Zero-Shot Learning of Causal Models

Divyat Mahajan[*]
Mila, Université de Montréal

Jannes Gladrow
Microsoft Research

Agrin Hilmkil
Microsoft Research

Cheng Zhang
Microsoft Research

Meyer Scetbon[*]
Microsoft Research

Abstract

With the increasing acquisition of datasets over time, we now have access to precise and varied descriptions of the world, capturing all sorts of phenomena. These datasets can be seen as empirical observations of unknown causal generative processes, or Structural Causal Models (SCMs). Recovering these causal generative processes from observations poses formidable challenges, and often require to learn a specific generative model for each dataset. In this work, we propose to learn a *single* model capable of inferring in a zero-shot manner the causal generative processes of datasets. Rather than learning a specific SCM for each dataset, we enable FiP, the architecture proposed in Scetbon et al. [2024], to infer the generative SCMs conditionally on their empirical representations, termed as cond-Fip. We show that cond-FiP is capable of predicting in zero-shot the true generative SCMs, and as a by-product, of (i) generating new dataset samples, and (ii) inferring intervened ones. Our experiments demonstrate that cond-FiP achieves performances on par with SoTA methods trained specifically for each dataset on both in and out-of-distribution problems.

1 Introduction

Learning the causal generative process from observations is a fundamental problem in several scientific domains [Sachs et al., 2005, Foster et al., 2011, Xie et al., 2012], as it offers a comprehensive understanding of the data generation process, and allows for simulating the effect of controlled experiments/interventions. With a learned model of the generative process, one could even accelerate scientific discoveries by reliably predicting the effects of unseen interventions, eliminating the need for laboratory experiments [Ke et al., 2023, Zhang et al., 2024].

A popular approach for modeling causal processes is the structural causal model (SCM) framework [Peters et al., 2017] where causal mechanisms are modeled via structured functional relationships, and causal structures are given by directed acyclic graphs (DAGs). Since in several applications we only have access to observational data, the task of recovering the SCM from observations is an important problem in causality [Pearl, 2009]. Solving this inverse problem is challenging as both the graph and the functional relationships are unknown a priori. Several works have focused on the graph recovery problem by approximating the discrete search space of DAGs [Chickering, 2002, Peters et al., 2014], or using continuous optimization objectives [Zheng et al., 2018, Lachapelle et al., 2019]. However, all these works focus only on the structure learning aspect and do not explicitly evaluate the learning of the functional mechanisms. Another line of work has studied the recovery of the functional relationships, often under structural assumptions like known causal graphs or topological

[*]Equal Contribution. Correspondence to: `t-mscetbon@microsoft.com`.

This work was done when Divyat Mahajan was an intern at Microsoft Research. Cheng Zhang worked on this project when she was affiliated with Microsoft Research and she is currently with Meta.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

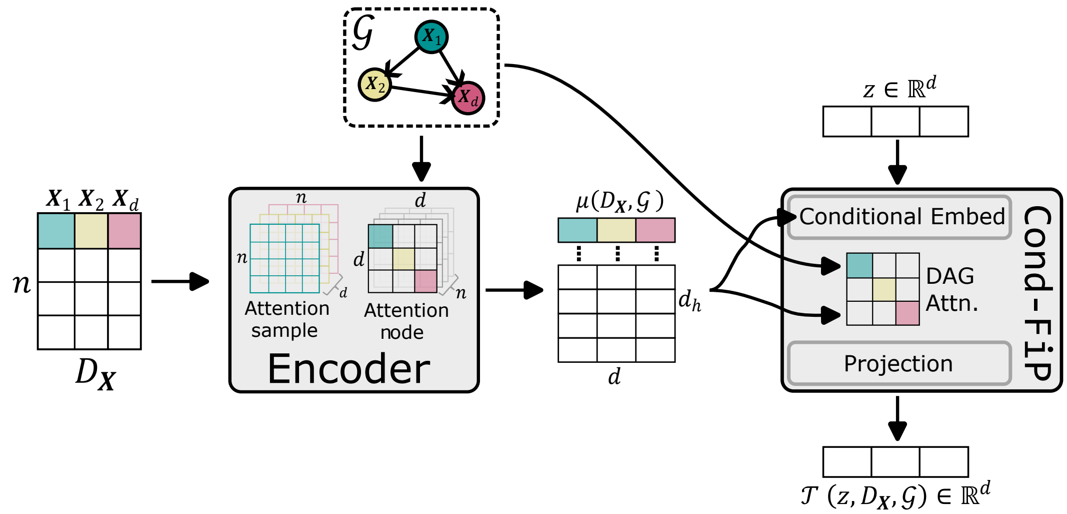


Figure 1: Sketch of the approach proposed in this work. Given a dataset of observations $D_{\boldsymbol{X}}$ and a causal graph $\boldsymbol{\mathcal{G}}$ obtained from an unkown SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F})$, the encoder produces a dataset embedding $\mu(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$, which serves as a condition to instantiate Cond-FiP. Then for any point $\boldsymbol{z} \in \mathbb{R}^d$, $\mathcal{T}(\boldsymbol{z}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ aims at replicating the functional mechanism $\boldsymbol{F}(\boldsymbol{z})$ of the generative SCM.

orders, using maximum likelihood estimation (MLE) independently per node [Blöbaum et al., 2022], autoregressive flows [Khemakhem et al., 2021, Geffner et al., 2022] to model causal generative processes, or transformer-based architectures to directly model SCMs [Scetbon et al., 2024]. Despite these advances in causal learning, a major limitation remains: each new dataset of observations requires training a specific model, which prevents sharing of causal knowledge across datasets.

Amortized learning [Amos, 2022] allows knowledge sharing across datasets through a supervised training objective. Rather than optimizing the parameters of a specific model for each dataset, amortized methods aim at training a *single* model that learn to predict the solutions to various instances of the same optimization problem by exploiting their shared structure. Once trained, such methods enables zero-shot inference (without updating parameters) to new problems at test time. Recent works have proposed techniques for amortized causal structure learning [Lorch et al., 2022, Ke et al., 2022, Scetbon et al., 2024], ATE estimation [Zhang et al., 2023], and model selection for causal discovery [Gupta et al., 2023]. However, none of these works have yet amortized the learning of the functional relationships to directly infer the SCMs.

Contributions. We introduce a conditional version of FiP [Scetbon et al., 2024], termed cond-FiP, that zero-shot predicts the functional mechanisms of SCMs associated with datasets given their causal graphs. As a by-product, cond-FiP can perform zero-shot generation of new samples and simulation of interventions. To achieve this, we enable FiP to be conditioned on dataset embeddings, and propose to amortize its training by minimizing the reconstruction errors of observed data on synthetically generated problems. To obtain the dataset embeddings, we separately train an encoder in an amortized manner to infer the noise associated to observations and use its latent representations in cond-FiP. We show empirically that our method achieves similar performances as SoTA approaches trained specifically for each dataset on both in and out-of-distribution problems. To the best of our knowledge, this is the first time that SCMs are inferred in a zero-shot manner from observations, paving the way for a paradigmatic shift towards the assimilation of causal knowledge across datasets.

2 Background on Causal Learning

Structural Causal Models. An SCM defines the causal generative process of a set of d endogenous (causal) random variables $\boldsymbol{V} = \{X_1, \cdots, X_d\}$, where each causal variable X_i is defined as a function of a subset of other causal variables ($\boldsymbol{V} \setminus \{X_i\}$) and an exogenous noise variable N_i:

$$X_i = F_i(PA(X_i), N_i) \quad \text{s.t.} \;\; PA(X_i) \subset \boldsymbol{V} \;, \;\; X_i \notin PA(X_i) \;. \tag{1}$$

Hence, an SCM describes the data-generation process of $\boldsymbol{X} := [X_1, \cdots, X_d] \sim \mathbb{P}_{\boldsymbol{X}}$ from the noise variables $\boldsymbol{N} := [N_1, \cdots, N_d] \sim \mathbb{P}_{\boldsymbol{N}}$ via the function $\boldsymbol{F} := [F_1, \cdots, F_d]$, and a graph $\boldsymbol{\mathcal{G}} \in \{0,1\}^{d\times d}$ indicating the parents of each variable X_i, that is $[\boldsymbol{\mathcal{G}}]_{i,j} := 1$ if $X_j \in PA(X_i)$. $\boldsymbol{\mathcal{G}}$ is assumed to be a directed acyclic graph (DAG), and we assume assume SCMs to be Markovian (independent noise variables) and denote them as $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F})$. In addition, we only consider additive noise models (ANM), which are SCMs of the form $X_i = F_i(PA(X_i)) + N_i$.

DAG-Attention Mechanism. In FiP [Scetbon et al., 2024] the authors propose to leverage the transformer architecture to learn SCMs from observations. By reparameterizing an SCM according to a topological ordering induced by its graph, the authors propose to formulate an SCM as a fixed-point problem on $\boldsymbol{X}$ of the form $\boldsymbol{X} = \boldsymbol{H}(\boldsymbol{X}, \boldsymbol{N})$ where $\boldsymbol{H}$ admits a simple triangular structure:

$$[\mathrm{Jac}_{\boldsymbol{x}} \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{n})]_{i,j} = 0, \quad \text{if} \quad j \geq i, \quad \text{and} \quad [\mathrm{Jac}_{\boldsymbol{n}} \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{n})]_{i,j} \quad = 0, \quad \text{if} \quad i \neq j.$$

Motivated by this fixed-point reformulation, FiP considers a transformed-based architecture to model the functional relationships of SCMs and propose a new attention mechanism to represent DAGs in a differentiable manner. Recall that the standard attention matrix is defined as:

$$\mathrm{A}_{\boldsymbol{M}}(\boldsymbol{Q}, \boldsymbol{K}) = \frac{\exp((\boldsymbol{Q}\boldsymbol{K}^T - \boldsymbol{M})/\sqrt{d_h})}{\exp((\boldsymbol{Q}\boldsymbol{K}^T - \boldsymbol{M})/\sqrt{d_h})\ \mathbf{1}_d} \tag{2}$$

where $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{d\times d_h}$ denote the keys and queries for a single attention head, and $\boldsymbol{M} \in \{0, +\infty\}^{d\times d}$ is a (potential) mask. When $\boldsymbol{M}$ is chosen to be a triangular mask, the attention mechanism (2) enables to parameterize the effects of previous nodes on the current ones. However, the normalization inherent to the softmax operator prevents effective modeling of root nodes, which are not influenced by any other node in the graph. To alleviate this issue, Scetbon et al. [2024] propose to considering instead:

$$\mathrm{DA}_{\boldsymbol{M}}(\boldsymbol{Q}, \boldsymbol{K}) = \frac{\exp((\boldsymbol{Q}\boldsymbol{K}^T - \boldsymbol{M})/\sqrt{d_h})}{\mathcal{V}\big(\ \exp((\boldsymbol{Q}\boldsymbol{K}^T - \boldsymbol{M})/\sqrt{d_h})\ \mathbf{1}_d\big)} \tag{3}$$

where $\mathcal{V}_i(\boldsymbol{v}) = v_i$ if $v_i \geq 1$, else $\mathcal{V}_i(\boldsymbol{v}) = 1$ for any $\boldsymbol{v} \in \mathbb{R}^d$. While softmax forces the weights along each row of the attention matrix to sum to one, the attention mechanism described in (3) allows the rows to sum in $[0,1]$, thus enabling to modelize root nodes in attention.

3 Conditional FiP

Our approach is composed of two key components: (1) a dataset encoder that generates dataset embeddings based on observations and causal structures, and (2) a conditional variant of FiP, designed to zero-shot infer the generative SCMs of datasets when conditioned on their dataset embeddings produced by the encoder. We first present our dataset encoder and then introduce cond-FiP.

3.1 Dataset Encoder

To obtain dataset embeddings, we propose to train an encoder that predicts in a zero-shot manner the noise samples from their associated observations given the causal structures. We consider the amortized setting at training time, where we have access to empirical representations of K SCMs $\big(\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}^{(k)}, \boldsymbol{\mathcal{G}}^{(k)}, \boldsymbol{F}^{(k)})\big)_{k=1}^K$ that have been sampled independently according to a distribution $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}^{(k)}, \boldsymbol{\mathcal{G}}^{(k)}, \boldsymbol{F}^{(k)}) \sim \mathbb{P}_{\mathcal{S}}$. These empirical representations, denoted $(D_{\boldsymbol{X}}^{(k)}, \boldsymbol{\mathcal{G}}^{(k)})_{k=1}^K$ respectively, contain each n observations $D_{\boldsymbol{X}}^{(k)} := [\boldsymbol{X}_1^{(k)}, \ldots, \boldsymbol{X}_n^{(k)}]^T \in \mathbb{R}^{n\times d}$, and the causal graph $\boldsymbol{\mathcal{G}}^{(k)} \in \{0,1\}^{d\times d}$. At training time, we also require to have access to the associated noise samples $D_{\boldsymbol{N}}^{(k)} := [\boldsymbol{N}_1^{(k)}, \ldots, \boldsymbol{N}_n^{(k)}]^T \in \mathbb{R}^{n\times d}$, which play the role of the target variable in our supervised task. For the sake of clarity, we will omit the dependence on k in our notation and assume access to the full distribution of SCMs $\mathbb{P}_{\mathcal{S}}$. Our goal here is to learn a model that given a dataset of observations $D_{\boldsymbol{X}}$ and the causal graph associated $\boldsymbol{\mathcal{G}}$, recovers the true noise $D_{\boldsymbol{N}}$ from which the observations have been generated. This will provide us with dataset embeddings as well, detailed below.

Encoder Architecture. Following [Lorch et al., 2021], we propose to encode datasets with a transformer-based architecture that alternatively attends on both the sample and node dimensions of the input. More specifically, after having embedded the dataset $D_{\boldsymbol{X}}$ into a higher dimensional space using a linear operation $L(D_{\boldsymbol{X}}) \in \mathbb{R}^{n\times d\times d_h}$ where d_h is the hidden dimension, the encoder E alternates the application of transformer blocks, consisting of a self-attention block followed by an MLP block [Vaswani et al., 2017], where the attention mechanism is applied either across the samples n or the nodes d. When attending over samples, the encoder uses standard self-attention (2) without masking ($\boldsymbol{M} = \{0\}^{n\times n}$), however, when the model attends over the nodes, we leverage the knowledge of the causal graph to mask the undesirable relationships between the nodes, i.e., we set $\boldsymbol{M} = +\infty \times (1 - \boldsymbol{\mathcal{G}})$, with the convention that $0 \times (+\infty) = 0$, in the standard attention (2). Finally, the obtained embeddings $E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}}) \in \mathbb{R}^{n\times d\times d_h}$ are transormed to the original space via prediction network $H : \mathbb{R}^{n\times d\times d_h} \to \mathbb{R}^{n\times d}$ defined as 2-hidden layers MLP.

Training Procedure. To infer the noise samples in a zero-shot manner, we propose to minimize the mean squared error (MSE) of predicting the target noises $D_{\boldsymbol{N}}$ from the input $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ over the distribution of SCMs $\mathbb{P}_{\mathcal{S}}$ available during training:

$$\mathbb{E}_{\mathcal{S}\sim\mathbb{P}_{\mathcal{S}}} ||D_{\boldsymbol{N}} - H \circ E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})||_2^2 \;.$$

Further, as we restrict ourselves to the case of ANMs, we can equivalently reformulate our training objective in order to predict the functional relationships rather than the noise samples. Recall that for an ANM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F})$, we have $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{X} - \boldsymbol{N}$. Hence, with the new targets as $\boldsymbol{F}(D_{\boldsymbol{X}}) := D_{\boldsymbol{X}} - D_{\boldsymbol{N}}$, we can instead to train our encoder to predict the evaluations of the functional relationships over the SCM distribution by minimizing:

$$\mathbb{E}_{\mathcal{S}\sim\mathbb{P}_{\mathcal{S}}} ||\boldsymbol{F}(D_{\boldsymbol{X}}) - H \circ E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})||_2^2 \;.$$

Inference. Given a new dataset $D_{\boldsymbol{X}}$ and its causal graph $\boldsymbol{\mathcal{G}}$, the proposed encoder is able to both provide an embedding $E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}}) \in \mathbb{R}^{n\times d\times d_h}$, and to evaluate the functional mechanisms associated to the current observations $\widehat{F}(D_{\boldsymbol{X}}) := H \circ E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})$. However, this model alone is insufficient for generating new data, which is addressed by the conditional decoder described ahead.

3.2 Cond-FiP: Conditional Fixed-Point Decoder

In this section, we present our proposed approach to infer the functional mechanisms of SCMs in zero-shot via amortized training using synthetically generated datasets. To do so, we propose to extend the formulation of FiP introduced in Scetbon et al. [2024] by enabling it to predict functions, and use the dataset embeddings $E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})$ obtained by our trained encoder as conditions to infer the correct functional mechanisms of the associated SCMs. See Figure 1 for a sketch of Cond-FiP. Analogous to the encoder training setup, we assume access to a distribution of SCMs $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$ at training time, from which we can extract empirical representations of the form $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ containing the observations and the associated causal graphs respectively. Here, we aim at learning a *single* model $\mathcal{T}$ that can infer in zero-shot the functional mechanisms of an SCM given its empirical representation. More formally, we aim at training $\mathcal{T}$ such that for given any dataset $D_{\boldsymbol{X}}$ and its associated causal graph $\boldsymbol{\mathcal{G}}$ obtained from an SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$, the conditional function $\boldsymbol{z} \in \mathbb{R}^d \to \mathcal{T}(\boldsymbol{z}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) \in \mathbb{R}^d$ induced by the model approximates the true functional relationship $\boldsymbol{F} : \boldsymbol{z} \in \mathbb{R}^d \to \boldsymbol{F}(\boldsymbol{z}) \in \mathbb{R}^d$. We achieve this by enabling the FiP to be conditioned on dataset embeddings provided by our dataset encoder, described ahead.

Decoder Architecture. The design of our decoder is based on the FiP architecture for fixed-point SCM learning, with two major differences enabling conditional predictions: (1) we use the dataset embeddings obtained from our encoder as a high dimensional codebook to embed the nodes, and (2) we leverage adaptive layer norm operators [Peebles and Xie, 2023] in the transformer blocks of FiP to enable conditional attention mechanisms. The key change of our decoder compared to the original FiP is in the embedding of the input. In FiP, they proposed to embed a data point $\boldsymbol{z} := [z_1, \ldots, z_d] \in \mathbb{R}^d$ into a high dimensional space using a learnable codebook $\boldsymbol{C} := [C_1, \ldots, C_d]^T \in \mathbb{R}^{d\times d_h}$ and positional embedding $\boldsymbol{P} := [P_1, \ldots, P_d]^T \in \mathbb{R}^{d\times d_h}$, from which they define:

$$\boldsymbol{z}_{\text{emb}} := [z_1 * C_1, \ldots, z_d * C_d]^T + \boldsymbol{P} \in \mathbb{R}^{d\times d_h} \;.$$

By doing so, FiP ensures that the embedded samples admit the same causal structure as the original samples. However, this embedding layer is only adapted if the samples considered are all drawn from the same observational distribution, as the representation of the nodes, that is given by the codebook $\boldsymbol{C}$, is fixed. In order to generalize their embedding strategy to the case where multiple SCMs are considered, we consider conditional codebooks and positional embeddings adapted for each dataset. More formally, given a dataset $D_{\boldsymbol{X}}$ and a causal graph $\boldsymbol{\mathcal{G}}$, we propose to define the conditional codebook and positional embedding associated as

$$\boldsymbol{C}(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) := \mu(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})W_{\boldsymbol{C}}, \text{ and } \boldsymbol{P}(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) := \mu(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})W_{\boldsymbol{P}} \ ,$$

where $\mu(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) := \text{MaxPool}(E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})) \in \mathbb{R}^{d\times d_h}$ is obtained by max-pooling w.r.t the sample dimension the dataset embedding $E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}}) \in \mathbb{R}^{n\times d\times d_h}$ produced by our trained encoder, and $W_{\boldsymbol{C}}, W_{\boldsymbol{P}} \in \mathbb{R}^{d_h\times d_h}$ are learnable parameters. Then we propose to embed any point $\boldsymbol{z} \in \mathbb{R}^d$ conditionally on $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ by considering:

$$\boldsymbol{z}_{\text{emb}} := [z_1 * C_1(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}), \ldots, z_d * C_d(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})]^T + \boldsymbol{P}(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) \in \mathbb{R}^{d\times d_h} \ .$$

Once an input $\boldsymbol{z} \in \mathbb{R}^d$ has been embedded into a higher dimensional space $\boldsymbol{z}_{\text{emb}} \in \mathbb{R}^{d\times d_h}$, FiP models SCMs by simulating the reconstruction of the data from noise. Starting from $\boldsymbol{n}_0 \in \mathbb{R}^{d\times d_h}$ a learnable parameter, they propose to update the current noise $L \geq 1$ times by computing:

$$\boldsymbol{n}_{\ell+1} = h(\text{DA}_{\boldsymbol{M}}(\boldsymbol{n}_\ell, \boldsymbol{z}_{\text{emb}})\boldsymbol{z}_{\text{emb}} + \boldsymbol{n}_\ell)$$

where h refers to the MLP block, and for clarity, we omit both the layer's dependence on its parameters and the inclusion of layer normalization in the notation. Note that FiP considers the DAG-Attention (3) mechanism in order to modelize correctly the root nodes of the SCM. To obtain a conditional formulation of their computational scheme, we propose first to replace the starting noise $\boldsymbol{n}_0$ by a conditional one w.r.t. $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ and defined as $\boldsymbol{n}_0 := \mu(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})W_{\boldsymbol{n}_0} \in \mathbb{R}^{d\times d_h}$, where $W_{\boldsymbol{n}_0} \in \mathbb{R}^{d_h\times d_h}$ is a learnable parameter. To project back the latent representation of $\boldsymbol{z}$ obtained from previous stages, that is $\boldsymbol{n}_L \in \mathbb{R}^{d\times d_h}$, we propose to simply use a linear operation to get $\widehat{\boldsymbol{z}} = \boldsymbol{n}_L W_{\text{out}} \in \mathbb{R}^d$, where $W_{\text{out}} \in \mathbb{R}^{d_h}$ is learnable.

Training Procedure. Recall that our goal is to infer in zero-shot the functional mechanisms of SCMs given their empirical representations. Therefore, to train our model $\mathcal{T}$, we propose to minimize the reconstruction error of the true functional mechanisms estimated by our model over the distribution of SCMs $\mathbb{P}_{\mathcal{S}}$. More precisely, for any SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$ and its empirical representation $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$, we aim at minimizing

$$\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{X}}} \|\mathcal{T}(\boldsymbol{z}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) - \boldsymbol{F}(\boldsymbol{z})\|_2^2 \tag{4}$$

where $\boldsymbol{z} \sim \mathbb{P}_{\boldsymbol{X}}$ is chosen independent of the random dataset $D_{\boldsymbol{X}}$. Therefore, when integrating over the distribution of SCMs, we obtain the following: $\mathbb{E}_{\mathcal{S}\sim\mathbb{P}_{\mathcal{S}}}\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}_{\boldsymbol{X}}} \|\mathcal{T}(\boldsymbol{z}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) - \boldsymbol{F}(\boldsymbol{z})\|_2^2$.

To compute (4), we propose to sample n independent samples $\boldsymbol{X}'_1, \ldots, \boldsymbol{X}'_n$ from $\mathbb{P}_{\boldsymbol{X}}$, leading to a new dataset $D_{\boldsymbol{X}'}$ independent of $D_{\boldsymbol{X}}$, from which we obtain the following optimization problem:

$$\mathbb{E}_{\mathcal{S}\sim\mathbb{P}_{\mathcal{S}}} \|\mathcal{T}(D_{\boldsymbol{X}'}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) - \boldsymbol{F}(D_{\boldsymbol{X}'})\|_2^2 \ .$$

Therefore our training objective aims at learning $\mathcal{T}$ such that for any given empirical representation $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ of an unknown SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$, the conditional function induced by our model, that is $\boldsymbol{z} \to \mathcal{T}(\boldsymbol{z}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$, is close to the true functional mechanism $\boldsymbol{F}$ in the MSE sense. Appendix B provides more details on how to use cond-FiP for causal generation and inference tasks.

4 Experiments

4.1 Experimental Setup

Data Generation Process. We use the synthetic data generation procedure proposed by Lorch et al. [2022] to generate SCMs in our empirical study, as it offers a wide variety of SCMs, making it ideal for amortized training. We have the option to sample graphs from various schemes and noise variables from diverse distributions. Further, we can control the complexity of causal relationships: either we set them to be linear (LIN) functions randomly sampled, or use random fourier features (RFF) for generating random non-linear causal relationships. We construct two distribution of SCMs

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
DoWhy	10	0.03 (0.0)	0.13 (0.02)	0.04 (0.01)	0.11 (0.01)
DECI	10	0.09 (0.01)	0.23 (0.03)	0.12 (0.01)	0.23 (0.03)
FiP	10	0.04 (0.0)	0.09 (0.01)	0.06 (0.01)	0.08 (0.01)
Cond-FiP	10	0.06 (0.01)	0.1 (0.01)	0.07 (0.01)	0.1 (0.01)
DoWhy	20	0.03 (0.01)	0.15 (0.02)	0.03 (0.0)	0.23 (0.01)
DECI	20	0.10 (0.02)	0.21 (0.03)	0.08 (0.02)	0.23 (0.02)
FiP	20	0.04 (0.0)	0.12 (0.02)	0.05 (0.0)	0.15 (0.02)
Cond-FiP	20	0.06 (0.01)	0.09 (0.01)	0.07 (0.0)	0.12 (0.0)
DoWhy	50	0.03 (0.0)	0.18 (0.03)	0.03 (0.0)	0.29 (0.03)
DECI	50	0.09 (0.01)	0.24 (0.02)	0.07 (0.01)	0.29 (0.02)
FiP	50	0.04 (0.0)	0.14 (0.03)	0.04 (0.0)	0.23 (0.04)
Cond-FiP	50	0.06 (0.01)	0.10 (0.01)	0.07 (0.01)	0.14 (0.01)
DoWhy	100	0.03 (0.0)	0.2 (0.03)	0.03 (0.0)	0.31 (0.02)
DECI	100	0.08 (0.02)	0.26 (0.03)	0.07 (0.01)	0.30 (0.02)
FiP	100	0.04 (0.0)	0.16 (0.03)	0.04 (0.0)	0.24 (0.02)
Cond-FiP	100	0.05 (0.0)	0.1 (0.01)	0.07 (0.01)	0.16 (0.01)

Table 1: **Results for Noise Prediction.** Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario. Shaded rows denote the case where the graph size is larger than the train graph sizes ($d = 20$) for Cond-FiP.

$\mathbb{P}_{\text{IN}}$, and $\mathbb{P}_{\text{OUT}}$, which vary based on the choice for sampling causal graphs, noise variables, and causal relationships. Please refer to Appendix C.1 for more details.

Training Datasets. We randomly sample SCMs from the $\mathbb{P}_{\text{IN}}$ distribution, and we restrict the total nodes of each SCM to be $d = 20$ nodes. From each of these SCMs, we extract the causal graph $\mathcal{G}$ and further generate $n_{\text{train}} = 400$ observations to obtain $D_{\boldsymbol{X}}$. This procedure is used for generating the training datasets for both amortized training of the dataset encoder and Cond-FiP.

Test Datasets. We evaluate the model for both in-distribution and out-of-distribution generalization by sampling datasets from $\mathbb{P}_{\text{IN}}$ and $\mathbb{P}_{\text{OUT}}$ respectively. Our test datasets are categorized as follows: LIN **IN** and RFF **IN** where the SCM are sampled from $\mathbb{P}_{\text{IN}}$ with linear and non-linear functional relationships respectively. Similarly, we define LIN **OUT** and RFF **OUT** where the SCMs are sampled from $\mathbb{P}_{\text{OUT}}$ instead. For each category, we vary total nodes $d \in [10, 20, 50, 100]$ and sample for each dimension d either 6 or 9 SCMs, depending on the number of possible schemes for sampling the causal graphs, from which we generate $n_{\text{test}} = 800$ observational samples. Hence, we have a total of 120 test datasets, allowing for a comprehensive evaluation of methods.

4.2 Benchmark of Cond-FiP

We compare Cond-FiP against non-amortized baselines FiP [Scetbon et al., 2024], DECI [Geffner et al., 2022], and DoWhy [Blöbaum et al., 2022] that are trained from scratch on each test dataset. For a fair comparison, we use $n_{\text{train}} = 400$ samples to train baselines, and evaluate the performance on the remaining 400 test samples. Also, we provide the true graph $\mathcal{G}$ to all the baselines for consistency. Finally, we use 400 sample to obtain the dataset embedding, and evaluate Cond-FiP on the remaining ones. Please refer to Appendix C.2 for further implementation details.

We evaluate the performance of all the methods on the following three tasks.

- **Noise Prediction:** given the observations $D_{\boldsymbol{X}}$ and the true graph $\mathcal{G}$, infer the noise variables $\widehat{D_{\boldsymbol{N}}}$ and compute the root-mean-square error (RMSE) (Appendix C.3) with true noise $D_{\boldsymbol{N}}$.
- **Sample Generation:** given the noise samples $D_{\boldsymbol{N}}$ and the true graph $\mathcal{G}$, generate the causal variables $\widehat{D_{\boldsymbol{X}}}$ and compute RMSE with the true causal variables $D_{\boldsymbol{X}}$.
- **Interventional Generation:** Given the noise samples $D_{\boldsymbol{N}}$ and true graph $\mathcal{G}$, generate the intervened samples and compute RMSE with the true intervened samples.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
DoWhy	10	0.05 (0.0)	0.18 (0.03)	0.06 (0.01)	0.12 (0.02)
DECI	10	0.15 (0.02)	0.33 (0.04)	0.16 (0.02)	0.27 (0.03)
FiP	10	0.07 (0.0)	0.13 (0.02)	0.08 (0.01)	0.11 (0.02)
Cond-FiP	10	0.06 (0.01)	0.14 (0.02)	0.05 (0.01)	0.08 (0.01)
DoWhy	20	0.06 (0.01)	0.27 (0.05)	0.05 (0.0)	0.39 (0.04)
DECI	20	0.16 (0.02)	0.39 (0.05)	0.13 (0.02)	0.44 (0.04)
FiP	20	0.08 (0.01)	0.23 (0.05)	0.08 (0.01)	0.27 (0.04)
Cond-FiP	20	0.05 (0.01)	0.24 (0.06)	0.07 (0.01)	0.30 (0.03)
DoWhy	50	0.08 (0.01)	0.35 (0.09)	0.06 (0.01)	0.54 (0.06)
DECI	50	0.15 (0.01)	0.46 (0.06)	0.13 (0.02)	0.67 (0.06)
FiP	50	0.09 (0.01)	0.26 (0.05)	0.08 (0.01)	0.48 (0.06)
Cond-FiP	50	0.08 (0.01)	0.25 (0.05)	0.07 (0.0)	0.48 (0.07)
DoWhy	100	0.06 (0.0)	0.33 (0.07)	0.06 (0.01)	0.63 (0.07)
DECI	100	0.14 (0.02)	0.50 (0.09)	0.14 (0.02)	0.71 (0.08)
FiP	100	0.08 (0.01)	0.3 (0.06)	0.09 (0.01)	0.55 (0.08)
Cond-FiP	100	0.07 (0.01)	0.29 (0.07)	0.09 (0.01)	0.57 (0.07)

Table 2: **Results for Sample Generation.** Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario. Shaded rows denote the case where the graph size is larger than the train graph sizes ($d = 20$) for Cond-FiP.

Results on Noise Predictions. Table 1 presents the results for the case of inferring noise from observations. Across all the different in-distribution and out-of-distribution scenarios, Cond-FiP is competitive with the baselines that were trained from scratch at test time. Further, Cond-FiP is able to generalize to larger graphs ($d = 50, d = 100$) despite being trained for only graphs of size $d = 20$.

Results on Generation. Table 2 presents results for generating observational data, and shows that Cond-FiP is competitive with the baselines across all the scenarios. Similar to the case of noise prediction, Cond-FiP can generalize to larger graphs at test time! Further, Cond-FiP can also simulate interventional data while being robust to distribution shifts and graph sizes (Table 3). This is especially interesting as we never explicitly trained Cond-FiP for interventional tasks. This provide further evidence towards Cond-FiP capturing the true functional mechanisms.

We also obtain similar findings with the CSuite benchmark (Figure 2), which is a different simulator than what we used for training Cond-FiP. We also add a real-world experiment in Appendix E to benchmark Cond-Fip. Further, Appendix F provides results for ablations of the decoder of Cond-FiP, and Appendix G provides results on Cond-FiP's ability to generalize to larger sample size.

5 Conclusion.

In this work, we demonstrate that a single model can be trained to infer Structural Causal Models (SCMs) in a zero-shot manner through amortized training. Our proposed method, Cond-FiP, not only generalizes effectively to novel SCMs at test time but also remains robust across varying SCM distributions. To our knowledge, this is the first approach to establish the feasibility of learning causal generative models in a foundational manner. Future work will focus on scaling to larger problem instances and applying the method to real-world scenarios.

References

Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv e-prints*, pages arXiv–2202, 2022.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

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

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

P Erdos and A Renyi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Jared C Foster, Jeremy MG Taylor, and Stephen J Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in medicine*, 30(24):2867–2880, 2011.

Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Shantanu Gupta, Cheng Zhang, and Agrin Hilmkil. Learned causal method prediction. *arXiv preprint arXiv:2311.03989*, 2023.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Adrián Javaloy, Pablo Sanchez-Martin, and Isabel Valera. Causal normalizing flows: from theory to practice. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. *ArXiv*, abs/2312.02696, 2023. URL `https://api.semanticscholar.org/CorpusID:265659032`.

Nan Rosemary Ke, Silvia Chiappa, Jane Wang, Anirudh Goyal, Jorg Bornschein, Melanie Rey, Theophane Weber, Matthew Botvinic, Michael Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. *arXiv preprint arXiv:2204.04875*, 2022.

Nan Rosemary Ke, Sara-Jane Dunn, Jorg Bornschein, Silvia Chiappa, Melanie Rey, Jean-Baptiste Lespiau, Albin Cassirer, Jane Wang, Theophane Weber, David Barrett, Matthew Botvinick, Anirudh Goyal, Mike Mozer, and Danilo Rezende. Discogen: Learning to discover gene regulatory networks, 2023.

Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *International Conference on Artificial Intelligence and Statistics*, pages 3520–3528. PMLR, 2021.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. *Advances in Neural Information Processing Systems*, 35: 13104–13118, 2022.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 2014.

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

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Meyer Scetbon, Joel Jennings, Agrin Hilmkil, Cheng Zhang, and Chao Ma. Fip: a fixed-point approach for causal generative modeling, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Menghua Wu, Yujia Bao, Regina Barzilay, and Tommi Jaakkola. Sample, estimate, aggregate: A recipe for causal discovery foundation models. *arXiv preprint arXiv:2402.01929*, 2024.

Yu Xie, Jennie E Brand, and Ben Jann. Estimating heterogeneous treatment effects with observational data. *Sociological methodology*, 42(1):314–347, 2012.

Jiaqi Zhang, Joel Jennings, Cheng Zhang, and Chao Ma. Towards causal foundation model: on duality between causal inference and attention. *arXiv preprint arXiv:2310.00809*, 2023.

Jiaqi Zhang, Kristjan Greenewald, Chandler Squires, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. *Advances in Neural Information Processing Systems*, 36, 2024.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/e347c51419ffb23ca3fd5050202f9c3d-Paper.pdf`.

A Related Works

Amortized Causal Learning. Amortized methods have been explored in causality research in order to learn general algorithms that can infer in a zero-shot manner causal knowledge from observations. [Zhang et al., 2023] proposed to amortize the learning of a causal inference method to estimate average treatment effect (ATE), [Gupta et al., 2023] used amortized learning to perform model selection for causal discovery, and [Lorch et al., 2022, Ke et al., 2022, Wu et al., 2024, Scetbon et al., 2024] proposed to infer causal structures, such as causal graphs or topological orders, from observations using amortized learning. All these methods rely on the availability of synthetic datasets generated during training, enabling their learning using supervised objectives to predict the causal knowledge of interest. In this work, we extend this line of works, and propose to infer in a zero-shot manner the functional relationships of SCMs from observations and their associated causal structures. To achieve this, we propose to amortize the learning of causal embeddings of synthetically generated datasets, which then served as conditions to train a generalized version of FiP [Scetbon et al., 2024] that infers the generative SCMs in zero-shot.

Autoregressive Causal Learning. While a vast majority of the literature on causal discovery concerns structure learning, recent works on causal autoregressive flows [Khemakhem et al., 2021, Javaloy et al., 2023] focus on state-of-the-art generative modeling techniques for learning the causal generative processes induced by SCMs. Khemakhem et al. [2021] proved a novel connection between SCMs and autoregressive flows, as the mapping from noise variables to observable variables in SCMs is a triangular map given the topological order of the causal graph. While their work restricted the functional relationships to additive and affine flows, this was extended by Javaloy et al. [2023] to more flexible triangular monotonic increasing maps. More recently, Scetbon et al. [2024] proposed to directly model SCMs, viewed as fixed-point problems on the ordered nodes, using transformer-based architectures. While these methods enable efficient learning of SCMs and their generative processes, they all require to train a specific generative model per dataset. In contrast, we present a novel extension of FiP [Scetbon et al., 2024] by conditioning the fixed-point process on dataset embeddings, thereby amortizing the learning of functional relationships across different instances from the functional class of SCMs.

B Inference with Cond-FiP: Generation and Intervention

Generation with Cond-FiP. Given a random vector noise $\boldsymbol{n} \sim \mathbb{P}_{\boldsymbol{N}}$, we can estimate the observational sample associated according to an unknown SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$ as long as we have access to its empirical representation $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$. More formally, starting from $\boldsymbol{n}_0 = \boldsymbol{n}$, we infer the associated observation by computing for $\ell = 1, \dots, d$:

$$\boldsymbol{n}_\ell = \mathcal{T}(\boldsymbol{n}_{\ell-1}, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}}) + \boldsymbol{n} \ . \tag{5}$$

After (at most) d iterations, $\boldsymbol{n}_d$ corresponds to the observational sample associated to the original noise $\boldsymbol{n}$ according to our conditional SCM $\mathcal{T}(\cdot, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$. To sample noise from $\mathbb{P}_{\boldsymbol{N}}$, we leverage cond-FiP that can estimates noise samples under the ANM assumption by computing $\widehat{D_{\boldsymbol{N}}} := D_{\boldsymbol{X}} - \mathcal{T}(D_{\boldsymbol{X}}, E(L(D_{\boldsymbol{X}}), \boldsymbol{\mathcal{G}})$. From these estimated noise samples, we can efficiently estimate the joint distribution of the noise thanks to the Markovian assumption by computing the inverse cdfs of the marginals as proposed in Scetbon et al. [2024].

Interventional Predictions. Cond-FiP also enables the estimation of interventions given an empirical representation $(D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ of an unkown SCM $\mathcal{S}(\mathbb{P}_{\boldsymbol{N}}, \boldsymbol{\mathcal{G}}, \boldsymbol{F}) \sim \mathbb{P}_{\mathcal{S}}$. To achieve this, we start from a noise sample $\boldsymbol{n}$, and we generate the associated intervened sample $\widehat{\boldsymbol{z}}^{\text{do}}$ by directly modifying the conditional SCM provided by Cond-FiP. More specifically, we modify in place the SCM obtained by Cond-FiP, leading to its interventional version $\mathcal{T}^{\text{do}}(\cdot, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$. Now, generating an intervened sample can be done by applying the loop defined in (5), starting from $\boldsymbol{n}$ and using the intervened SCM $\mathcal{T}^{\text{do}}(\cdot, D_{\boldsymbol{X}}, \boldsymbol{\mathcal{G}})$ rather than the original one.

C Experiment Setup Details

C.1 Data Generation Process

- **In Distribution ($\mathbb{P}_{\text{IN}}$):** We sample causal graphs using the Erods-Renyi [Erdos and Renyi, 1959] and scale-free models [Barabási and Albert, 1999] schemes. Noise variables are sampled from the gaussian distribution, and we allow for both LIN and RFF causal relationships.
- **Out of Distribution ($\mathbb{P}_{\text{OUT}}$):** Causal graphs are drawn from Watts-Strogatz [Watts and Strogatz, 1998] and stochastic block models [Holland et al., 1983] schemes. Noise variables follow the laplace distribution, and both the LIN and RFF cases are used to sample functions. However, the parameters of these distributions are sampled from a different range as compared to $\mathbb{P}_{\text{IN}}$ to create a distribution shift.

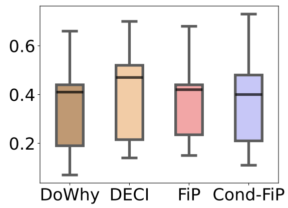
C.2 Cond-FiP Model and Training Configuration

For both the dataset encoder and cond-FiP, we set the embedding dimension to $d_h = 256$ and the hidden dimension of MLP blocks to 512. Both of our transformer-based models contains 4 attention layers and each attention consists of 8 attention heads. The models were trained for a total of $10k$ epochs with the Adam optimizer [Paszke et al., 2017], where we used a learning rate of $1e-4$ and a weight decay of $5e-9$. We also use the EMA implementation of [Karras et al., 2023] to train our models. Each epoch contains $\simeq 400$ randomly generated datasets from the distribution $\mathbb{P}_{\text{IN}}$, which are processed with a batch size of 2 on a single L40 GPU with 48GB memory.

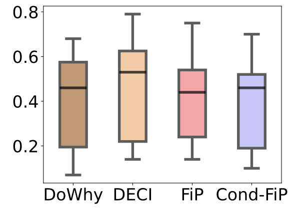
C.3 Evaluation Metric

Let us denote a predicted target as $\widehat{\boldsymbol{Y}} \in \mathbb{R}^{n_{\text{test}} \times d}$ and the true target as $\boldsymbol{Y} \in \mathbb{R}^{n_{\text{test}} \times d}$. The RMSE is computed on a sample basis and then averaged over all the test samples available. More formally, the metric used here is $\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \sqrt{\frac{1}{d} \| [\boldsymbol{Y}]_i - [\widehat{\boldsymbol{Y}}]_i \|_2^2}$. This metric allows us to compare results across different graph sizes as it is scaled by the dimension d.

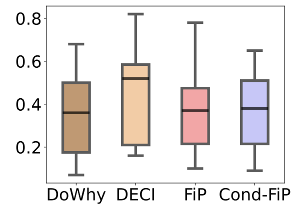
D Additional Results: Interventional Generation



(a) Noise Prediction



(b) Sample Generation



(c) Interventional Generation

Figure 2: We compare Cond-FiP against the baselines for the different evaluation tasks on the **CSuite benchmark**. The y-axis denotes the RMSE for the respective tasks across the 9 datasets.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
`DoWhy`	10	0.08 (0.03)	0.19 (0.04)	0.05 (0.01)	0.12 (0.02)
`DECI`	10	0.17 (0.02)	0.34 (0.04)	0.13 (0.02)	0.25 (0.03)
`FiP`	10	0.08 (0.01)	0.15 (0.02)	0.07 (0.01)	0.09 (0.01)
`Cond-FiP`	10	0.10 (0.03)	0.21 (0.03)	0.07 (0.01)	0.11 (0.01)
`DoWhy`	20	0.06 (0.01)	0.27 (0.06)	0.05 (0.0)	0.36 (0.03)
`DECI`	20	0.16 (0.02)	0.38 (0.05)	0.15 (0.04)	0.42 (0.03)
`FiP`	20	0.09 (0.01)	0.23 (0.05)	0.12 (0.04)	0.25 (0.03)
`Cond-FiP`	20	0.09 (0.01)	0.24 (0.05)	0.14 (0.03)	0.31 (0.03)
`DoWhy`	50	0.08 (0.01)	0.29 (0.05)	0.06 (0.01)	0.53 (0.06)
`DECI`	50	0.17 (0.02)	0.44 (0.06)	0.13 (0.02)	0.64 (0.06)
`FiP`	50	0.11 (0.02)	0.25 (0.05)	0.09 (0.01)	0.46 (0.06)
`Cond-FiP`	50	0.13 (0.02)	0.27 (0.04)	0.12 (0.02)	0.48 (0.07)
`DoWhy`	100	0.05 (0.0)	0.33 (0.07)	0.06 (0.01)	0.60 (0.07)
`DECI`	100	0.14 (0.02)	0.49 (0.08)	0.15 (0.02)	0.70 (0.08)
`FiP`	100	0.08 (0.01)	0.29 (0.07)	0.10 (0.01)	0.54 (0.08)
`Cond-FiP`	100	0.10 (0.01)	0.30 (0.06)	0.14 (0.02)	0.58 (0.07)

Table 3: **Results for Interventional Generation.** We compare Cond-FiP against the baselines for the task of generating interventional data from the input noise variables. Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario. Shaded rows denote the case where the graph size is larger than the train graph sizes ($d = 20$) for Cond-FiP.

E Experiments on Real World Benchmark

We use the real world flow cytometry dataset [Sachs et al., 2005] to benchmark Cond-FiP againts the baselines. This dataset contains $n \simeq 800$ observational samples expressed in a $d = 11$ dimensional space, and the reference (true) causal graph. We sample a train dataset $D_{\boldsymbol{X}}^{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times d}$ and test dataset $D_{\boldsymbol{X}}^{\text{test}} \in \mathbb{R}^{n_{\text{test}} \times d}$ of size $n_{\text{train}} = n_{\text{test}} = 400$ each, where the train dataset is to used to train the baselines and obtain dataset embedding for Cond-FiP.

Since we don't have access to the true functional relationships, we cannot compute RMSE for noise prediction or sample generation like we did in our experiments with synthetic benchmarks. Instead for each method, we obtain the noise predictions $\widehat{D_{\boldsymbol{N}}^{\text{train}}}$ on the train split, and use it to fit a gaussian distribution for each component (node). Then we use the learned gaussian distribution to sample new noise variables, $\widehat{D_{\boldsymbol{N}}^{\text{sample}}}$, which are mapped to the observations as per the causal mechanisms learned by each method, $\widehat{D_{\boldsymbol{X}}^{\text{sample}}}$. Finally, we compute the maximum mean discrepancy (MMD) distance between $\widehat{D_{\boldsymbol{X}}^{\text{sample}}}$ and $D_{\boldsymbol{X}}^{\text{test}}$ as metric to determine whether the method has captured the true causal relationships. For consistency, we also evaluate the reconstruction performances of the models by using directly the inferred noise $\widehat{D_{\boldsymbol{N}}^{\text{train}}}$ from the models, and the compute MMD between the reconstructed data and the test data.

Table 4 presents our results, where for reference we also report the MMD distance between the true train and test split, which should be very small since both the datasets are sampled from the same distribution. We find that Cond-FiP is competitive with the baselines that were trained from scratch. Except DoWhy, the MMD distance with reconstructed samples from the methods are close to oracle performance.

Method	MMD($\widehat{D_{\boldsymbol{X}}^{\text{sample}}}$, $D_{\boldsymbol{X}}^{\text{test}}$)	MMD($\widehat{D_{\boldsymbol{X}}^{\text{train}}}$, $D_{\boldsymbol{X}}^{\text{test}}$)	MMD($D_{\boldsymbol{X}}^{\text{train}}$, $D_{\boldsymbol{X}}^{\text{test}}$)
`DoWhy`	0.015	0.014	0.005
`DECI`	0.014	0.005	0.005
`FiP`	0.015	0.005	0.005
`Cond-FiP`	0.013	0.005	0.005

Table 4: **Results for Sachs dataset.** We compare Cond-FiP against the baselines for the task of generating sample data on the real world benchmark. Each cell reports the MMD, and we also report the reconstruction error for all of the methods.

F Ablation Study

We conduct an ablation study where we train two variants of the decoder Cond-FiP described as follows:

- *Cond-FiP (LIN):* We sample SCMs with linear functional relationships during training.
- *Cond-FiP (RFF):* We sample SCMs with non-linear functional relationships for training.

Note that in the main results (Tabel 2, Table 3) we show the performances of Cond-FiP trained by sampling SCMs with both linear and non-linear functional relationships. Hence, this ablations helps us to understand whether the strategy of training on mixed functional relationships can bring more generalization to the amortization process, or if we should have trained decoders specialized for linear and non-linear functional relationships.

We present the results of our ablation study in Table 5 and Table 6, for the task of sample generation and interventional generation respectively. Our findings indicate that Cond-FiP decoder trained for both linear and non-linear functional relationships is able to specialize for both the scenarios. While Cond-FiP (LIN) is only able to perform well for linear benchmarks, and similarly Cond-FiP (RFF) can only achieve decent predictions for non-linear benchmarks, Cond-FiP is achieve the best performances on both the linear and non-linear benchmarks.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
`Cond-FiP`(LIN)	10	0.07 (0.02)	0.4 (0.06)	0.07 (0.01)	0.25 (0.06)
`Cond-FiP`(RFF)	10	0.1 (0.02)	0.15 (0.02)	0.08 (0.01)	0.09 (0.01)
`Cond-FiP`	10	0.06 (0.01)	0.14 (0.02)	0.05 (0.01)	0.08 (0.01)
`Cond-FiP`(LIN)	20	0.07 (0.01)	0.44 (0.07)	0.10 (0.01)	0.58 (0.02)
`Cond-FiP`(RFF)	20	0.11 (0.01)	0.26 (0.06)	0.14 (0.01)	0.31 (0.03)
`Cond-FiP`	20	0.05 (0.01)	0.24 (0.06)	0.07 (0.01)	0.3 (0.03)
`Cond-FiP`(LIN)	50	0.10 (0.01)	0.5 (0.07)	0.14 (0.02)	0.69 (0.04)
`Cond-FiP`(RFF)	50	0.15 (0.02)	0.27 (0.05)	0.19 (0.02)	0.5 (0.07)
`Cond-FiP`	50	0.08 (0.01)	0.25 (0.05)	0.07 (0.0)	0.48 (0.07)
`Cond-FiP`(LIN)	100	0.1 (0.01)	0.51 (0.07)	0.15 (0.02)	0.72 (0.04)
`Cond-FiP`(RFF)	100	0.16 (0.03)	0.29 (0.07)	0.27 (0.04)	0.59 (0.06)
`Cond-FiP`	100	0.07 (0.01)	0.29 (0.07)	0.09 (0.01)	0.57 (0.07)

Table 5: **Ablation for Sample Generation.** We compare Cond-FiP for the task of generating samples from input noise variables against two variants. One variant corresponds to a decoder trained on SCMs with only linear functional relationships, `Cond-FiP`(LIN). Similarly, we have another variant where the decoder was trained on SCMs with only rff functional relationships, `Cond-FiP`(RFF). Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
`Cond-FiP`(LIN)	10	0.09 (0.02)	0.40 (0.07)	0.06 (0.01)	0.22 (0.04)
`Cond-FiP`(RFF)	10	0.16 (0.05)	0.22 (0.03)	0.08 (0.01)	0.11 (0.01)
`Cond-FiP`	10	0.10 (0.03)	0.21 (0.03)	0.07 (0.01)	0.11 (0.01)
`Cond-FiP`(LIN)	20	0.10 (0.01)	0.45 (0.07)	0.16 (0.03)	0.57 (0.02)
`Cond-FiP`(RFF)	20	0.14 (0.02)	0.26 (0.05)	0.21 (0.03)	0.32 (0.02)
`Cond-FiP`	20	0.09 (0.01)	0.24 (0.05)	0.14 (0.03)	0.31 (0.03)
`Cond-FiP`(LIN)	50	0.14 (0.02)	0.49 (0.07)	0.14 (0.02)	0.68 (0.04)
`Cond-FiP`(RFF)	50	0.19 (0.03)	0.28 (0.05)	0.21 (0.03)	0.49 (0.06)
`Cond-FiP`	50	0.13 (0.02)	0.27 (0.04)	0.12 (0.02)	0.48 (0.07)
`Cond-FiP`(LIN)	100	0.12 (0.02)	0.52 (0.07)	0.18 (0.03)	0.71 (0.04)
`Cond-FiP`(RFF)	100	0.18 (0.03)	0.32 (0.07)	0.24 (0.04)	0.59 (0.07)
`Cond-FiP`	100	0.10 (0.01)	0.30 (0.06)	0.14 (0.02)	0.58 (0.07)

Table 6: **Ablation for Interventional Generation.** We compare Cond-FiP against two variants for the task of interventional data from input noise variables. One variant corresponds to a decoder trained on SCMs with only linear functional relationships, `Cond-FiP`(LIN). Similarly, we have another variant where the decoder was trained on SCMs with only rff functional relationships, `Cond-FiP`(RFF). Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario.

G Evaluating generalization of Cond-Fip to larger sample size

In the main tables (Table 1, Table 5, and Table 3), we evaluated Cond-FiP's generalization capabilities to larger graphs ($d = 50$, $d = 100$) than those used for training ($d = 20$). In this section, we carry a similar experiment where instead of increasing the total nodes in the graph, we test Cond-FiP on datasets with more samples $n_{\text{test}} = 1000$, while Cond-FiP was only trained for datasets with sample size $n_{\text{train}} = 400$.

The results for the experiments are presented in Table 7, Table 8, and Table 9 for the task of noise prediction, sample generation, and interventional generation respectively. Our findings indicate that Cond-FiP is still able to compete with other baseline in this regime. However, we observe that the performances of Cond-FiP did not improve by increasing the sample size compared to the results obtained for the 400 samples case, meaning that the performance of our models depends exclusively on the setting used at training time. We leave for future works the learning of a larger instance of Cond-FiP trained on larger sample size problems.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
`DoWhy`	10	0.02 (0.0)	0.10 (0.01)	0.21 (0.04)	0.23 (0.02)
`DECI`	10	0.05 (0.01)	0.12 (0.01)	0.21 (0.04)	0.27 (0.03)
`FiP`	10	0.03 (0.0)	0.06 (0.0)	0.21 (0.04)	0.23 (0.02)
`Cond-FiP`	10	0.05 (0.01)	0.11 (0.01)	0.21 (0.04)	0.25 (0.02)
`DoWhy`	20	0.02 (0.0)	0.11 (0.02)	0.16 (0.01)	0.3 (0.02)
`DECI`	20	0.04 (0.01)	0.11 (0.02)	0.16 (0.01)	0.29 (0.02)
`FiP`	20	0.03 (0.0)	0.08 (0.02)	0.16 (0.01)	0.26 (0.02)
`Cond-FiP`	20	0.06 (0.01)	0.09 (0.01)	0.18 (0.01)	0.26 (0.01)

Table 7: **Results for Noise Prediction with larger sample size ($n_{\text{test}} = 1000$).** We compare Cond-FiP against the baselines for the task of predicting noise variables from the input observations. Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
`DoWhy`	10	0.04 (0.0)	0.14 (0.02)	0.29 (0.04)	0.3 (0.03)
`DECI`	10	0.07 (0.01)	0.17 (0.02)	0.29 (0.04)	0.33 (0.04)
`FiP`	10	0.05 (0.0)	0.09 (0.01)	0.29 (0.04)	0.29 (0.03)
`Cond-FiP`	10	0.05 (0.01)	0.14 (0.02)	0.29 (0.04)	0.29 (0.03)
`DoWhy`	20	0.04 (0.01)	0.21 (0.05)	0.28 (0.01)	0.55 (0.06)
`DECI`	20	0.07 (0.01)	0.21 (0.04)	0.29 (0.01)	0.59 (0.06)
`FiP`	20	0.05 (0.0)	0.17 (0.04)	0.28 (0.01)	0.53 (0.06)
`Cond-FiP`	20	0.05 (0.0)	0.24 (0.05)	0.28 (0.01)	0.53 (0.06)

Table 8: **Results for Sample Generation with larger sample size ($n_{\text{test}} = 1000$).** We compare Cond-FiP against the baselines for the task of generating samples from the input noise variables. Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario.

Method	Total Nodes	LIN **IN**	RFF **IN**	LIN **OUT**	RFF **OUT**
DoWhy	10	0.04 (0.01)	0.16 (0.03)	0.26 (0.03)	0.27 (0.03)
DECI	10	0.09 (0.01)	0.19 (0.02)	0.26 (0.03)	0.31 (0.04)
FiP	10	0.05 (0.01)	0.12 (0.02)	0.26 (0.03)	0.27 (0.03)
Cond-FiP	10	0.09 (0.02)	0.19 (0.03)	0.27 (0.03)	0.3 (0.03)
DoWhy	20	0.04 (0.0)	0.20 (0.04)	0.26 (0.01)	0.53 (0.06)
DECI	20	0.08 (0.01)	0.20 (0.03)	0.29 (0.02)	0.54 (0.05)
FiP	20	0.06 (0.01)	0.16 (0.04)	0.28 (0.02)	0.48 (0.06)
Cond-FiP	20	0.07 (0.01)	0.27 (0.05)	0.30 (0.02)	0.51 (0.06)

Table 9: **Results for Interventional Generation with larger sample size ($n_{\text{test}} = 1000$).** We compare Cond-FiP against the baselines for the task of generating interventional data from the input noise variables. Each cell reports the mean (standard error) RMSE over the multiple test datasets for each scenario.