervention set T^*, i.e.,

$$(\tau_{T^*})_i = \begin{cases} 1, & i \in T^*, \\ 0, & i \notin T^*. \end{cases}$$

The indicator vector of the permuted intervention set T_π^* is $\tau_{T^*,P} := P\tau_{T^*}$. Indeed, for every $j \in \{1, \ldots, n\}$,

$$(\tau_{T^*,P})_j = (P\tau_{T^*})_j = (\tau_{T^*})_{\pi^{-1}(j)},$$

which is equal to 1 precisely when $\pi^{-1}(j) \in T^*$, or equivalently when $j \in \pi(T^*)$.

Recall that the counterfactual input x^* contains the intervention values at the indices in T^* and is zero elsewhere:

$$x_i^* = \begin{cases} t_i^*, & i \in T^*, \\ 0, & i \notin T^*. \end{cases}$$

For every j,

$$(x^*_P)_j = (Px^*)_j = x^*_{\pi^{-1}(j)}.$$

$$(x^*_P)_j = \begin{cases} t^*_{\pi^{-1}(j)}, & \pi^{-1}(j) \in T^*, \\ 0, & \pi^{-1}(j) \notin T^*. \end{cases} \quad \Leftrightarrow \quad (x^*_P)_j = \begin{cases} (Pt^*)_j, & j \in T^*_\pi, \\ 0, & j \notin T^*_\pi. \end{cases}$$

Thus, the intervention values and the intervention indices are permuted consistently.

We now show the characteristics for each of the model components.

Equivariance of nodewise MLPs. We first consider the nodewise MLPs applied at the beginning of the model. For every observational sample O_k, the same MLP h_O is applied independently to every node. Therefore, $h_O(O_{k,P}) = Ph_O(O_k)$.

For the factual observation z, define the nodewise transformation

$$h_z(z, \tau_{T^*})_i = (\tau_{T^*})_i h_{z,\text{int}}(z_i) + \big(1 - (\tau_{T^*})_i\big) h_{z,\text{common}}(z_i).$$

We now verify permutation equivariance coordinatewise. For an arbitrary $j \in \{1, \dots, n\}$:

$$\begin{aligned} h_z(z_P, \tau_{T^*,P})_j &= (\tau_{T^*,P})_j h_{z,\text{int}}\big((z_P)_j\big) + \big(1 - (\tau_{T^*,P})_j\big) h_{z,\text{common}}\big((z_P)_j\big) \\ &= (\tau_{T^*})_{\pi^{-1}(j)} h_{z,\text{int}}\big(z_{\pi^{-1}(j)}\big) \\ &\quad + \big(1 - (\tau_{T^*})_{\pi^{-1}(j)}\big) h_{z,\text{common}}\big(z_{\pi^{-1}(j)}\big) \\ &= h_z(z, \tau_{T^*})_{\pi^{-1}(j)} \\ &= \big(Ph_z(z, \tau_{T^*})\big)_j. \end{aligned}$$

Since this equality holds for every j,

$$h_z(Pz, P\tau_{T^*}) = Ph_z(z, \tau_{T^*}).$$

The same argument applies to the counterfactual input.

$$h_x(Px^*, P\tau_{T^*}) = Ph_x(x^*, \tau_{T^*}).$$

Equivariance of the GATs. We now consider the GATs. A GAT m_θ is permutation equivariant in the sense that, for a node-feature matrix H, adjacency matrix G, and permutation matrix Q,

$$m_\theta(QH, QGQ^\top) = Qm_\theta(H, G).$$

For the observational graph, the same permutation is applied to every observational sample and to the aggregation nodes q. The corresponding block-diagonal permutation matrix is

$$Q_O = \text{diag}\left(\underbrace{P, \dots, P}_{N_{\text{obs}}+1 \text{ times}}\right) = I_{N_{\text{obs}}+1} \otimes P.$$

The adjacency matrix G_c consists only of blocks M_n, I_n, and $\mathbf{0}_n$. Since

$$PM_nP^\top = M_n, \qquad PI_nP^\top = I_n, \qquad P\mathbf{0}_nP^\top = \mathbf{0}_n,$$

we have $Q_O G_c Q_O^\top = G_c$. This suppresses the unchanged adjacency matrix from the notation,

$$m^{(1)}_\theta\left(Q_O \begin{pmatrix} O' \\ q \end{pmatrix}\right) = Q_O m^{(1)}_\theta\left(\begin{pmatrix} O' \\ q \end{pmatrix}\right) = Q_O \begin{pmatrix} \widetilde{O} \\ q' \end{pmatrix}.$$

It follows in particular that $q'_P = Pq'$.

For the graph containing the factual representation z and the aggregated observational representation q', define the permutation

$$Q_{zq} = \begin{pmatrix} P & \mathbf{0}_n \\ \mathbf{0}_n & P \end{pmatrix}.$$

Because $Q_{zq} G'_C Q_{zq}^\top = G'_C$, permutation equivariance gives

$$\begin{aligned} m_\theta^{(2)}\left(Q_{zq}\begin{pmatrix} z \\ q' \end{pmatrix}\right) &= Q_{zq} m_\theta^{(2)}\left(\begin{pmatrix} z \\ q' \end{pmatrix}\right) \\ &= Q_{zq}\begin{pmatrix} z' \\ q'' \end{pmatrix} \\ &= \begin{pmatrix} Pz' \\ Pq'' \end{pmatrix}. \end{aligned}$$

For the factual–counterfactual graph, define

$$Q_{zx^*} = \begin{pmatrix} P & \mathbf{0}_n \\ \mathbf{0}_n & P \end{pmatrix}.$$

With

$$Q_{zx^*} G''_C Q_{zx^*}^\top = G''_C,$$

we similarly obtain

$$\begin{aligned} m_\theta^{\mathrm{cf}}\left(Q_{zx^*}\begin{pmatrix} z' \\ x^* \end{pmatrix}\right) &= Q_{zx^*} m_\theta^{\mathrm{cf}}\left(\begin{pmatrix} z' \\ x^* \end{pmatrix}\right) \\ &= Q_{zx^*}\begin{pmatrix} z'' \\ x^{*\prime} \end{pmatrix} \\ &= \begin{pmatrix} Pz'' \\ Px^{*\prime} \end{pmatrix}. \end{aligned}$$

In particular, the counterfactual representation satisfies ${x^{*\prime}}_P = Px^{*\prime}$.

The final MLP. The final MLP is applied independently to every node of $x^{*\prime}$, with the same parameters at each node. Therefore,

$$m_\theta^{(3)}(Px^{*\prime}) = P m_\theta^{(3)}(x^{*\prime}).$$

Let

$$R := m_\theta^{(3)}(x^{*\prime})$$

denote the nodewise output containing the mixture parameters. The preceding arguments imply that the nodewise output of the model transforms as

$$R_P = PR.$$

The final nodewise output $R \in \mathbb{R}^{n \times 3N_c}$ contains one row for each node, and the CFGNP returns only the rows indexed by the target set:

$$F(z, T^*, t^*, Y^*, O) = R\big|_{Y^*}.$$

Moreover, the target set is relabelled according to $Y^*_\pi = \pi(Y^*)$. Since

$$(PR)_{\pi(i)} = R_i \qquad \text{for every } i \in Y^*,$$

restricting PR to the indices in Y^*_π yields precisely the same nodewise outputs as restricting R to Y^*, but indexed according to the permuted target nodes. Hence,

$$F(z_P, T^*_\pi, t^*_P, Y^*_\pi, O_P) = (PR)\big|_{Y^*_\pi} = P_{Y^*}\left(R\big|_{Y^*}\right),$$

where P_{Y^*} denotes the permutation induced by π on the target outputs. Therefore,

$$F(z_P, T^*_\pi, t^*_P, Y^*_\pi, O_P) = P_{Y^*} F(z, T^*, t^*, Y^*, O).$$

This establishes permutation equivariance with respect to the target nodes. If the permutation acts only on nodes in $(Y^*)^c$, then every target node is fixed and the induced permutation P_{Y^*} is the identity. The same identity therefore reduces to

$$F(z_P, T^*_\pi, t^*_P, Y^*, O_P) = F(z, T^*, t^*, Y^*, O),$$

which establishes permutation invariance with respect to permutations of the non-target nodes. □

A.4 Graph-Size and Observation-Count Transferability of CFGNP

The CFGNP architecture can be applied to causal graphs with different numbers of nodes and to observational datasets containing different numbers of samples. In particular, the learnable parameters of the model do not depend on either the graph size n or the number of observational samples N_{obs}. Consequently, a model trained using graphs with n nodes and N_{obs} observational samples can be applied, without modifying its learned parameters, to graphs with m nodes and N'_{obs} observational samples.

We formalize this property in the following proposition.

Proposition A.3 (Graph-size and observation-count transferability)**.** *Let*

$$F_\theta^{(n,N_{\text{obs}})}$$

denote the CFGNP applied to a causal query with n endogenous variables and N_{obs} observational samples. Suppose that the number of message-passing layers is fixed depending on n and independently of N_{obs}, or, alternatively, repeated message-passing iterations reuse the same parameters.

Then the parameter space Θ of $F_\theta^{(n,N_{\text{obs}})}$ is independent of both n and N_{obs}. Consequently, for arbitrary

$$n, m, N_{\text{obs}}, N'_{\text{obs}} \in \mathbb{N},$$

the same parameter vector $\theta \in \Theta$ defines both

$$F_\theta^{(n,N_{\text{obs}})} \qquad \textit{and} \qquad F_\theta^{(m,N'_{\text{obs}})}.$$

Hence, a CFGNP trained on one graph size and one number of observational samples can be evaluated on different graph sizes and different numbers of observational samples without introducing, removing, or resizing any learned parameters.

Proof. We show that every learnable component of the CFGNP is independent of both the number of endogenous variables n and the number of observational samples N_{obs}.

First, consider the node-wise input transformations. Let

$$h_O : \mathbb{R}^{d_{\text{data}}} \to \mathbb{R}^{d_{\text{hidden}}}$$

denote the MLP applied to the observational data. For

$$O \in \mathbb{R}^{N_{\text{obs}} \times n \times d_{\text{data}}},$$

the same map h_O is applied independently to every node in every observational sample,

$$O_{k,i} \mapsto h_O(O_{k,i}), \qquad k = 1, \dots, N_{\text{obs}}, \qquad i = 1, \dots, n.$$

Thus, changing either n or N_{obs} changes only the number of evaluations of h_O, but not its input dimension, output dimension, or learned parameters. The same argument applies to the node-wise maps $h_{z,\text{int}}$, $h_{z,\text{common}}$, $h_{x,\text{int}}$, and $h_{x,\text{common}}$.

Next, consider a GAT layer. For a node i, its update is of the form

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\right),$$

where

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})},$$

and

$$e_{ij} = \text{LeakyReLU}\left(a^\top [Wh_i \| Wh_j]\right).$$

The learnable quantities W and a depend only on the feature dimensions. They do not depend on the total number of nodes in the graph, the number of edges, or the number of observational samples represented in the graph. Changing n or N_{obs} therefore changes only the set of nodes and neighborhoods on which the same local update rule is evaluated.

We now consider the observational aggregation graph G_c. For n endogenous variables and N_{obs} observational samples, its adjacency matrix is

$$G_c^{(n,N_{\text{obs}})} = \begin{pmatrix} M_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & I_n \\ \mathbf{0}_n & M_n & \ddots & \vdots & I_n \\ \vdots & \ddots & \ddots & \mathbf{0}_n & \vdots \\ \mathbf{0}_n & \cdots & \mathbf{0}_n & M_n & I_n \\ \mathbf{0}_n & \cdots & \cdots & \mathbf{0}_n & \mathbf{0}_n \end{pmatrix}.$$

Thus, $G_c^{(n,N_{\text{obs}})}$ contains N_{obs} diagonal blocks M_n, one for each observational sample, and N_{obs} corresponding blocks I_n connecting the observational nodes to the aggregation nodes q.

Changing N_{obs} only changes the number of M_n- and I_n-blocks, while changing n only changes their dimensions. The construction rule itself remains unchanged. Since the GAT $m_\theta^{(1)}$ applies the same attention and message-passing functions to every node and edge, its learned parameters are independent of both n and N_{obs}.

In particular, if

$$[O', q'] = m_\theta^{(1)}([O', q]; G_c),$$

then changing the number of observational samples only changes the size of O' and the number of incoming messages available to the aggregation nodes q. It does not change the dimensionality of the representation $q'_i \in \mathbb{R}^{d_{\text{hidden}}}$ associated with each endogenous variable. Hence, the remainder of the architecture receives the same type of representation regardless of N_{obs}.

Next, consider the graph used to combine the observational representation with the factual observation. For n variables,

$$G_C'^{(n)} = \begin{pmatrix} \mathbf{0}_n & I_n \\ \mathbf{0}_n & I_n \end{pmatrix}.$$

For m variables, the corresponding graph is

$$G_C'^{(m)} = \begin{pmatrix} \mathbf{0}_m & I_m \\ \mathbf{0}_m & I_m \end{pmatrix}.$$

These graphs have different dimensions but are generated according to the same construction rule. Moreover, the number N_{obs} no longer appears explicitly at this stage, because the observational samples have already been aggregated into the n representations contained in q'. Therefore, the parameters of the GAT used in this stage are independent of both n and N_{obs}.

The same argument applies to the graph used for counterfactual inference. For n variables,

$$G_C''^{(n)} = \begin{pmatrix} M_n & I_n \\ I_n & M_n \end{pmatrix},$$

whereas for m variables,

$$G_C''^{(m)} = \begin{pmatrix} M_m & I_m \\ I_m & M_m \end{pmatrix}.$$

Changing the graph size changes only the number of nodes and edges on which the shared message-passing functions are evaluated. The learned parameters of the GAT are unchanged. Furthermore, N_{obs} does not affect the dimensionality of the node representations entering this stage.

Finally, the output network is applied independently and identically to each target node,

$$m_\theta^{(3)} : \mathbb{R}^{d_{\text{hidden}}} \to \mathbb{R}^{3N_c}.$$

Changing the number of endogenous variables changes only the number of node-wise predictions, while changing N_{obs} has no effect on the dimensions of the output map. Hence, neither quantity changes the learned parameters of $m_\theta^{(3)}$.

Consequently, every learned component of the CFGNP is parameterized only by quantities such as d_{data}, d_{hidden}, N_c, and not by n or N_{obs}. The matrices M_n, I_n, and $\mathbf{0}_n$ and the repetition of the

observational blocks determine only the structure and size of the computational graph; they contain no learned parameters. Therefore,

$$\Theta_{n,N_{\text{obs}}} = \Theta_{m,N'_{\text{obs}}} = \Theta$$

for arbitrary

$$n, m, N_{\text{obs}}, N'_{\text{obs}} \in \mathbb{N}.$$

Thus, the same parameter vector θ can be applied to causal graphs with different numbers of nodes and to observational datasets containing different numbers of samples. □

This establishes architectural transferability across both graph size and observation count. It does not, by itself, guarantee identical predictive performance when n or N_{obs} differs substantially from the values encountered during training, since such performance additionally depends on the distribution of training and test instances.

B Extended Model Architecture

In the following, let M_n be the $n \times n$ matrix that has 0 on the diagonal and everywhere else 1. Let I_n be the $n \times n$ identity matrix. Finally, let $\mathbf{0}_n$ be the $n \times n$ matrix with only 0.

B.1 Model Input

Our input is a tuple $((z, T^*, t^*, Y^*), O)$ with $O \in \mathbb{R}^{N_{obs} \times n \times d_{data}}$, $z \in \mathbb{R}^{1 \times n \times d_{data}}$, and $x^* \in \mathbb{R}^{N_{count} \times n \times d_{data}}$.

B.2 Changing data dimension

We apply MLPs on each node. On O we apply the same MLP h_O across all nodes. However, for the other parts we want the model to learn to distinguish between nodes we intervene on and the rest. Therefore, we apply an MLP $h_{z,int}$ on the indices T^* in z and an MLP $h_{z,common}$ on the other nodes. Similarly, we apply the MLP $h_{x,int}$ on the indices T^* of x^* and an MLP $h_{x,common}$ on the other indices.

B.3 Propagation of Observational Information

We reshape O into $O' \in \mathbb{R}^{(N_{obs} \cdot n) \times d_{hidden}}$. Furthermore, we introduce a set of nodes $q \in \mathbb{R}^{n \times d_{hidden}}$, initialized as $q = \mathbf{0}$.

For the GAT we use the adjacency matrix

$$G_c = \begin{pmatrix} M_n & \mathbf{0}_n & \cdots & \mathbf{0}_n & I_n \\ \mathbf{0}_n & M_n & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_n & \vdots \\ \mathbf{0}_n & \cdots & \mathbf{0}_n & M_n & I_n \\ \mathbf{0}_n & \cdots & \cdots & \mathbf{0}_n & \mathbf{0}_n \end{pmatrix}.$$

We then propagate information on G_c using a Graph Attention Network (GAT) $m_\theta^{(1)}$ consisting of K layers. A single GAT layer updates each node representation by computing attention coefficients over its neighbors. Let $h_i^{(t)}$ denote the representation of node i at layer k. The attention coefficients are computed as

$$\alpha_{ij}^{(k)} = \frac{\exp(e_{ij}^{(k)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(k)})}, \qquad e_{ij}^{(k)} = \text{LeakyReLU}\Big(a^\top [W h_i^{(k)} \| W h_j^{(k)}]\Big),$$

where W and a are learnable parameters. The node representations are then updated according to

$$h_i^{(k+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} W h_j^{(k)}\right).$$

Applying the entire GAT to the initial node representations yields

$$[O', q'] = m_\theta^{(1)}([O', q]).$$

B.4 Mixing Factual Observation

We construct a new graph G'_C and apply a GAT $m_\theta^{(2)}$ consisting of three layers.

The resulting adjacency matrix is

$$G'_C = \begin{pmatrix} \mathbf{0}_n & I_n \\ \mathbf{0}_n & I_n \end{pmatrix}.$$

Applying the GAT three times yields

$$[z', q''] = m_\theta^{(2)}([z, q']).$$

B.5 Estimating Counterfactual Distribution

We build a new graph G''_C with adjacency matrix

$$G''_C = \begin{pmatrix} M_n & I_n \\ I_n & M_n \end{pmatrix}.$$

To propagate information through the entire graph, we use $\min\{\lfloor \frac{n}{2} \rfloor, 5\}$ GAT layers, collected in the MPNN $m_\theta^{(2)}$, which yields

$$[z'', x^{*\prime}] = m_\theta^{(2)}([z', x^*]).$$

Finally, we apply an MLP $m_\theta^{(3)}$ with hidden dimension $3N_c$. After reshaping, the network outputs the parameters $\{\mu_k, \sigma_k, w_k\}_{k=1}^{N_c}$ for each node.

C Experimental Setup

C.1 Model Hyperparameters

For the (semi-)synthetic datasets, we use a hidden dimension of 80, 5 attention heads in the graph attention network (GAT) layers (Brody et al., 2022), and a batch size of 128. For the predicted MoG we use $N_C = 3$ Gaussian mixture components. For the image-based counterfactual datasets, which consist of chest X-rays, we use a hidden dimension of 256, 9 attention heads in the GAT layers, and a batch size of 32. We choose a hidden dimension of 256 to match the dimensionality of the encoded image representations, providing a natural choice for the width of the network.

We train each model for 200 epochs although we include early stopping after 25 epochs. Further, we use the AdamW (Loshchilov & Hutter, 2019) with initial learning rate 10^{-4} and a weight decay of 10^{-5}.

C.2 CheXpert Experiment Setup

We use $N_{obs} = 100$ for the CheXpert samples. For each image, the disease information was represented by the 13 CheXpert findings. We combine them into one `No Finding` attribute. The `Disease` attribute is just the complement. The resulting dataset has some data imbalances. The Sex feature has 41.24% female values and the rest is male. Only 17.24% of samples have PA as value and only 9.05% of the samples exhibit a disease. Finally, the average age in the samples is 60.66 years.

Counterfactual Evaluation. To evaluate which attributes a generated counterfactual image exhibits, we use classifiers/regression models. For the classifiers and regression models of attributes Sex, AP/PA, Age, and Disease we stick closely to Gichoya et al. (2022). We apply the pretrained ResNet (He et al., 2016) and subsequently apply multiple MLP layers. For regression attributes we

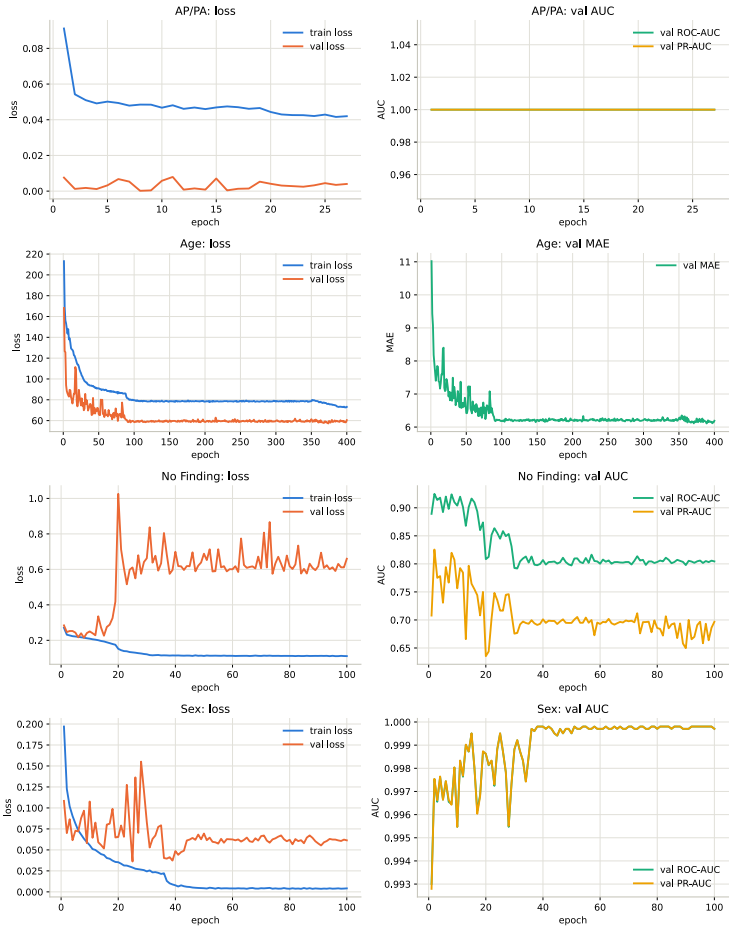

Figure 5: The training curves of the classifiers and regression models used for the classification of the generated CheXpert images.

have final dimension 1 and for the classification attributes we apply the softmax activation function in the end to normalize the outputs to probabilities. We add here the performance of the classification/regression models of the features. The Sex classifier has a test accuracy of 0.9901 and a ROC-AUC of 0.9995. The Age regression model has MAE 6.1473. Further, the AP/PA classifier has both accuracy and ROC-AUC of 1.0. Finally, the `No Finding` classifier has accuracy 0.9010 and ROC-AUC 0.9314. Figure 5 shows the training curves for each of the classifiers.

CheXpert training objective. To the normal classification/regression losses we also add a reconstruction loss to ensure that the images look like actual images. We use MSE, LPIPS(Zhang et al., 2018), and the total variation (Javanmardi et al., 2018) of $\hat{x}_{\text{cf}}$, to get

$$\mathcal{L}_{\text{rec}} = \lambda_M \,\text{MSE}\left(x_{\text{real}}, \hat{x}_{\text{cf}}\right) + \lambda_L \,\text{LPIPS}\left(x_{\text{real}}, \hat{x}_{\text{cf}}\right) + \lambda_{TV} \,\text{TV}\left(\hat{x}_{\text{cf}}\right), \tag{2}$$

with $\lambda_M = 15$, $\lambda_L = 30$, and $\lambda_{TV} = 30$. The resulting total loss combines the classification, regression, and reconstruction objectives:

$$\mathcal{L} = \sum_{i \in \mathcal{C}} \text{CE}\left(y_i, \hat{y}_i\right) + \sum_{r \in \mathcal{R}} \frac{1}{t_r} \,\text{MSE}\left(y_r, \hat{y}_r\right) + \mathcal{L}_{\text{rec}}. \tag{3}$$

where $\mathcal{C}$ and $\mathcal{R}$ denote the sets of classification and regression features, respectively.

C.3 Synthetic Experiments Setup

This appendix gives the complete definitions of the SCMs used in the (semi-) synthetic experiments. Their definitions follow the benchmark specification in Appendix E of Sanchez-Martin et al. (2021). They are reproduced here so that the experimental data-generating processes can be understood without the original paper. We also keep to Sanchez-Martin et al. (2021) for the computation of the standard deviation of the test MSE. We train the model with 5 different seeds and then compute mean and standard deviation of the runs. We use $N_{obs} = 100$ in all synthetic experiments.

C.3.1 Triangle

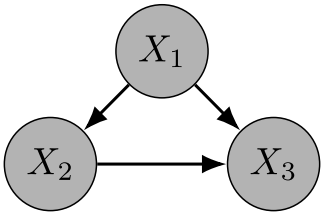

Figure 6: The causal graph of the triangle dataset. The causal graph has three nodes and three edges.

The Triangle SCM has the underlying causal graph as depicted in Figure 6. Hence X_1 affects X_3 through the direct path $X_1 \to X_3$ and through the mediated path $X_1 \to X_2 \to X_3$.

Exogenous distributions. For the LIN and NLIN variants,

$$U_1 \sim \tfrac{1}{2}\mathcal{N}(-2, 1.5) + \tfrac{1}{2}\mathcal{N}(1.5, 1). \tag{4}$$

For NADD,

$$U_1 \sim \tfrac{1}{2}\mathcal{N}(-2.5, 1) + \tfrac{1}{2}\mathcal{N}(2.5, 1). \tag{5}$$

The remaining noises are

$$\text{LIN:} \quad U_2 \sim \mathcal{N}(0,1), \qquad U_3 \sim \mathcal{N}(0,1), \tag{6}$$

$$\text{NLIN:} \quad U_2 \sim \mathcal{N}(0,0.1), \qquad U_3 \sim \mathcal{N}(0,1), \tag{7}$$

$$\text{NADD:} \quad U_2 \sim \mathcal{N}(0,0.25), \qquad U_3 \sim \mathcal{N}(0,0.0625). \tag{8}$$

Here the second parameter of $\mathcal{N}(\mu, \cdot)$ follows the convention used in the benchmark specification.

Linear (LIN).

$$X_1 = U_1, \tag{9}$$

$$X_2 = -X_1 + U_2, \tag{10}$$

$$X_3 = X_1 + 0.25X_2 + U_3. \tag{11}$$

Nonlinear additive (NLIN).

$$X_1 = U_1, \tag{12}$$

$$X_2 = -1 + \frac{3}{1 + \exp(-2X_1)} + U_2, \tag{13}$$

$$X_3 = X_1 + 0.25X_2^2 + U_3. \tag{14}$$

The causal mechanisms are nonlinear, but each exogenous variable enters its structural equation additively.

Nonlinear non-additive (NADD).

$$X_1 = U_1, \tag{15}$$

$$X_2 = 0.25\ \mathrm{sgn}(U_2)\frac{X_1^2}{1 + U_2^2}, \tag{16}$$

$$X_3 = -1 + 0.1\ \mathrm{sgn}(U_3)\left(X_1^2 + X_2^2\right) + U_3. \tag{17}$$

The function $\mathrm{sgn}(\cdot)$ returns the sign of its argument.

C.3.2 M-GRAPH

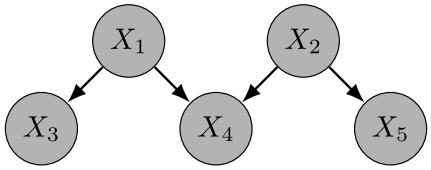

Figure 7: The causal graph of the M-shape data. It has five nodes and four edges.

The M-graph contains five endogenous variables as shown in Figure 7. All exogenous variables are independent standard normal variables:

$$U_i \sim \mathcal{N}(0,1), \qquad i = 1, \ldots, 5. \tag{18}$$

Linear (LIN).

$$X_1 = U_1, \tag{19}$$
$$X_2 = U_2, \tag{20}$$
$$X_3 = X_1 + U_3, \tag{21}$$
$$X_4 = -X_2 + 0.5X_1 + U_4, \tag{22}$$
$$X_5 = -1.5X_2 + U_5. \tag{23}$$

Nonlinear additive (NLIN).

$$X_1 = U_1, \tag{24}$$
$$X_2 = U_2, \tag{25}$$
$$X_3 = X_1 + 0.5X_1^2 + U_3, \tag{26}$$
$$X_4 = -X_2 + 0.5X_1^2 + U_4, \tag{27}$$
$$X_5 = -1.5X_2^2 + U_5. \tag{28}$$

Nonlinear non-additive (NADD).

$$X_1 = U_1, \tag{29}$$
$$X_2 = U_2, \tag{30}$$
$$X_3 = X_1 U_3, \tag{31}$$
$$X_4 = \left(-X_2 + 0.5X_1^2\right) U_4, \tag{32}$$
$$X_5 = -1.5X_2^2 U_5. \tag{33}$$

C.3.3 GERMAN LOAN

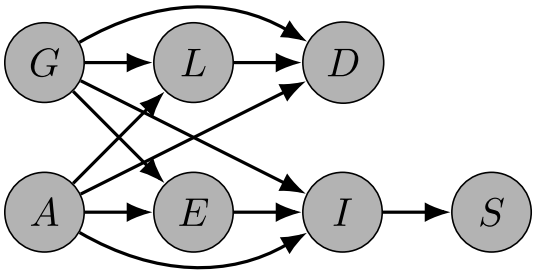

Figure 8: The causal graph of the German loan data. The nodes are real attributes used for German credit rating system. they are connected based on the real-world connections.

The Loan benchmark is a semi-synthetic SCM from the causal-recourse literature and is inspired by the German Credit domain. It contains seven endogenous variables as shown in Figure 8. In the figure, G is gender, A age, E education, L loan amount, D loan duration, I income, and S savings.

The structural assignments are

$$G = U_G, \tag{34}$$
$$A = -35 + U_A, \tag{35}$$
$$E = -0.5 + \Big(1 + \exp\Big[1 - 0.5G - (1 + \exp(-0.1A))^{-1} - U_E\Big]\Big)^{-1}, \tag{36}$$
$$L = 1 + 0.01(A - 5)(5 - A) + G + U_L, \tag{37}$$
$$D = -1 + 0.1A + 2G + L + U_D, \tag{38}$$
$$I = -4 + 0.1(A + 35) + 2G + GE + U_I, \tag{39}$$
$$S = -4 + 1.5\,\mathbb{I}_{\{I>0\}} I + U_S. \tag{40}$$

Here $\mathbb{I}_{\{I>0\}}$ is one if $I > 0$ and zero otherwise.

The exogenous distributions are

$$U_G \sim \text{Bernoulli}(0.5), \tag{41}$$
$$U_A \sim \text{Gamma}(10, 3.5), \tag{42}$$
$$U_E \sim \mathcal{N}(0, 0.25), \tag{43}$$
$$U_L \sim \mathcal{N}(0, 4), \tag{44}$$
$$U_D \sim \mathcal{N}(0, 9), \tag{45}$$
$$U_I \sim \mathcal{N}(0, 4), \tag{46}$$
$$U_S \sim \mathcal{N}(0, 25). \tag{47}$$

C.4 Structural Equations for Artificial Graphs

Variables are generated as $X = f(\text{Pa}(X)) + \epsilon$, where f is sampled from the following set of transformations:

$$\text{Linear:} \quad f_1(\text{Pa}(X)) = a \sum_{X' \in \text{Pa}(X)} X' + b, \qquad a = 0.5,\ b = 0,$$

$$\text{Sum-sqrt:} \quad f_2(\text{Pa}(X)) = a \sum_{X' \in \text{Pa}(X)} \sqrt{|X'|} + b, \qquad a = 0.5,\ b = 0,$$

$$\text{Sum-sine:} \quad f_3(\text{Pa}(X)) = a \sum_{X' \in \text{Pa}(X)} \sin(cX') + b, \qquad a = 1,\ b = 0,\ c = 0.5,$$

$$\text{Sum-tanh:} \quad f_4(\text{Pa}(X)) = a \sum_{X' \in \text{Pa}(X)} \tanh(cX') + b, \qquad a = 1,\ b = 0,\ c = 2,$$

$$\text{Geometric mean:} \quad f_5(\text{Pa}(X)) = a \left(\prod_{X' \in \text{Pa}(X)} |X'| \right)^{\frac{1}{|\text{Pa}(X)|}} + b, \qquad a = 3,\ b = 0.1,$$

$$\text{Log-sum-exp:} \quad f_6(\text{Pa}(X)) = a \log \left(\sum_{X' \in \text{Pa}(X)} e^{X'} \right) + b, \qquad a = 1,\ b = \log 2,$$

$$\text{Sqrt-sum:} \quad f_7(\text{Pa}(X)) = a \sqrt{\left| \sum_{X' \in \text{Pa}(X)} X' \right|} + b, \qquad a = 1,\ b = 0.$$

C.5 Identifiability in a Two-Node SCM

Setup. Let $\boldsymbol{X} = (X_0, X_1)$ follow the SCM

$$X_0 := U_0, \qquad X_1 := aX_0 + \langle \boldsymbol{g}_\lambda(X_0), \boldsymbol{W} \rangle, \tag{48}$$

where $U_0 \sim \mathcal{N}(0,1)$, $\boldsymbol{W} \sim \mathcal{N}(\mathbf{0}, I_2)$ are independent, and

$$s(x) := \sqrt{\sigma_u^2 + \sigma_v^2 x^2}, \qquad \varphi(x) := \arctan(\sigma_v x/\sigma_u), \tag{49}$$

$$\boldsymbol{g}_\lambda(x) := s(x)\big(\cos(\lambda\varphi(x)), \sin(\lambda\varphi(x))\big), \qquad \lambda \in [0,1]. \tag{50}$$

Here a controls the observational mechanism, whereas λ controls the cross-world coupling.

Proposition C.1 (Observational invariance)**.** *For every λ,*

$$X_0 \sim \mathcal{N}(0,1), \qquad X_1 \mid X_0 = x \sim \mathcal{N}\big(ax, s(x)^2\big)\,. \tag{51}$$

Hence the observational distribution is independent of λ. The same is true of the intervention $P_{\mathrm{do}(X_0=\alpha)}$.

Proof. Since

$$\|\boldsymbol{g}_\lambda(x)\|^2 = s(x)^2\big(\cos^2(\lambda\varphi(x)) + \sin^2(\lambda\varphi(x))\big) = s(x)^2,$$

we have $\langle \boldsymbol{g}_\lambda(x), \boldsymbol{W}\rangle \sim \mathcal{N}(0, s(x)^2)$, which gives equation 51. Under $\mathrm{do}(X_0 = \alpha)$,

$$X_1 = a\alpha + \langle \boldsymbol{g}_\lambda(\alpha), \boldsymbol{W}\rangle \sim \mathcal{N}(a\alpha, s(\alpha)^2),$$

again independently of λ. □

Backtracking semantics. Let the factual evidence be

$$e = \{X_0 = x_0,\ X_1 = x_1\}, \qquad r := x_1 - ax_0.$$

To construct a counterfactual world satisfying $X_0' = \alpha$, we use the minimum-change exogenous coupling

$$(U_0', \boldsymbol{W}') = (\alpha, \boldsymbol{W}). \tag{52}$$

Equivalently, equation 52 is the solution of

$$\min_{u_0', \boldsymbol{w}'} c_0|u_0' - U_0|^2 + c_W\|\boldsymbol{w}' - \boldsymbol{W}\|^2 \quad \text{s.t. } u_0' = \alpha,$$

for any $c_0, c_W > 0$. Thus backtracking changes only the exogenous realization necessary to produce $X_0' = \alpha$ while preserving the downstream noise $\boldsymbol{W}$.

Theorem C.2 (Backtracking counterfactual)**.** *Under equation 48–equation 52,*

$$X_1^*(\alpha) \mid e \sim \mathcal{N}\left(a\alpha + k_\lambda r, \tau_\lambda^2\right), \tag{53}$$

where

$$k_\lambda := \frac{s(\alpha)}{s(x_0)}\cos(\lambda\Delta\varphi), \qquad \tau_\lambda^2 := s(\alpha)^2\sin^2(\lambda\Delta\varphi), \tag{54}$$

and

$$\Delta\varphi := \varphi(\alpha) - \varphi(x_0).$$

Moreover, equation 53 is identical to the corresponding abduction–action–prediction counterfactual.

Proof. The evidence fixes $U_0 = x_0$ and imposes

$$\langle \boldsymbol{g}_\lambda(x_0), \boldsymbol{W}\rangle = r.$$

With $\hat{\boldsymbol{g}}_0 := \boldsymbol{g}_\lambda(x_0)/s(x_0)$, Gaussian conditioning gives

$$\boldsymbol{W} \mid e \sim \mathcal{N}\left(\frac{r}{s(x_0)}\hat{\boldsymbol{g}}_0, I - \hat{\boldsymbol{g}}_0\hat{\boldsymbol{g}}_0^\top\right). \tag{55}$$

Backtracking changes U_0 to α while preserving $\boldsymbol{W}$, hence

$$X_1^*(\alpha) = a\alpha + \langle \boldsymbol{g}_\lambda(\alpha), \boldsymbol{W}\rangle.$$

Therefore

$$\mathbb{E}[X_1^*(\alpha) \mid e] = a\alpha + \frac{r}{s(x_0)}\langle \boldsymbol{g}_\lambda(\alpha), \hat{\boldsymbol{g}}_0\rangle,$$

and

$$\mathrm{Var}[X_1^*(\alpha) \mid e] = s(\alpha)^2 - \frac{\langle \boldsymbol{g}_\lambda(\alpha), \boldsymbol{g}_\lambda(x_0)\rangle^2}{s(x_0)^2}.$$

Since

$$\langle \boldsymbol{g}_\lambda(\alpha), \boldsymbol{g}_\lambda(x_0)\rangle = s(\alpha)s(x_0)\cos(\lambda\Delta\varphi),$$

substitution yields equation 53–equation 54. The interventional abduction–action–prediction construction evaluates the same quantity $a\alpha + \langle \boldsymbol{g}_\lambda(\alpha), \boldsymbol{W}\rangle$ under the same conditional law equation 55, so the two counterfactual distributions coincide. □

Corollary 1 (Cross-world non-identifiability)**.** *Although λ is invisible to every observational and interventional distribution, it changes the backtracking counterfactual whenever $\lambda \mapsto (k_\lambda, \tau_\lambda)$ is non-constant. In particular,*

$$\tau_\lambda^2 = 0 \iff \lambda\Delta\varphi \in \pi\mathbb{Z}.$$

For $\lambda = 0$,

$$X_1^*(\alpha) \mid e = \delta_{a\alpha + \frac{s(\alpha)}{s(x_0)}r}, \tag{56}$$

whereas for $\lambda > 0$ the counterfactual has positive conditional variance for almost every (x_0, α) with $\alpha \neq x_0$.

For $\lambda = 1$,

$$\boldsymbol{g}_1(x) = (\sigma_u, \sigma_v x),$$

and

$$k_1 = \frac{\sigma_u^2 + \sigma_v^2 x_0 \alpha}{s(x_0)^2}, \qquad \tau_1^2 = \frac{\sigma_u^2\sigma_v^2(\alpha - x_0)^2}{s(x_0)^2}. \tag{57}$$

Use the prior

$$a \sim \mathcal{N}(0, \sigma_a^2), \qquad \lambda \sim \pi_\lambda, \qquad a \perp \lambda,$$

and let

$$D_{\mathrm{obs}} = \{(x_0^{(m)}, x_1^{(m)})\}_{m=1}^M$$

be observational context data generated from equation 51. Writing

$$\mathcal{E} := D_{\mathrm{obs}} \cup \{(x_0, x_1)\},$$

we get for the optimal backtracking predictive

$$p_{\mathrm{BCM}}(y \mid x_0, x_1, X_0' = \alpha, D_{\mathrm{obs}}) = \int p_{\mathrm{bt}}(y \mid e, \alpha, a, \lambda) p(a, \lambda \mid \mathcal{E})\, \mathrm{d}a\, \mathrm{d}\lambda. \tag{58}$$

Proposition C.3 (Exact posterior and model average)**.** *The likelihood is independent of λ, so*

$$p(a, \lambda \mid \mathcal{E}) = p(a \mid \mathcal{E})\pi_\lambda(\lambda).$$

With

$$w_i := s(x_0^{(i)})^{-2},$$

we have

$$a \mid \mathcal{E} \sim \mathcal{N}\left(\mu_a, \sigma_{a|\mathcal{E}}^2\right), \tag{59}$$

where

$$\sigma_{a|\mathcal{E}}^{-2} = \sigma_a^{-2} + \sum_{i\in\mathcal{E}} w_i (x_0^{(i)})^2, \qquad \mu_a = \sigma_{a|\mathcal{E}}^2 \sum_{i\in\mathcal{E}} w_i x_0^{(i)} x_1^{(i)}. \tag{60}$$

Consequently,

$$p_{\mathrm{BCM}}(y \mid \cdot) = \int \mathcal{N}\left(y; \mu_a\beta_\lambda + k_\lambda x_1, \tau_\lambda^2 + \sigma_{a|\mathcal{E}}^2\beta_\lambda^2\right) \pi_\lambda(\mathrm{d}\lambda), \tag{61}$$

where

$$\beta_\lambda := \alpha - k_\lambda x_0.$$

Proof. Proposition C.1 implies that the likelihood does not depend on λ, which yields the posterior factorization. equation 60 is the Gaussian posterior for weighted linear regression. Finally, substituting $r = x_1 - ax_0$ into equation 53 gives

$$a\alpha + k_\lambda r = a(\alpha - k_\lambda x_0) + k_\lambda x_1 = a\beta_\lambda + k_\lambda x_1.$$

Marginalizing the Gaussian posterior over a gives equation 61. □

Identifiable versus non-identifiable regimes. Let

$$p^*_{\mathrm{BCM}} := p_{\mathrm{bt}}(\cdot \mid e, \alpha, a^*, \lambda^*)$$

denote the backtracking counterfactual under the true mechanism. As $M \to \infty$,

$$\sigma^2_{a|\mathcal{E}} = \Theta(M^{-1}) \to 0, \qquad \mu_a \to a^* \quad \text{a.s.},$$

and therefore

$$p_{\mathrm{BCM}} \Longrightarrow \int \mathcal{N}\left(y; a^*\alpha + k_\lambda r^*, \tau^2_\lambda\right) \pi_\lambda(\mathrm{d}\lambda), \qquad r^* := x_1 - a^* x_0. \tag{62}$$

If $\pi_\lambda = \delta_{\lambda^*}$, the limit in equation 62 equals p^*_{BCM}. If instead π_λ is non-degenerate, then for almost every (x_0, α) with $\alpha \neq x_0$ the map

$$\lambda \mapsto (k_\lambda, \tau_\lambda)$$

is non-constant, so the limiting Bayes predictor remains a genuine mixture over observationally indistinguishable cross-world mechanisms.

Thus even infinite observational data identifies a but not λ: uncertainty about the observational mechanism vanishes, while uncertainty about the backtracking counterfactual need not.

D Additional Results

D.1 Additional CheXpert Results

D.1.1 CheXpert Experiments

We include some counterfactual images generated by the model.

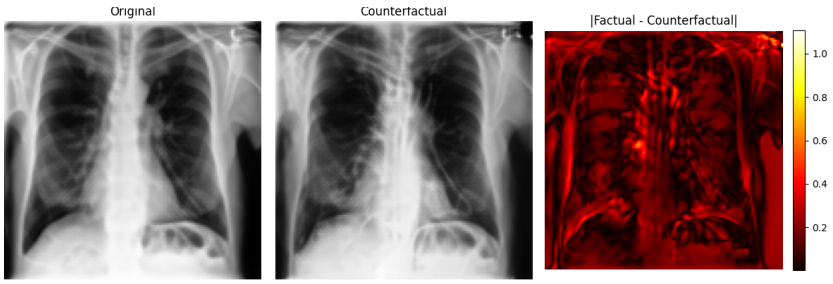

Figure 9: Example counterfactual of a CheXpert image. Observed factual unit (left) has age 80 and the intervention value is 58 in the counterfactual (middle). The difference between the pixels is on the right.

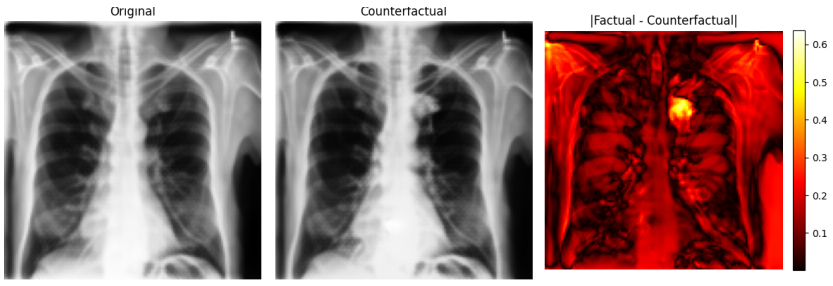

Figure 10: Example counterfactual of a CheXpert image. Observed factual unit (left) has AP/PA as 1 and the intervention value is 0 in the counterfactual (middle). The difference between the pixels is on the right.

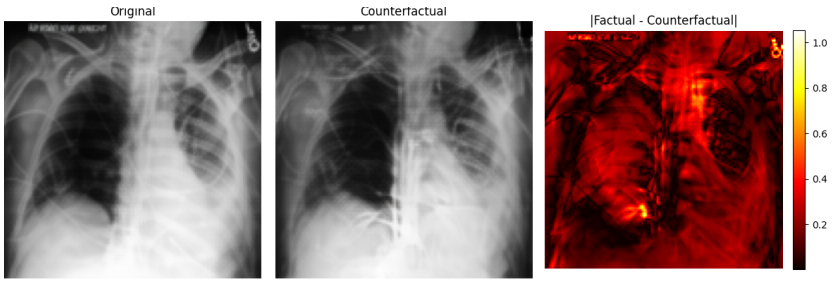

Figure 11: An image where the observed factual unit of age is 74 and the counterfactual intervention is 85.

D.1.2 CheXpert Out-of-Distribution Experiment

This model was trained with sex chosen to be the out-of-distribution feature. However, during inference the sex feature is included again to test the performance.

Out-of-Distribution Feature Sex. We here include the results when the model was trained with OOD feature sex.

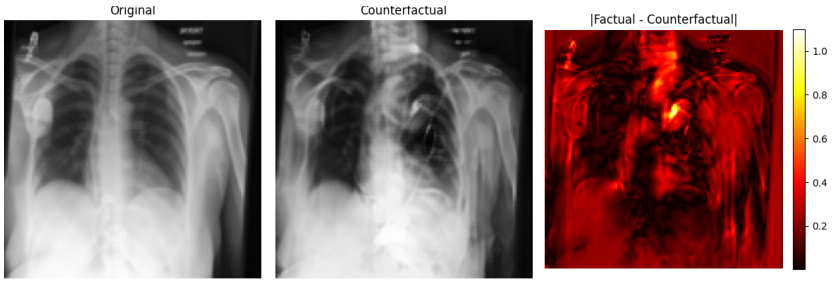

Figure 12: An image where the observed factual unit of age is 65 and the counterfactual intervention is 72. The model was trained with OOD index sex.

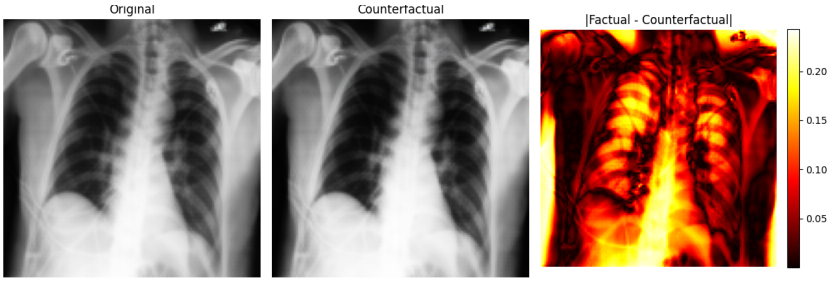

Figure 13: An image where the observed factual unit of AP/PA is 0 and the counterfactual intervention is 1. The model was trained with OOD index sex.

Out-of-Distribution Feature Age. We here include the results when the model was trained with OOD feature age.



Figure 14: An image where the observed factual unit is age 30 and the counterfactual intervention is age 52. The model was trained with OOD index age.

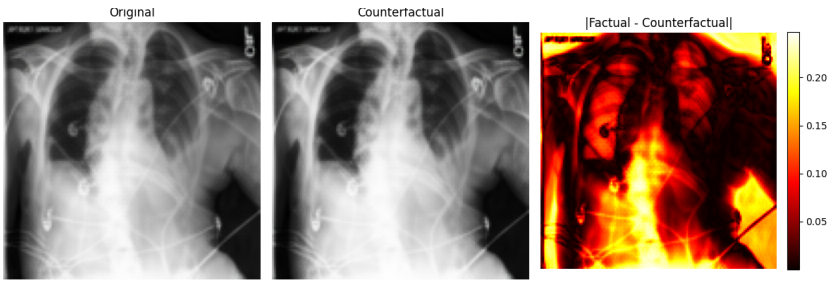

Figure 15: An image where the observed factual unit of sex is 1 and the counterfactual intervention is 0. The model was trained with OOD index age.

Out-of-Distribution Feature AP/PA. We here include the results when the model was trained with OOD feature AP/PA.

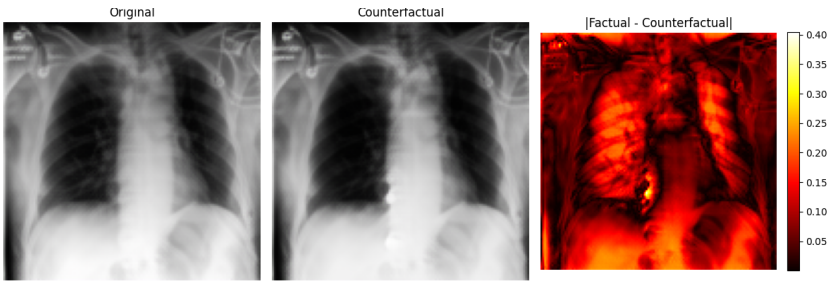

Figure 16: n image where the observed factual unit of sex is 0 and the counterfactual intervention is 1. The model was trained with OOD index AP/PA.

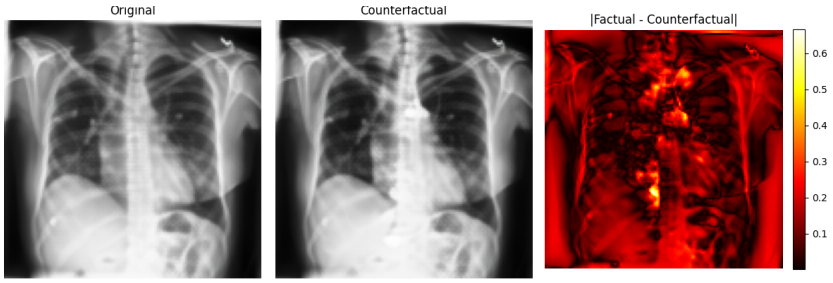

Figure 17: An image where the observed factual unit is age 26 and the counterfactual intervention is age 64. The model was trained with OOD index AP/PA.

Out-of-Distribution Feature Disease. We here include the results when the model was trained with OOD feature disease.

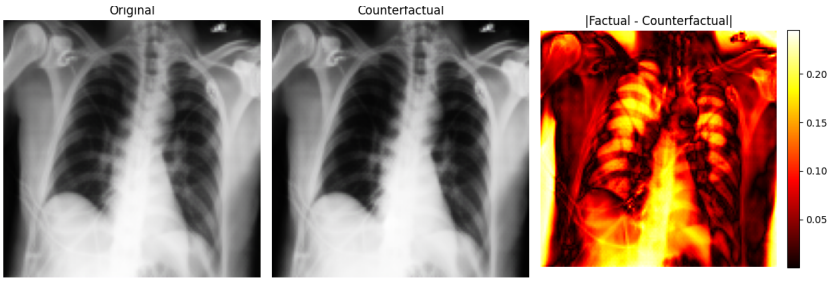

Figure 18: An image where the observed factual unit of AP/PA is 0 and the counterfactual intervention is 1. The model was trained with OOD index disease.

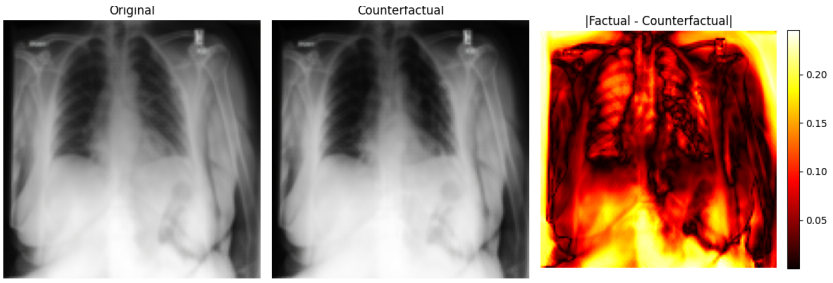

Figure 19: An image where the observed factual unit is age 67 and the counterfactual intervention is age 37. The model was trained with OOD index disease.

D.2 Graph Size Results

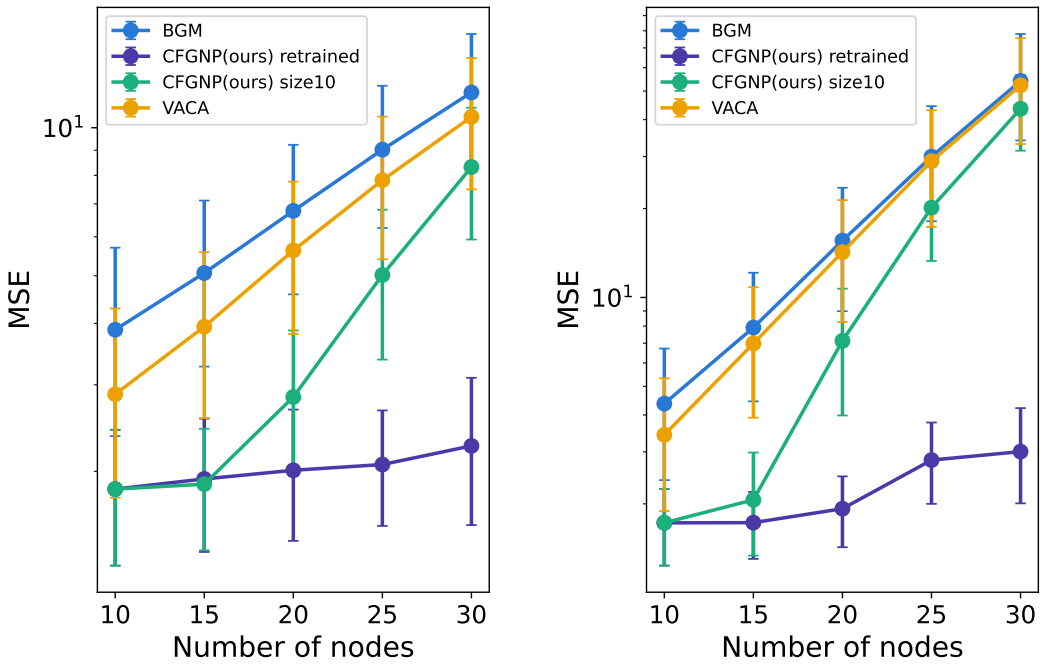

Figure 20: Performance (MSE) of models with regard to the size of the causal graph for sqrt-sum (left) and sum-sqrt (right) structural equations. The MSE is averaged over distinct causal graphs. Vertical lines show the 25th and 75th quantiles.

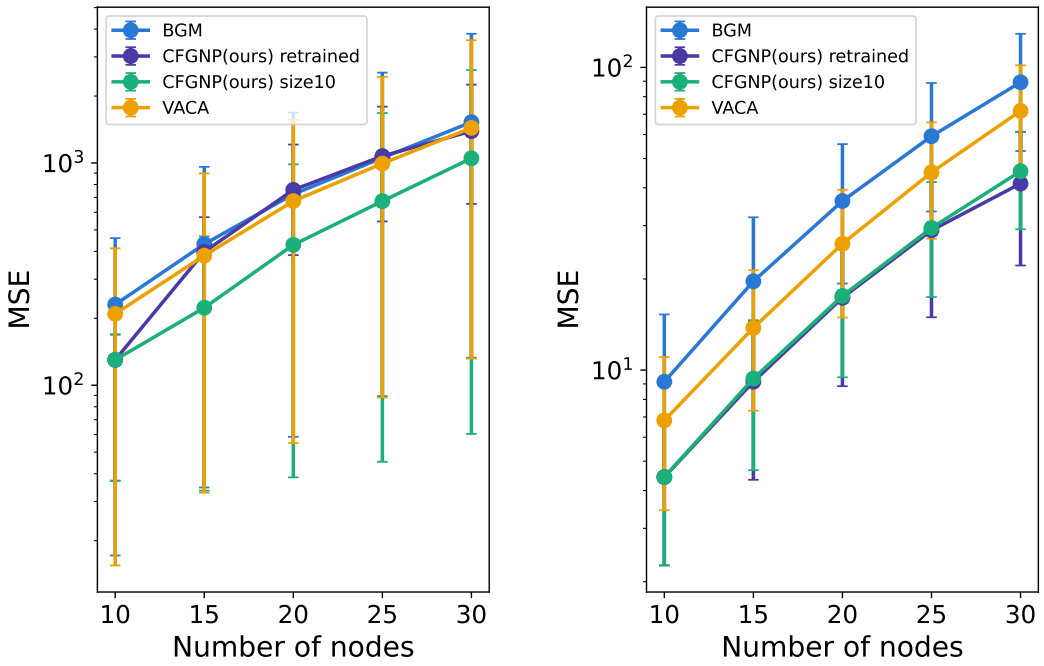

Figure 21: Performance (MSE) of models with regard to the size of the causal graph for geometric-mean (left) and sum-tanh (right) structural equations. The MSE is averaged over distinct causal graphs. Vertical lines show the 25th and 75th quantiles.

D.3 Runtime of Experiments

As before, we use a batch size of 128 for the (semi-)synthetic experiments and a batch size of 32 for the CheXpert dataset. All experiments are run on an NVIDIA H100 GPU.

Dataset	CFGNP (Ours)	BGM	CFM	VACA
German Loan	31.5719s	5.2537s	115.1188s	5.3015s
Triangle LIN	8.1853s	2.9878s	26.3716s	3.7258s
Triangle NLIN	7.9175s	3.0323s	26.2195s	3.7365s
Triangle NADD	7.9084s	3.0364s	26.3470s	3.7963s
M-Shape LIN	12.9088s	3.3865s	41.5352s	3.8361s
M-Shape NLIN	12.9274s	3.3941s	41.8969s	3.8385s
M-Shape NADD	12.9784s	3.4396s	41.9043s	3.8209s
Artificial Size 10	18.4545s	3.8820s	–	2.4226s
Artificial Size 15	27.6381s	5.3534s	–	2.6127s
Artificial Size 20	39.2469s	9.2543s	–	2.6588s
Artificial Size 25	54.7471s	8.5072s	–	2.5591s
Artificial Size 30	74.2878s	13.9569s	–	2.7384s

Table 4: Runtime comparison of CFGNP (Ours), BGM, CFM, and VACA across (semi-) synthetic datasets. The datasets were used throughout the results section.

Dataset	CFGNP (Ours)	VCI	DeepBC
CheXpert	47.7369s	3.8506s	6032.4057s

Table 5: Runtime comparison of CFGNP (Ours), VCI, and DeepBC on the CheXpert dataset.

D.4 Convergence in (Non-)Identifiable Cases

We empirically evaluate the convergence behaviour of CFGNP.

To this end, we evaluate CFGNP on an SCM for which the ground-truth counterfactual distribution can be computed analytically. We consider a causal graph with two nodes and at most one edge. In Appendix C.5, we show that, under certain choices of parameters, such an SCM is identifiable. Furthermore, we analytically compute the ground-truth counterfactual distribution

$$p_T^* := p_T(Y^* = y^* \mid z, T^* = t^*, \mathbf{f}, \mathcal{G}),$$

given the true causal graph and structural mechanisms.

We compare the distribution predicted by CFGNP to both p_T^* and

$$p_T := p_T(Y^* = y^* \mid O, z, T^* = t^*),$$

which conditions only on the observations rather than on knowledge of the SCM $(\mathbf{f}, \mathcal{G})$.

Subsequently, we repeat the experiment for a non-identifiable parameterization of the two-node SCM.

In both cases, we use 40000 samples, split into 21000 train samples, 9000 validation samples, and 10000 test samples.

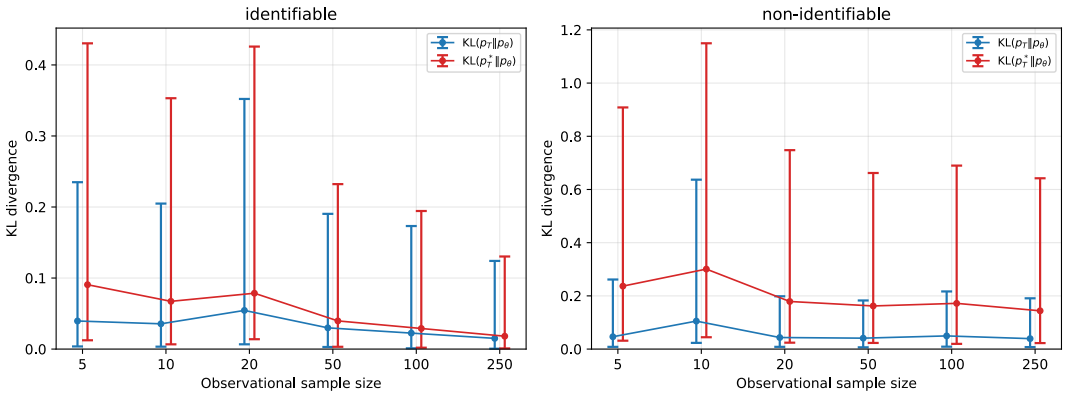

Figure 22: KL divergences for the two-node experiments across different numbers N_{obs} of observations supplied to each sample. Vertical lines show the 10th and 90th quantiles. Results are shown for the identifiable (left) and non-identifiable (right) settings.

D.5 Transformer Ablation

To assess the importance of the CFGNP architecture, we consider a Transformer-based ablation that replaces the core CFGNP processing mechanism with a standard Transformer encoder while retaining the same input and output representations. The observation, factual unit, and counterfactual/intervention inputs are first embedded separately using the same embedding dimensionality as CFGNP. Again, eparate embeddings are retained for intervened and non-intervened variables. Then, we concatenate all these vectors and run a transformer architecture on them. The output of the transformer for x^* is handled in the same way as for CFGNP

Unless stated otherwise, all hyperparameters are kept identical to those used for CFGNP, including the base hidden dimension, number of attention heads, number of mixture components, and input/output dimensionality. As in CFGNP, the effective embedding dimension is given by the base hidden dimension multiplied by the number of attention heads. The Transformer-specific hyperparameters are five encoder layers, a feed-forward dimension of four times the embedding dimension, a dropout rate of 0.1, and GELU activations. Pre-layer normalization is used within each Transformer encoder layer, followed by a final layer normalization after the encoder stack.

Table 6 compares the resulting Transformer ablation against the full CFGNP model.

Dataset	**Variant**	**CFGNP (Ours)**	**Transformer**
		MSE $\downarrow$	
Triangle	LIN	$\mathbf{(2.59 \pm 2.13) \times 10^{-7}}$	$(8.33 \pm 0.25) \times 10^{-4}$
	NLIN	$\mathbf{(2.14 \pm 0.34) \times 10^{-7}}$	$(8.37 \pm 0.46) \times 10^{-4}$
	NADD	$\mathbf{(2.81 \pm 1.11) \times 10^{-7}}$	$(5.02 \pm 0.03) \times 10^{-4}$
M-shape	LIN	$\mathbf{(3.60 \pm 7.00) \times 10^{-5}}$	$(4.79 \pm 0.12) \times 10^{-3}$
	NLIN	$\mathbf{(2.50 \pm 0.47) \times 10^{-6}}$	$(6.04 \pm 0.19) \times 10^{-3}$
	NADD	$\mathbf{(1.08 \pm 0.35) \times 10^{-4}}$	$(3.04 \pm 0.17) \times 10^{-3}$
German Loan	–	$\mathbf{(0.19 \pm 0.09) \times 10^{-2}}$	$(46.92 \pm 7.72) \times 10^{-2}$

Table 6: Mean MSE $\pm$ standard deviation of the full CFGNP model and the Transformer ablation across the Triangle, M-shape, and German Loan datasets. Lower values indicate better performance. Best scores are marked in bold.