Effective Bayesian Causal Inference via Structural Marginalisation and Autoregressive Orders

Christian Toth [1] **Christian Knoll** [2] **Franz Pernkopf** [1] **Robert Peharz** [1]

Abstract

Bayesian causal inference (BCI) naturally incorporates epistemic uncertainty about the true causal model into down-stream causal reasoning tasks by posterior averaging over causal models. However, this poses a tremendously hard computational problem due to the intractable number of causal structures to marginalise over. In this work, we decompose the structure learning problem into inferring (i) a causal order and (ii) a parent set for each variable given a causal order. By limiting the number of parents per variable, we can exactly marginalise over the parent sets in polynomial time, which leaves only the causal order to be marginalised. To this end, we propose a novel autoregressive model over causal orders (ARCO) learnable with gradient-based methods. Our method yields state-of-the-art in structure learning on simulated non-linear additive noise benchmarks with scale-free and Erdos-Renyi graph structures, and competitive results on real-world data. Moreover, we illustrate that our method accurately infers interventional distributions, which allows us to estimate posterior average causal effects and many other causal quantities of interest.

1. Introduction

Few topics in science and philosophy have been as controversial as the discussion about the nature of causality. Interestingly, the discussion becomes relatively benign, from a philosophical perspective, as soon as one agrees on a well-defined mathematical model of causality, such as a *structural causal model* (SCM) (Pearl, 2009). A fully specified SCM induces a wide range of joint probability distributions, organised on three levels denoted as *association*, *intervention* and *counterfactuals*, collectively forming the so-called *Pearl's causal hierarchy* (Bareinboim et al., 2022). Any causal question then elegantly reduces to standard probabilistic inference in some of these joint distributions. Hence, assuming that the data comes from *some* model within a considered class of SCMs, causal questions reduce, in principle, to *epistemic* questions, i.e., questions about what and how much is known about the model.

The view that causal questions reduce to epistemic questions about the true SCM naturally invites a *Bayesian* treatment, i.e., specifying a prior distribution over SCMs and inferring the posterior given collected data. Bayesian approaches to *causal discovery*, i.e., learning the causal structure (represented as *directed acyclic graph*, DAG), dates back to work as early as (Heckerman, 1995; Murphy, 2001; Tong & Koller, 2001). More intriguing, however, is to use a Bayesian approach over entire SCMs for the purpose of *Bayesian causal inference* (BCI), i.e., consistent reasoning about a *down-stream* causal query: usually, one is less interested in the SCM per se, but rather in some causal quantity, such as the existence of a *direct causal connection* (edge in the DAG), or the *average causal effect* one variable has on another. The goal of BCI is then to reason about such down-stream tasks by computing a Bayesian average over the whole posterior over SCMs.

Most existing BCI approaches are restricted to linear Gaussian models (Geiger & Heckerman, 1994; Viinikka et al., 2020; Pensar et al., 2020; Horii, 2021) or binary variables (Moffa et al., 2017; Kuipers et al., 2019), while a full Bayesian treatment of *non-linear* SCMs has been proposed only recently (Toth et al., 2022; Tigas et al., 2022). Both approaches leverage the DiBS method (Lorch et al., 2021), a gradient-based framework for Bayesian inference over DAGs, and infer causal mechanisms conditionally on the DAG. While Tigas et al. (2022) use Bayesian neural nets to represent the causal mechanisms and require stochastic variational inference over their parameters, Toth et al. (2022) leverage Gaussian Processes (GPs) for the mechanisms, which are marginalised in closed-form. The latter effectively represents a form of Rao-Blackwellisation, reducing

[1]Graz University of Technology, Austria [2]Levata GmbH, Graz, Austria. Correspondence to: Christian Toth <christian.toth@tugraz.at>, Robert Peharz <robert.peharz@tugraz.at>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria. Copyright 2024 by the author(s).

Code available at: https://github.com/chritoth/bci-arco-gp

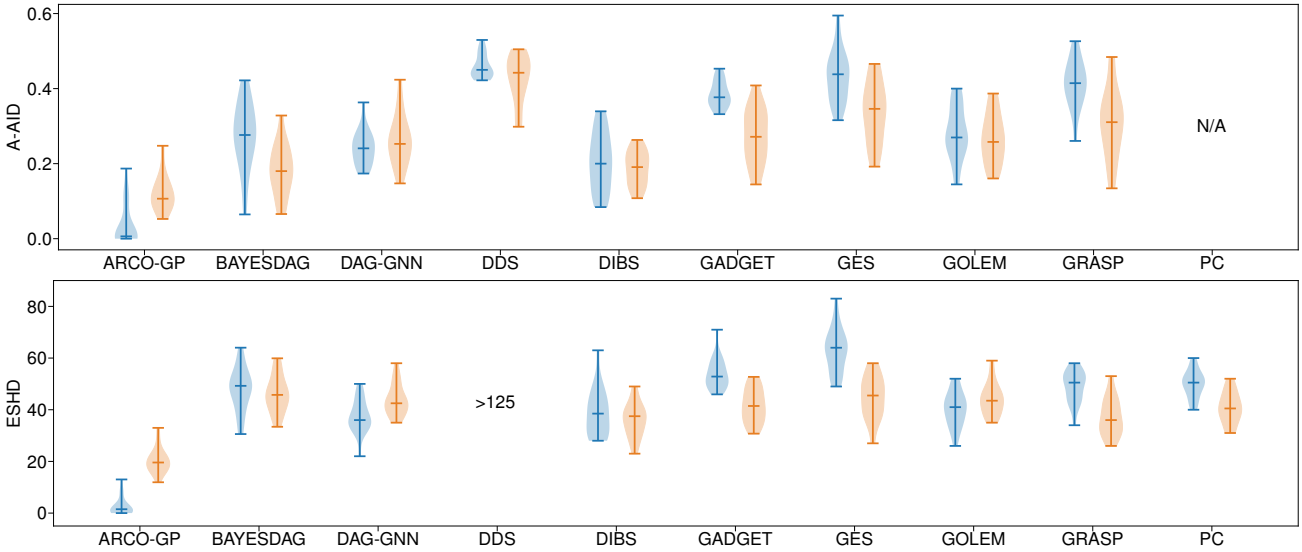

Figure 1. **Causal discovery on nonlinear additive noise models.** Structure learning results in terms of *expected Hamming distance* (ESHD) and *ancestor adjustment identification distance* (A-AID) on simulated non-linear models with scale-free (left, blue) and Erdös-Renyi (right, orange) graphs, each with 20 nodes and 200 data samples. Whiskers indicate maximum, minimum and median values across 20 simulated ground truth instances. For both metrics lower is better. Range for ESHD is set for better readability, omitting the result for DDS (> 125).

inference to the problem of marginalising over DAGs.[1]

In this paper, we enhance non-linear BCI and decompose structure inference to (i) inferring a causal order over the endogenous variables and (ii) inferring parent sets for each variable conditional on the order.[2] We perform the latter by restricting the number of parents per variable to a fixed maximal number K and exactly marginalising over all possible parent sets in polynomial ($\mathcal{O}(d^K)$) time, where d is the number of endogenous variables. This technique is well-established in Markov chain Monte Carlo (MCMC) techniques for Bayesian structure learning (Koller & Friedman, 2003; Koivisto & Sood, 2004; Viinikka et al., 2020), but has—to the best of our knowledge—not been exploited in gradient-based structure learning so far. This structural marginalisation, in combination with the analytic treatment of the causal mechanisms via GPs (Toth et al., 2022), reduces the inference problem to the marginalisation w.r.t. causal orders. To this end, we propose a novel neural generative model over topological orders which can be understood as an *A*uto-*R*egressive distribution estimator (Larochelle & Murray, 2011), constrained to *C*ausal *O*rders, hence we dub this model *ARCO*.

[1]The term Rao-Blackwellisation is used in a generalised sense here, denoting a generally observed variance reduction in Monte Carlo estimates when "nuisance variables" are integrated out (Robert & Roberts, 2021).

[2]In this paper, we use the term *causal order* for the topological order of a causal DAG. We use this term to emphasise that this order per se might be an object of interest to the practitioner.

Figure 1 show results for simulated scale-free and Erdös-Renyi graphs, with non-linear causal mechanisms and additive Gaussian noise, demonstrating that ARCO-GP (meaning BCI via ARCO and GPs) is highly effective for causal discovery. In particular, for scale-free graphs, where the assumption of a restricted number of parents matches our modelling assumption, ARCO-GP clearly outperforms a wide range of competitor methods. Moreover, even when the assumptions are violated (in our case for Erdos-Renyi graphs), we demonstrate that ARCO-GP still outperforms the baselines, albeit with a smaller margin. Moreover, in Figure 2 we demonstrate ARCO-GP's efficacy in down-stream BCI, by sampling from various interventional distributions. Specifically, by sampling from the whole Bayesian model (including causal structure and mechanisms) we effectively perform posterior marginalisation over SCMs. Detailed results, including further evaluation metrics and results on a real-world dataset (Sachs et al., 2005), are presented in Section Section 5 and Appendix C.

In summary, our main contributions are:

- We propose an effective BCI framework which *exactly* marginalises both causal mechanisms and parent sets from the posterior over SCMs, reducing the hard BCI problem to inferring the causal (topological) order.
- We propose ARCO, a neural autoregressive model over causal orders and devise a gradient-based learning scheme for it. While here ARCO is used mainly for causal reasoning, it might well find applications in

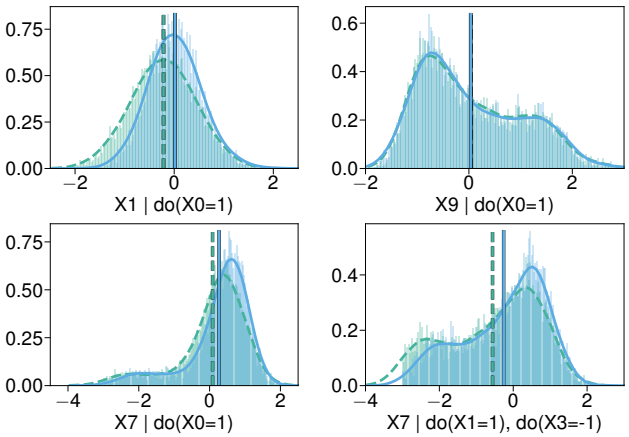

Figure 2. **Posterior interventional distributions.** Several interventional distributions as inferred by ARCO-GP (blue, solid) and the corresponding ground truth (green dashed). Specifically, we sampled full SCMs (orders, parent sets, mechanisms, exogeneous variables) and performed the indicated intervention to produce a sample from the corresponding distribution, which effectively marginalises over the posterior over SCMs. Vertical lines indicate the estimated means (averages). The underlying SCM is a simulated non-linear additive noise model with graph structure taken as the consensus protein interaction graph from Sachs et al. (2005) (see Figure 4). See Appendix C for details.

many other domains requiring generative models over total orders (e.g., rankings).

- We demonstrate in experiments that ARCO-GP sets state-of-the-art against a wide range of baselines when our model assumption are exactly matched. When the model assumptions are violated, we still outperform or are competitive with all baselines.

2. Background

Structural Causal Models. An SCM $\mathcal{M}$ over observed endogenous variables $\mathbf{X} = \{X_1, \ldots, X_d\}$ and unobserved exogenous variables $\mathbf{U} = \{U_1, \ldots, U_d\}$ consists of structural equations, or mechanisms,

$$X_i := f_i(\mathbf{Pa}_i, U_i), \qquad \text{for} \quad i \in \{1, \ldots, d\}, \quad (1)$$

which assign the value of each X_i as a deterministic function f_i of its direct causes, or causal parents, $\mathbf{Pa}_i \subseteq \boldsymbol{X} \setminus \{X_i\}$ and an exogenous variable U_i, and a joint distribution $p(\mathbf{U})$ over the exogenous variables. In this paper we assume that the exogenous variables are independent Gaussian and enter in additive fashion, i.e. $f_i(\mathbf{Pa}_i, U_i) = f_i(\mathbf{Pa}_i) + U_i$, thus implying causal sufficiency. Associated with each SCM is a directed graph G induced by the set of parent sets $\mathbf{Pa} = \{\mathbf{Pa}_i\}_{i=1}^d$ with vertices $\boldsymbol{X}$ and edges $X_j \to X_i$ if and only if $X_j \in \mathbf{Pa}_i$. Any SCM with an *acyclic* directed graph (DAG) then induces a unique observational distribution $p(\boldsymbol{X} \mid \mathcal{M})$ over the endogenous variables $\boldsymbol{X}$, which is obtained as the pushforward measure of $p(\boldsymbol{U})$ through the causal mechanisms in Equation (1).

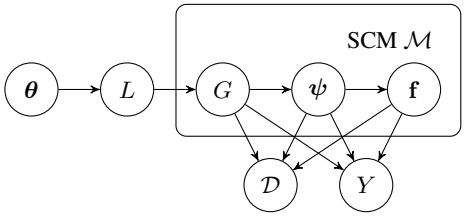

Figure 3. **Bayesian network view of our generative model.** We characterise a Structural Causal Model (SCM) by a causal graph G, causal mechanisms $\mathbf{f}$ and parameters $\boldsymbol{\psi}$ of a joint distribution over mechanisms and exogenous variables $p(\mathbf{f}, \mathbf{U} \mid \boldsymbol{\psi})$. We model the mechanisms $\mathbf{f}$ using Gaussian Processes (GPs) and $\mathbf{U}$ as additive Gaussian noise, implying that $\boldsymbol{\psi}$ is a set of GP hyperparameters. The SCM gives rise to the data-generating likelihood $p(\mathcal{D} \mid \mathbf{f}, \boldsymbol{\psi}, G)$ and determines the (distribution over the) causal query Y. To sample a directed acyclic causal graph G, we first sample a causal order L from a neural autoregressive distribution over causal orders (ARCO) $p(L \mid \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$. Given a causal order, we can then sample or marginalise exactly over causal graphs G by limiting the maximum cardinality of parent sets.

A (hard) intervention $do(\mathbf{W} = \mathbf{w})$ on a set of endogenous variables $\mathbf{W} \subset \mathbf{X}$ replaces the targeted mechanisms with constants $\mathbf{w}$, resulting in a modified SCM. The entailed modified causal graph lacks the incoming edges into any intervention target. The pushforward through the modified SCM yields an interventional distribution $p(\mathbf{X} \mid do(\mathbf{W} = \mathbf{w}), \mathcal{M})$.

Causal Orders. A permutation $L = \langle L_1, \ldots, L_d \rangle$ of the endogenous variables $\mathbf{X} = \bigcup_{i=1}^d \{L_i\}$, where $L_i \neq L_j$ for all $i \neq j$, entails a strict total order $L_1 \prec L_2 \prec \cdots \prec L_d$ among the variables. Henceforth, we refer to such a permutation L as a *causal order*. A causal order L constrains the possible causal interactions between the variables, i.e., X_i can be a (direct) cause of X_j if, and only if, $X_i \prec X_j$ in L. We define $L_{<k} = \langle L_1, \ldots, L_{k-1} \rangle$ to be the first $k-1$ elements in L. Finally, let $\lambda^L \colon \mathbf{X} \mapsto \{1, \ldots, d\}$ be the bijective mapping between $\boldsymbol{X}$ and indices in L, i.e., $\lambda^L(X_i) = k \iff L_k = X_i$. We then denote by $Q^L \in \{0,1\}^{d \times d}$ the *permutation matrix* representing L, where $Q^L_{ij} = 1$ iff $\lambda^L(X_i) = j$.

3. Bayesian Causal Inference via Structural Marginalisation

When performing causal reasoning in the BCI framework, we define the causal quantity of our interest as Y, called the *causal query*, which is a function of the SCM $\mathcal{M}$ (see (Toth et al., 2022)). This causal query could be, for example,

an endogenous variable under some intervention, (features of) the true causal graph, or even the entire SCM. Since we are following a Bayesian approach, $\mathcal{M}$ is a random variable equipped with a prior distribution $p(\mathcal{M})$, and hence also Y is a random variable.

Given a set of (observational) data $\mathcal{D} = \{\mathbf{x}_n \overset{\text{i.i.d}}{\sim} p(\mathbf{X} \mid \mathcal{M}^*)\}_{n=1}^N$ collected from the true underlying SCM $\mathcal{M}^*$, we are interested in inferring the posterior $p(Y \mid \mathcal{D})$ of the causal query, reflecting our uncertainty about $\mathcal{M}^*$. Since the causal query Y is determined by the SCM, the query posterior is given as

$$p(Y \mid \mathcal{D}) = \mathbb{E}_{\mathcal{M} \mid \mathcal{D}}[p(Y \mid \mathcal{M})], \tag{2}$$

marginalising over posterior SCMs. Our inference procedure, described in Algorithm 1 and the generative model over SCMs shown in Figure 3, divides into a parameter learning and an inference phase. In the learning phase (Algorithm 1, lines 1-5), we infer posterior parameters $p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D})$. In the inference phase (Algorithm 1, lines 6-11), we use samples drawn from the learned generative model to approximate the query posterior. We first provide an overview of the individual phases starting with the inference phase, as it motivates our learning objective. We elaborate on the more technical details in Sections 3.1 to 3.3.

Inference Phase. Assuming we have already learned a set of posterior parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$ approximating $p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D})$ (see the learning phase), we want to employ our model to approximate the query posterior in Equation (2). A key observation similar in spirit to (Lorch et al., 2021; Toth et al., 2022) is that the expectation w.r.t. SCMs can be written as the following importance weighted expectation (for a derivation see Appendix A.1)

$$\begin{aligned} p(Y \mid \mathcal{D}) &= \mathbb{E}_{\mathcal{M} \mid \mathcal{D}}\left[p(Y \mid \mathcal{M})\right] \\ &= \mathbb{E}_{\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{L \mid \boldsymbol{\theta}}\left[w^L \cdot \mathbb{E}_{G \mid L, \boldsymbol{\psi}, \mathcal{D}}\left[\mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi}, \mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right]\right]\right] \end{aligned} \tag{3}$$

with importance weights

$$w^L := \frac{\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]}{\mathbb{E}_{L' \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G' \mid L'}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G') \cdot p(\boldsymbol{\psi} \mid G')\right]\right]}, \tag{4}$$

On a high level, the query posterior $p(Y \mid \mathcal{D})$ is estimated in Eq. (3) by (i) sampling several candidate SCMs from the learned generative model in a nested manner given $\boldsymbol{\theta}, \boldsymbol{\psi}$—first sampling order L, then a graph G (parent sets) conditional on L, and mechanisms $\mathbf{f}$ conditional on G, (ii) sample queries given the candidate SCM from $p(Y \mid \mathcal{M})$, and (iii) weighting the sampled queries with their corresponding importance weight w^L.

Learning Phase. The formulation in Equation (3) transforms the inference problem over posterior SCMs in Equation (2) into the problem of inferring posterior parameters $p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid D)$ of our proposed generative model in Figure 3 (cf. (Lorch et al., 2021; Toth et al., 2022)). By ensuring that our generative distributions $p(L \mid \boldsymbol{\theta})$ and $p(\mathbf{f}, \mathbf{U} \mid \boldsymbol{\psi})$ are sufficiently expressive, it suffices to infer a maximum a posteriori (MAP) estimate of the posterior parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$ via gradient-based optimisation of (see Appendix A.2 for a derivation)

$$\begin{aligned} \nabla \log p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D}) &= \nabla \log p(\boldsymbol{\theta}) \\ &\quad + \nabla \log \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]\right]. \end{aligned} \tag{5}$$

A major issue for the training of such a model is the quality of the estimated gradients in the face of the high-dimensional problem space and the coupling of the parameters $\boldsymbol{\theta}, \boldsymbol{\psi}$. However, simultaneously updating all model parameters in a single gradient step is prone to yield very noisy gradients:

1. The gradient w.r.t. $\boldsymbol{\theta}$ (causal order proposal) depends on $\boldsymbol{\psi}$ (mechanism and noise parameters) through the quality of the estimated importance weights w^L in Equation (6). A bad estimate of $\boldsymbol{\psi}$ will result in poor estimates of the importance weights and consequently the gradient w.r.t. $\boldsymbol{\theta}$.

2. The gradient w.r.t. $\boldsymbol{\psi}$ likewise depends on $\boldsymbol{\theta}$ via the quality of the sampled orders and their induced parent sets. In general, the more often a parent set occurs in the sampled orders relative to other parent sets, the stronger its gradient w.r.t. $\boldsymbol{\psi}$.[3]

To mitigate these issues, we propose a nested optimisation procedure as laid out in Algorithm 1 (lines 1-5). In an outer loop, we sample causal orders from the learned proposal, estimate the gradient w.r.t. $\boldsymbol{\theta}$ in Equation (5) and update the parameters $\boldsymbol{\theta}$ of the proposal distribution $p(L \mid \boldsymbol{\theta})$ (see Section 3.1). To estimate this gradient, we need to compute the importance weights (Equation (4)) for the sampled orders, which however depend on the GP hyper-parameters $\boldsymbol{\psi}$ via the marginal likelihood $p(\mathcal{D} \mid \boldsymbol{\psi}, G)$. Therefore, *before* computing the importance weights, we optimise the GP hyper-parameters (see Section 3.3) for each yet unseen parent set compatible with some sampled order. This ensures truthful estimates of the importance weights for the sampled orders and, consequently, provides a good gradient estimate.

3.1. Causal Order Inference

While Equations (3) and (5) elegantly reduce BCI to learning a model over causal orders and GP hyper-parameters, computing the expectation w.r.t. causal orders in Equation (3)

[3] Arguably, this may be especially problematic when sharing parameters between mechanisms with different parent sets, as is the case, e.g., when modeling the mechanisms with a single, masked neural network.

Algorithm 1: BCI with ARCO-GP

Input: (Observational) data $\mathcal{D}$.
Output: Posterior parameters $\boldsymbol{\theta}$, $\boldsymbol{\psi}$. Estimated (posterior over the) causal query Y and importance weights $\mathbf{w}^L$.

/* Learning Phase */
1 **repeat**
2 sample causal orders $\mathbf{L} \leftarrow \{L_m \sim p(L \mid \boldsymbol{\theta})\}$ ▷Section 3.1
3 $\mathbf{w}^L, \boldsymbol{\psi} \leftarrow$ ComputeIW($\mathbf{L}$, $\mathcal{D}$) ▷Equation (4)
4 estimate the gradients for $\boldsymbol{\theta}$ and perform a gradient step ▷Equation (6)
5 **until** stopping criterion fulfilled
/* Inference Phase */
6 sample causal orders $\mathbf{L} \leftarrow \{L_m \sim p(L \mid \boldsymbol{\theta})\}$ ▷Section 3.1
7 $\mathbf{w}^L, \boldsymbol{\psi} \leftarrow$ ComputeIW($\mathbf{L}$, $\mathcal{D}$) ▷Equation (4)
8 *[optional]* sample graphs $\mathbf{G} \leftarrow \{G_k \sim p(G \mid L, \boldsymbol{\psi}, \mathcal{D})\}$ ▷Section 3.2
9 *[optional]* sample mechanisms $\mathbf{f} \leftarrow \{\mathbf{f}_j \sim p(\mathbf{f} \mid \boldsymbol{\psi}, \mathcal{D})\}$ ▷Section 3.3
10 sample candidate queries $\mathbf{Y} \leftarrow \{Y_i \sim p(Y \mid \mathbf{f}, \boldsymbol{\psi}, G)\}$ ▷Section 3.3
11 **return** $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, $\mathbf{Y}$, $\mathbf{w}^L$
/* Subroutine: compute importance weights and update mechanisms. */
12 **Subroutine** ComputeIW($\mathbf{L}$, $\mathcal{D}$)
13 **foreach** causal order $L_m \in \mathbf{L}$ **do**
14 **foreach** parent set $\mathbf{pa}_i$ compatible with L_m **do**
15 Learn (or retrieve) the corresponding GP hyper-parameters $\boldsymbol{\psi}_i$ ▷Equation (11)
16 **end**
17 Compute the importance weight w^{L_m}
18 **end**
19 **return** $\mathbf{w}^L$, $\boldsymbol{\psi}$

poses a hard combinatorial problem, as there are $d!$ possible causal orders over d variables. The involved distribution over causal orders $p(L \mid \boldsymbol{\theta})$ appearing in the BCI estimator Equation (3) can be interpreted as proposal distribution, with importance weights as defined in Equation (4), where the optimal proposal is the true posterior over causal orders. Hence, we require an expressive representation over causal orders which can account for *multi-modal distributions* over orders. In particular, consider the example of a Markov equivalence class (MEC) including a chain graph. Since the chain is contained in the MEC, also the reverse chain graph must be in the MEC, and thus, the proposal over causal orders must be able to represent both orders with equal probability.

A simple parameterisation of orders as proposed in (Charpentier et al., 2022) is not able to represent the true posterior over causal orders in this case. Specifically, they sample orders using the Gumbel Top-k trick (Kool et al., 2019) by perturbing d logits (corresponding to the d variables) with Gumbel noise and sorting these perturbed logits, yielding an order over variables. In essence, learning such a model boils down to ordering (and spreading) the d logits on the real line. Now, to sample a chain graph and a reverse chain graph, some variable is the first element in the causal order in one case and must thus have the highest (perturbed) logit, and it is the last element in the causal order in the other case where it must have the lowest (perturbed) logit, which is contradictory. In practice, we observe that when trying to learn a multi-modal distribution over causal orders with this model, the logits cluster together, resulting approximately in a uniform distribution over causal orders.

Here, we propose an expressive, *A*uto-*R*egressive distribution (Larochelle & Murray, 2011) $p(L \mid \boldsymbol{\theta}) = p(L_1 \mid \boldsymbol{\theta}) \cdot \prod_{k=2}^{d} p(L_k \mid L_{<k}, \boldsymbol{\theta})$ over *C*ausal *O*rders (ARCO) that is amenable to gradient-based optimisation and can represent multi-modal distributions over orders, avoiding the shortcoming described above.

Sampling Causal Orders. The ordering of variables naturally implies a sequential sampling procedure as listed in Algorithm 2. In each step of the sampling procedure, we sample the next variable in the order from a categorical distribution $p(L_k \mid L_{<k}, \boldsymbol{\theta})$ over the set of yet unassigned variables, conditional on all preceding variables in the order (Algorithm 2, line 6). To account for the dependence on the preceding order $L_{<k}$, we compute the logits of the categorical distribution using a differentiable function $g_{\boldsymbol{\theta}} : \mathbb{R}^{d \times d} \mapsto \mathbb{R}^d$ (Algorithm 2, line 4) and re-normalise them to exclude the elements in $L_{<k}$ (Algorithm 2, line 5). We implement $g_{\boldsymbol{\theta}}$ as feed-forward neural network, taking as input a suitable encoding of the so-far sampled order $L_{<k}$. To this end, we encode $L_{<k}$ using its induced permutation matrix $Q^{L_{<k}}$ (see Section 2) and mask the rows corresponding to elements $L_{>=k}$ with zeros.

Training ARCO. Training ARCO amounts to learning the parameters $\boldsymbol{\theta}$ of the neural network $g_{\boldsymbol{\theta}}$ by performing gradient ascent on Equation (5). To estimate the gradient in Equation (5) w.r.t. $\boldsymbol{\theta}$ we use the score-function estimator, yielding (see Appendix A.2)

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \mathbb{E}_{L \mid \boldsymbol{\theta}} \left[w^L \cdot \nabla_{\boldsymbol{\theta}} \log p(L \mid \boldsymbol{\theta}) \right]. \tag{6}$$

Algorithm 2: Sample Causal Order with ARCO

Input: Logit function $g_{\boldsymbol{\theta}} : \mathbb{R}^{d \times d} \mapsto \mathbb{R}^d$
Output: Causal order L sampled from $p(L \mid \boldsymbol{\theta})$

$\mathbf{R} \leftarrow \mathbf{X}$ ▷set of unassigned elements
$L \leftarrow \varnothing$ ▷causal order
for $k = 1 \dots d$ **do**
  $\boldsymbol{\phi} \leftarrow g_{\boldsymbol{\theta}}(Q^{L_{<k}})$ ▷compute logits
  $\phi_i \leftarrow \begin{cases} \phi_i - \log \sum_{j \mid X_j \in \mathbf{R}} \exp \phi_j & \text{if} \quad X_i \in \mathbf{R} \\ -\infty & \text{otherwise} \end{cases}$ ▷normalise logits
  $l \sim \text{CATEGORICAL}(\boldsymbol{\phi})$ ▷sample next element
  $L \leftarrow L \cup \langle X_l \rangle$ ▷update causal order
  $\mathbf{R} \leftarrow \mathbf{R} \setminus \{X_l\}$
end

with w^L as defined in Equation (4) (see Appendix A.1 for the derivation). To evaluate $\log p(L \mid \boldsymbol{\theta})$ for a given causal order L, we simply need to sum the log-probabilities of the categorical distributions $\log p(L_k \mid L_{<k}, \boldsymbol{\theta})$ for the respective elements L_k. The necessary log-probabilities (i.e., the normalised logits) are computed as described in Algorithm 2. Note that, although we need to compute the logits sequentially in the sampling procedure, we can compute them in parallel during evaluation.

3.2. Marginalising over Causal Graphs

The marginalisation w.r.t. causal graphs G in Equations (3) to (5) is in general intractable, as the number of DAGs consistent with any given (causal) order is $2^{\frac{d \cdot (d-1)}{2}}$. Although this is significantly smaller than the total number of DAGs with d nodes (which grows super-exponentially in d, see e.g. (OEIS Foundation Inc., 2024)), an exhaustive enumeration is still infeasible.

In this work, we tackle this problem by restricting the number of parents per variable. By restricting the maximum size of any admissible parent set to some integer K, the number of distinct parent sets consistent with any causal order L is in $\mathcal{O}(d^K)$. Although the exhaustive enumeration of all DAGs with restricted parent set size is still infeasible, it turns out that, given a causal order, the expectation w.r.t. graphs can be tractably computed under certain assumptions: by choosing a prior over graphs $p(G \mid L) = \prod_i p(\mathbf{Pa}_i \mid L)$ that factorises over parent sets, we can compute quantities $h(G)$ that decompose over the parent sets by the following two propositions.[4]

[4] In practice, we assume a uniform prior over parent sets consistent with a given causal order.

Proposition 3.1. *Let $h(G) = \prod_i h_i(\boldsymbol{pa}_i^G)$ and $w(G) = \prod_i w(\boldsymbol{pa}_i^G)$ be factorising over the parent sets, then we can compute*

$$\mathbb{E}_{G \mid L}\left[w(G)h(G)\right] = \prod_{i=1}^{d} \sum_{\boldsymbol{pa}_i} p(\boldsymbol{pa}_i \mid L) w_i(\boldsymbol{pa}_i) h_i(\boldsymbol{pa}_i). \tag{7}$$

Proposition 3.2. *Let $h(G) = \sum_i h_i(\boldsymbol{pa}_i^G)$ be summing and $w(G) = \prod_i w(\boldsymbol{pa}_i^G)$ be factorising over the parent sets, then we can compute*

$$\mathbb{E}_{G \mid L}\left[w(G)h(G)\right] = \sum_{i=1}^{d} \left(\prod_{j \neq i} \alpha_j(L) \right) \cdot \sum_{\boldsymbol{pa}_i} p(\boldsymbol{pa}_i \mid L) w_i(\boldsymbol{pa}_i) h_i(\boldsymbol{pa}_i), \tag{8}$$

where

$$\alpha_j(L) = \sum_{\boldsymbol{pa}_i} p(\boldsymbol{pa}_i \mid L) w_i(\boldsymbol{pa}_i).$$

These propositions generalise the results presented by Koller & Friedman (2003); Koivisto & Sood (2004) on how to compute the posterior probabilities of edges or parent sets, which is a special case of Proposition 3.1, to our setting of Bayesian causal inference (proofs in Appendix A.3).

Employing Proposition 3.1, we can compute $\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]$, and consequently

$$p(G \mid L, \boldsymbol{\psi}, \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G) \cdot p(G \mid L)}{\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]}, \tag{9}$$

in closed-form by letting $h(G) = p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)$ and $w(G) = 1$. This allows us to tractably compute the respective expectations w.r.t. causal graphs in Equations (3) to (5). Furthermore, for any causal query that does not decompose over parent sets, we can compute a Monte-Carlo estimate of the query by sampling DAGs from the true posterior $p(G \mid L, \boldsymbol{\psi}, \mathcal{D})$ in Equation (3), as Equation (9) factorised over parent sets.

3.3. Mechanism Inference and Marginalisation

To compute the importance weights in Equation (4) we need to compute the marginal log-likelihood $p(\mathcal{D} \mid \boldsymbol{\psi}, G)$ which is intractable for general models. As we focus on non-linear additive noise models in this work, we follow (von Kügelgen et al., 2019; Toth et al., 2022) and model each mechanism via a distinct GPs, assuming homoscedastic Gaussian noise and causal sufficiency. Under these assumptions, we can compute the marginal likelihood $p(\mathcal{D}_i \mid \boldsymbol{\psi}_i, G) = \mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi}_i}\left[p(\mathcal{D}_i \mid \mathbf{f}_i, \boldsymbol{\psi}_i, \mathbf{pa}_i)\right]$ in closed form. The same holds true for the predictive posterior

$p(\mathcal{D}_i^{\text{test}} \mid \boldsymbol{\psi}_i, G, \mathcal{D})$ possibly needed in Equation (3). We factorise the GP hyper-parameter prior[5] and the likelihood over parent sets for each node i, i.e.,

$$p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G) = \prod_i p(\mathcal{D}_i \mid \boldsymbol{\psi}_i, G) \cdot p(\boldsymbol{\psi}_i \mid \mathbf{Pa}_i^G), \tag{10}$$

allowing for efficient computation by caching intermediate results. For the individual GP we infer a MAP-Type II estimate of its hyper-parameters by performing gradient-ascent on[6]

$$\nabla_{\boldsymbol{\psi}_i} \log p(\boldsymbol{\psi}_i \mid \mathcal{D}, G) = \nabla_{\boldsymbol{\psi}_i} \log p(\mathcal{D}_i \mid \boldsymbol{\psi}_i, G) + \nabla_{\boldsymbol{\psi}_i} \log p(\boldsymbol{\psi}_i \mid G). \tag{11}$$

For root nodes, i.e., nodes without parents, we place a conjugate normal-inverse-gamma prior on the mean and variance of that node, which also allows for closed-form inference.

4. Related Work

BCI in the simpler form of Bayesian structure inference dates back to (Heckerman, 1995; Heckerman et al., 1997; Madigan et al., 1995), often using MCMC techniques, where many works utilise (causal) orders, e.g., (Koller & Friedman, 2003; Koivisto & Sood, 2004; Teyssier & Koller, 2012; Ellis & Wong, 2008; Niinimäki et al., 2016; Kuipers & Moffa, 2017; Viinikka et al., 2020), allowing a better exploration of the posterior space. Moreover, restricting the maximum number of parents per node allows an exhaustive enumeration of all parent sets in polynomial time for a given order. However, MCMC inference comes with its own set of challenges, and none of these works implement non-linear mechanism models. In contrast, our work focuses on non-linear additive noise models and utilises gradient-based learning of a generative model over causal orders. Besides sampling based inference, orders can also facilitate exact optimization schemes, e.g. (Cussens, 2010; De Campos & Ji, 2011; Peharz & Pernkopf, 2012).

After the advent of gradient-based (Bayesian) DAG structure learning methods (Zheng et al., 2018; Yu et al., 2019; Brouillard et al., 2020; Lachapelle et al., 2020; Lorch et al., 2021), inference via orders recently gained interest in the gradient-based causal structure learning community as a vehicle to sample DAGs without the need of utilising soft acyclicity constraints during optimisation (Cundy et al., 2021; Charpentier et al., 2022; Annadani et al., 2023). In contrast to these works, we stress the importance of an expressive gradient-based model for causal orders and we utilise causal orders to marginalise our Bayesian causal inference procedure by exhaustive parent set enumeration.

Finally, most existing BCI approaches are restricted to linear Gaussian models (Geiger & Heckerman, 1994; Viinikka et al., 2020; Pensar et al., 2020; Horii, 2021) or binary variables (Moffa et al., 2017; Kuipers et al., 2019). Only recently, works on a Bayesian treatment of entire *non-linear* SCMs (i.e., including mechanisms and exogenous noise) have been proposed by Toth et al. (2022); Tigas et al. (2022); Giudice et al. (2023; 2024). Both Toth et al. (2022) and Tigas et al. (2022) operate in an active learning scenario using DIBS (Lorch et al., 2021) for inferring posterior causal graphs, whereas Toth et al. (2022) focus on BCI while using GPs for mechanism inference. Similarly, Giudice et al. (2023) use GPs for the Bayesian inference of SCMs in combiniation with MCMC inference for causal graphs. In a concurrent preprint, Giudice et al. (2024) follow (Toth et al., 2022) in using GPs for BCI but use MCMC for sampling posterior causal graphs. None of these approaches features gradient-based inference of causal orders, nor utilises causal orders for structure marginalisation.

5. Experiments

Task I: Causal Structure Learning. For our results in Figure 1, we sample a fixed set of 200 training samples from the observational distribution of non-linear additive noise ground truth SCMs with Erdös-Rényi (ER) (Erdös & Rényi, 1959) and scale-free (SF) (Barabási & Albert, 1999) graph structures. The small sample size emulates the setting of significant uncertainty relevant to the Bayesian inference scenario. The reported *expected structural Hamming distance* (ESHD) is a standard metric used structure learning metric. Additionally, to asses the inferred structures in terms of causal implications, we report the *ancestor adjustment identification distance* (A-AID) recently proposed by Henckel et al. (2024) (see Appendix D for details).

We compare our inference model (ARCO-GP) with maximum parent set size $K = 2$ to a diverse set of nine different structure learning methods: BAYESDAG ((Annadani et al., 2023)), DAG-GNN ((Yu et al., 2019)), DDS ((Charpentier et al., 2022)), DIBS-GP ((Toth et al., 2022; Lorch et al., 2021)), GADGET ((Viinikka et al., 2020)), GES ((Chickering, 2003)), GOLEM ((Ng et al., 2020)), GRASP ((Lam et al., 2022)) and PC ((Spirtes et al., 2000)), that we briefly describe in Appendix D.

Summarising Figure 1, when our assumption about the maximum parent set size is reflected by the ground truth models (as holds in our simulated SF graphs), ARCO-GP clearly outperforms the baselines in terms of ESHD and A-AID.

[5]The parameter set $\boldsymbol{\psi}_i$ contains the GP's kernel parameters and the noise variance. Note that, other than the structural dependence on the parent sets, the prior parameters $\boldsymbol{\psi}$ are independent of $\boldsymbol{\theta}$.

[6]Importantly, in Appendix A.2 we show that this not an ad-hoc choice, but a consequence of optimising Equation (5).

Moreover, even when the ground truth models exhibit larger parent sets and our restriction is violated (as is the case in our simulated ER graphs), ARCO-GP still outperforms the baselines, albeit by a smaller margin. Additional experiments with varying number of variables $d \in \{11, 20, 50\}$, as well as experiments on real-world data from Sachs et al. (2005) and ablations are provided in Appendix D. For these experiments we report an extended set of eight additional metrics.

Task II: Inferring Interventional Distributions and Average Causal Effects. We illustrate ARCO-GP's causal reasoning capabilities by visualising a selected set of estimated posterior interventional distributions $p(X_i \mid do(\mathbf{W} = \mathbf{w}), \mathcal{D})$ in Figure 2. In order to have access to the corresponding ground truth distributions, we simulate a non-linear additive noise model on the consensus protein interaction graph reported by Sachs et al. (2005) (see Figure 4). We generate samples from the interventional distributions with the procedure laid out in Algorithm 1 and Section 3 and smoothing the empirical distribution with a kernel density estimate (see Appendix B for details). As can be seen, the posterior distribution inferred by our method closely resemble the true ones. Importantly, the multi-modality of the (inferred) distributions illustrate the benefits of a Bayesian approach to causal inference, as we can represent uncertainty via full distributions instead of single point estimates. In Appendix D we further compare our ability to infer pairwise average causal effects to two baselines, namely DIBS-GP (Toth et al., 2022; Lorch et al., 2021) and GADET+BEEPS (Viinikka et al., 2020).

6. Discussion

We demonstrated that our proposed ARCO-GP method for BCI, leveraging structural marginalisation, yields superior structure learning performance on non-linear additive noise models against a set of nine state-of-the-art baseline methods. Moreover, we illustrate the capability of ARCO-GP to accurately infer posterior interventional distributions and average causal effects. The capabilities our method relies upon the following assumptions and limitations.

Assumptions. Our assumptions on the data generating process include causal sufficiency and additive Gaussian noise, as well as a limited maximum parent set size. Presumably, for very dense graphs the performance of ARCO-GP will deteriorate in comparison to other methods. While our framework and implementation can handle interventional data, we do not evaluate this scenario experimentally because not all baselines support the use of interventional data, and the observational case is the more difficult problem from the perspective of model identifiability.

Scalability and Computation. The main driver of complexity is the exact inference using GPs, which grows with N^3 in the number of available data points. Although we used only CPUs for running our experiments, scaleable GPU inference techniques for GPs were proposed, e.g., by Gardner et al. (2018); Pleiss et al. (2018). Additionally, the training of the GPs could be straightforwardly parallelised. Conceptually, our framework is flexible and modular, allowing to use alternative mechanism models like normalising flows as in (Brouillard et al., 2020; Pawlowski et al., 2020). A second driver of computational complexity is the exhaustive enumeration of parent sets, which may be prohibitive on larger problem instances and bigger parent sets. Note, however, that the individual parent set contributions necessary to compute the importance weights in Equation (4) could be pre-computed in parallel.

Acknowledgements

The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged. The computational results presented have in part been achieved using the Vienna Scientific Cluster (VSC).

References

Annadani, Y., Pawlowski, N., Jennings, J., Bauer, S., Zhang, C., and Gong, W. BayesDAG: Gradient-Based Posterior Inference for Causal Discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C. K., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., and Chintala, S. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, New York, NY, USA, 2024. ACM. ISBN 9798400703850. doi: 10.1145/3620665.3640366.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science*, 286, 1999.

Bareinboim, E., Correa, J. D., Ibeling, D., and Icard, T. On Pearl's Hierarchy and the Foundations of Causal In-

ference. In *Probabilistic and Causal Inference*. ACM, 2022.

Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S., and Drouin, A. Differentiable Causal Discovery from Interventional Data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

Charpentier, B., Kibler, S., and Günnemann, S. Differentiable {DAG} Sampling. In *International Conference on Learning Representations*, 2022.

Chickering, D. M. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3, 2003. ISSN 1532-4435.

Cundy, C., Grover, A., and Ermon, S. BCD Nets: Scalable Variational Approaches for Bayesian Causal Discovery. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Cussens, J. Maximum likelihood pedigree reconstruction using integer programming. In *WCB@ ICLP*, pp. 8–19, 2010.

De Campos, C. P. and Ji, Q. Efficient structure learning of bayesian networks using constraints. *The Journal of Machine Learning Research*, 12:663–689, 2011.

Ellis, B. and Wong, W. H. Learning Causal Bayesian Network Structures from Experimental Data. *Journal of the American Statistical Association*, 2008.

Erdös, P. and Rényi, A. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018.

Geiger, D. and Heckerman, D. Learning gaussian networks. In *Uncertainty in Artificial Intelligence*, pp. 235–243. Elsevier, 1994.

Giudice, E., Kuipers, J., and Moffa, G. A Bayesian Take on Gaussian Process Networks. In *Advances in Neural Information Processing Systems*, 2023.

Giudice, E., Kuipers, J., and Moffa, G. Bayesian causal inference with gaussian process networks. *arXiv preprint arXiv:2402.00623*, 2024.

Heckerman, D. A bayesian approach to learning causal networks. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

Heckerman, D., Meek, C., and Cooper, G. A Bayesian Approach to Causal Discovery. *Computation, Causation, and Discovery*, 1997.

Henckel, L., Würtzen, T., and Weichwald, S. Adjustment Identification Distance: A gadjid for Causal Structure Learning. *arXiv:2402.08616*, 2024.

Hinton, G. Neural networks for machine learning - lecture 6a - overview of mini-batch gradient descent, 2012. https://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf.

Horii, S. Bayesian Model Averaging for Causality Estimation and its Approximation based on Gaussian Scale Mixture Distributions. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In And, Y. B. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, Conference Track Proceedings*, San Diego, CA, USA, 2015.

Koivisto, M. and Sood, K. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 2004.

Koller, D. and Friedman, N. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 2003.

Kool, W., Van Hoof, H., and Welling, M. Stochastic Beams and Where To Find Them: The {G}umbel-Top-k Trick for Sampling Sequences Without Replacement. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 2019. URL https://proceedings.mlr.press/v97/kool19a.html.

Kuipers, J. and Moffa, G. Partition MCMC for Inference on Acyclic Digraphs. *Journal of the American Statistical Association*, 2017.

Kuipers, J., Moffa, G., Kuipers, E., Freeman, D., and Bebbington, P. Links between psychotic and neurotic symptoms in the general population: an analysis of longitudinal british national survey data using directed acyclic graphs. *Psychological Medicine*, 49(3):388–395, 2019.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-Based Neural DAG Learning. In *International Conference on Learning Representations*, 2020.

Lam, W.-Y., Andrews, B., and Ramsey, J. Greedy relaxations of the sparsest permutation algorithm. In Cussens, J. and Zhang, K. (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, Proceedings of Machine Learning Research. PMLR, 2022.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 29–37. JMLR Workshop and Conference Proceedings, 2011.

Lorch, L., Rothfuss, J., Schölkopf, B., and Krause, A. DiBS: Differentiable Bayesian Structure Learning. *Advances in Neural Information Processing Systems*, 2021.

Madigan, D., York, J., and Allard, D. Bayesian Graphical Models for Discrete Data. *International Statistical Review / Revue Internationale de Statistique*, 63, 1995. ISSN 03067734. doi: 10.2307/1403615.

Moffa, G., Catone, G., Kuipers, J., Kuipers, E., Freeman, D., Marwaha, S., Lennox, B. R., Broome, M. R., and Bebbington, P. Using directed acyclic graphs in epidemiological research in psychosis: an analysis of the role of bullying in psychosis. *Schizophrenia bulletin*, 43(6): 1273–1279, 2017.

Murphy, K. P. Active learning of causal Bayes net structure. Technical report, Department of Computer Science, U.C. Berkeley, 2001.

Murphy, K. P. Conjugate Bayesian Analysis of the Gaussian Distribution. Technical report, University of British Columbia, 2007.

Murphy, K. P. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

Ng, I., Ghassami, A., and Zhang, K. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

Niinimäki, T., Parviainen, P., and Koivisto, M. Structure Discovery in Bayesian Networks by Sampling Partial Orders. *Journal of Machine Learning Research*, 2016.

OEIS Foundation Inc. Number of acyclic digraphs (or dags) with n labeled nodes, 2024. Entry A003024 in The On-Line Encyclopedia of Integer Sequences, https://oeis.org/A003024.

Pawlowski, N., Castro, D. C., and Glocker, B. Deep Structural Causal Models for Tractable Counterfactual Inference. In *Advances in Neural Information Processing Systems*, 2020.

Pearl, J. *Causality*. Cambridge University Press, 2009. ISBN 9780511803161.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Peharz, R. and Pernkopf, F. Exact maximum margin structure learning of bayesian networks. *arXiv preprint arXiv:1206.6431*, 2012.

Pensar, J., Talvitie, T., Hyttinen, A., and Koivisto, M. A bayesian approach for estimating causal effects from observational data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5395–5402, 2020.

Peters, J. and Bühlmann, P. Structural Intervention Distance for Evaluating Causal Graphs. *Neural Computation*, 27, 2015. ISSN 0899-7667. doi: 10.1162/NECO_a_00708.

Pleiss, G., Gardner, J., Weinberger, K., and Wilson, A. G. Constant-time predictive distributions for Gaussian processes. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2018.

Reisach, A. G., Seiler, C., and Weichwald, S. Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy To Game. In Vaughan, M. R., Beygelzimer, A., Dauphin, Y., Liang, P., and Wortman, J. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021.

Robert, C. P. and Roberts, G. O. Rao-blackwellization in the mcmc era. *arXiv preprint arXiv:2101.01011*, 2021.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science*, 308, 2005. doi: 10.1126/science.1105809.

Spirtes, P., Glymour, C., and Scheines, R. *Causation, Prediction, and Search*. The MIT Press, second edition, 2000. ISBN 9780262527927.

Teyssier, M. and Koller, D. Ordering-Based Search: A Simple and Effective Algorithm for Learning Bayesian Networks. *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, UAI 2005*, 2012.

Tigas, P., Annadani, Y., Jesson, A., Schölkopf, B., Gal, Y., and Bauer, S. Interventions, Where and How? Experimental Design for Causal Models at Scale. In Koyejo, S.,

Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022.

Tong, S. and Koller, D. Active learning for structure in Bayesian networks. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608125.

Toth, C., Lorch, L., Knoll, C., Krause, A., Pernkopf, F., Peharz, R., and von Kügelgen, J. Active Bayesian Causal Inference. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022.

Viinikka, J., Hyttinen, A., Pensar, J., and Koivisto, M. Towards Scalable Bayesian Learning of Causal DAGs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.

von Kügelgen, J., Rubenstein, P. K., Schölkopf, B., and Weller, A. Optimal experimental design via Bayesian optimization: active causal structure learning for Gaussian process networks. *arXiv:1910.03962*, 2019.

Yu, Y., Chen, J., Gao, T., and Yu, M. DAG-GNN: DAG Structure Learning with Graph Neural Networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2019.

Zhang, K., Zhu, S., Kalander, M., Ng, I., Ye, J., Chen, Z., and Pan, L. gcastle: A python toolbox for causal discovery. *arXiv:2111.15155*, 2021.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

Zheng, Y., Huang, B., Chen, W., Ramsey, J., Gong, M., Cai, R., Shimizu, S., Spirtes, P., and Zhang, K. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25:1–8, 2024.

A. Proofs and Derivations

A.1. Derivation of the Posterior Expectation w.r.t. SCMs

In the following, we derive the expectation w.r.t. SCMs in Equation (3) and the corresponding importance weights in Equation (4).

$$\begin{aligned}
\mathbb{E}_{\mathcal{M} \mid \mathcal{D}}\left[p(Y \mid \mathcal{M})\right] &= \mathbb{E}_{G,\mathbf{f},\boldsymbol{\psi} \mid \mathcal{D}}\left[p(Y \mid \mathcal{M})\right] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{G,\mathbf{f} \mid \boldsymbol{\theta},\boldsymbol{\psi},\mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{G \mid \boldsymbol{\theta},\boldsymbol{\psi},\mathcal{D}}\left[\mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi},\mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{L \mid \boldsymbol{\theta},\boldsymbol{\psi},\mathcal{D}}\left[\mathbb{E}_{G \mid L,\boldsymbol{\psi},\mathcal{D}}\left[\mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi},\mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right]\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\frac{p(\mathcal{D},\boldsymbol{\psi} \mid L)}{p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta})} \cdot \mathbb{E}_{G \mid L,\boldsymbol{\psi},\mathcal{D}}\left[\mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi},\mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right]\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}}\left[\mathbb{E}_{L \mid \boldsymbol{\theta}}\left[w^L \cdot \mathbb{E}_{G \mid L,\boldsymbol{\psi},\mathcal{D}}\left[\mathbb{E}_{\mathbf{f} \mid \boldsymbol{\psi},\mathcal{D}}\left[p(Y \mid \mathcal{M})\right]\right]\right]\right]
\end{aligned}$$

with

$$\begin{aligned}
w^L &:= \frac{p(\mathcal{D},\boldsymbol{\psi} \mid L)}{p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta})} \\
&= \frac{\mathbb{E}_{G \mid L}\left[p(\mathcal{D},\boldsymbol{\psi} \mid G)\right]}{p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta})} \\
&= \frac{\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]}{p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta})} \\
&= \frac{\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]}{\mathbb{E}_{L' \mid \boldsymbol{\theta}}\left[p(\mathcal{D},\boldsymbol{\psi} \mid L')\right]} \\
&= \frac{\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]}{\mathbb{E}_{L' \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G' \mid L'}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G') \cdot p(\boldsymbol{\psi} \mid G')\right]\right]}.
\end{aligned}$$

A.2. Derivation of the Gradient Estimators

In the following, we derive the gradient estimators in Equations (5), (6) and (11). We denote by $\nabla = \nabla_{\boldsymbol{\theta},\boldsymbol{\psi}}$ to avoid clutter.

General Posterior Gradient. The general posterior gradient in Equation (5) reads as follows.

$$\begin{aligned}
\nabla \log p(\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}) &= \nabla \log \frac{p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})}{p(\mathcal{D})} \\
&= \nabla \log p(\boldsymbol{\theta}) + \nabla \log p(\mathcal{D},\boldsymbol{\psi} \mid \boldsymbol{\theta}) \\
&= \nabla \log p(\boldsymbol{\theta}) + \nabla \log \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[p(\mathcal{D},\boldsymbol{\psi} \mid L)\right] \\
&= \nabla \log p(\boldsymbol{\theta}) + \nabla \log \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G \mid L}\left[p(\mathcal{D},\boldsymbol{\psi} \mid G)\right]\right] \\
&= \nabla \log p(\boldsymbol{\theta}) + \nabla \log \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]\right]
\end{aligned}$$

ARCO Gradient. Using the above as starting point for the gradient in Equation (6), we get

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta},\boldsymbol{\psi} \mid \mathcal{D}) &= \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]\right] \\
&= \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \frac{\nabla_{\boldsymbol{\theta}} \mathbb{E}_{L \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G \mid L}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G)\right]\right]}{\mathbb{E}_{L' \mid \boldsymbol{\theta}}\left[\mathbb{E}_{G' \mid L'}\left[p(\mathcal{D} \mid \boldsymbol{\psi}, G') \cdot p(\boldsymbol{\psi} \mid G')\right]\right]}
\end{aligned}$$

and using $\nabla_{\boldsymbol{\theta}} p(L \mid \boldsymbol{\theta}) = p(L \mid \boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}} \log p(L \mid \boldsymbol{\theta})$

$$
\begin{aligned}
&= \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[p(\mathcal{D} \mid \boldsymbol{\psi}, G) \cdot p(\boldsymbol{\psi} \mid G) \right] \cdot \nabla_{\boldsymbol{\theta}} \log p(L \mid \boldsymbol{\theta}) \right]}{\mathbb{E}_{L' \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G' \mid L'} \left[p(\mathcal{D} \mid \boldsymbol{\psi}, G') \cdot p(\boldsymbol{\psi} \mid G') \right] \right]} \\
&= \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) + \mathbb{E}_{L \mid \boldsymbol{\theta}} \left[w^L \cdot \nabla_{\boldsymbol{\theta}} \log p(L \mid \boldsymbol{\theta}) \right]
\end{aligned}
$$

with w^L as defined in Equation (4).

GP Hyper-parameter Gradient. For the gradient of a distinct GP modeling the mechanism from parents $\mathbf{Pa}_k$ to target node X_k with corresponding hyper-parameters $\boldsymbol{\psi}_k$ as in Equation (11) we have

$$
\begin{aligned}
\nabla_{\boldsymbol{\psi}_k} \log p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \mathcal{D}) &= \nabla_{\boldsymbol{\psi}_k} \log p(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\psi}_k} \log \mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[p(\mathcal{D}, \boldsymbol{\psi} \mid G) \right] \right] \\
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[\nabla_{\boldsymbol{\psi}_k} p(\mathcal{D}, \boldsymbol{\psi} \mid G) \right] \right]}{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[p(\mathcal{D}, \boldsymbol{\psi} \mid G) \right] \right]} \\
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[\nabla_{\boldsymbol{\psi}_k} p(\mathcal{D}, \boldsymbol{\psi} \mid G) \right] \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})}
\end{aligned}
$$

and as the marginal likelihood and the prior over GP hyper-parameters factorise over parent sets we further get

$$
\begin{aligned}
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\mathbb{E}_{G \mid L} \left[\nabla_{\boldsymbol{\psi}_k} \prod_{i=1}^{d} p(\mathcal{D}_i, \boldsymbol{\psi}_i \mid \mathbf{Pa}_i^G) \right] \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})} \\
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\sum_G p(G \mid L) \cdot \nabla_{\boldsymbol{\psi}_k} \prod_{i=1}^{d} p(\mathcal{D}_i, \boldsymbol{\psi}_i \mid \mathbf{Pa}_i^G) \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})}.
\end{aligned}
$$

Now, note that for the summation over graphs G, the gradient is zero for all graphs that do not contain the parent set $\mathbf{Pa}_k$ corresponding to the GP with hyper-parameters ψ_k. Consequently, we get

$$
= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\sum_{G \mid \mathbf{Pa}_k \in G} p(G \mid L) \cdot \nabla_{\boldsymbol{\psi}_k} \prod_{i=1}^{d} p(\mathcal{D}_i, \boldsymbol{\psi}_i \mid \mathbf{Pa}_i^G) \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})}
$$

and since $\nabla_{\boldsymbol{\psi}_k} p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k) = p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k) \cdot \nabla_{\boldsymbol{\psi}_k} \log p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k)$

$$
\begin{aligned}
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\sum_{G \mid \mathbf{Pa}_k \in G} p(G \mid L) \cdot \prod_{i=1}^{d} p(\mathcal{D}_i, \boldsymbol{\psi}_i \mid \mathbf{Pa}_i^G) \cdot \nabla_{\boldsymbol{\psi}_k} \log p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k) \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})} \\
&= \frac{\mathbb{E}_{L \mid \boldsymbol{\theta}} \left[\sum_{G \mid \mathbf{Pa}_k \in G} p(G \mid L) \cdot p(\mathcal{D}, \boldsymbol{\psi} \mid G) \right]}{p(\mathcal{D}, \boldsymbol{\psi} \mid \boldsymbol{\theta})} \cdot \nabla_{\boldsymbol{\psi}_k} \log p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k).
\end{aligned}
$$

Note that the term preceding the gradient $\nabla_{\boldsymbol{\psi}_k} \log p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k)$ is a scalar factor that *does not* influence the *direction* of the gradient and can thus be practically omitted for gradient-based optimisation, as optimisation algorithms will scale the gradient depending on tune-able step size parameters anyways. Therefore, optimising the GP hyper-parameters w.r.t. the gradient in Equation (5) practically yields the same gradient direction as the common MAP type II gradient $\nabla_{\boldsymbol{\psi}_k} \log p(\mathcal{D}_k, \boldsymbol{\psi}_k \mid \mathbf{Pa}_k)$. Arguably, for mechanism models that do not decompose over individual mechanisms for each parent set and target variable, naively estimating the gradient in Equation (5) may yield very noisy gradients, as the magnitude of the estimated gradient will depend on the sampled structures used for its estimation.

A.3. Proofs regarding Exhaustive Parent Set Enumeration

Proof of Proposition 3.1

Proof.

$$\mathbb{E}_{G \mid L}\left[w(G) \cdot h(G)\right] = \sum_{G} p(G \mid L) \cdot w(G) \cdot h(G)$$

Since we assume that $w(G)$ and $h(G)$ factorise over the parent sets, we have

$$= \sum_{G} \prod_{i} p(\mathbf{pa}_i^G \mid L) \cdot w_i(\mathbf{pa}_i^G) \cdot h_i(\mathbf{pa}_i^G)$$

The sum over all graphs can be represented as sum over all combinations of possible parent sets to get

$$\begin{aligned}
&= \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \prod_{i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) \cdot h_i(\mathbf{pa}_i) \\
&= \sum_{\mathbf{pa}_1} p(\mathbf{pa}_1 \mid L) \cdot w_1(\mathbf{pa}_i) \cdot h_1(\mathbf{pa}_i) \sum_{\mathbf{pa}_2} p(\mathbf{pa}_2 \mid L) \cdot w_2(\mathbf{pa}_i) \cdot h_2(\mathbf{pa}_i) \ldots
\end{aligned}$$

Since each summation over parent sets is independent of the others, we get the final result

$$= \prod_{i} \sum_{\mathbf{pa}_i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) \cdot h_i(\mathbf{pa}_i)$$

□

Proof of Proposition 3.2

Proof.

$$\begin{aligned}
&\mathbb{E}_{G \mid L}\left[w(G) \cdot h(G)\right] \\
&\quad = \sum_{G} p(G \mid L) \cdot w(G) \cdot h(G)
\end{aligned}$$

Since we assume that $w(G)$ factorises and $h(G)$ sums over the parent sets, we have

$$= \sum_{G} \prod_{i} p(\mathbf{pa}_i^G \mid L) \cdot w_i(\mathbf{pa}_i^G) \cdot \sum_{j} h_j(\mathbf{pa}_j^G)$$

The sum over all graphs can be represented as sum over all combinations of possible parent sets to get

$$\begin{aligned}
&= \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \prod_{i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) \cdot \sum_{j} h_j(\mathbf{pa}_j) \\
&= \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \cdot \sum_{j} h_j(\mathbf{pa}_j) \prod_{i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) \\
&= \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} h_1(\mathbf{pa}_1) \prod_{i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) + \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \sum_{j=2}^{d} \ldots \\
&= \sum_{\mathbf{pa}_1} h_1(\mathbf{pa}_1) \cdot p(\mathbf{pa}_1 \mid L) \cdot w_1(\mathbf{pa}_1) \sum_{\mathbf{pa}_2} p(\mathbf{pa}_2 \mid L) \cdot w_2(\mathbf{pa}_2) \cdots + \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \sum_{j=2}^{d} \ldots
\end{aligned}$$

By abbreviating $\alpha_i(L) = \sum_{\mathbf{pa}_i} p(\mathbf{pa}_i \mid L) w_i(\mathbf{pa}_i)$ we get

$$= \sum_{\mathbf{pa}_1} h_1(\mathbf{pa}_1) \cdot p(\mathbf{pa}_1 \mid L) \cdot w_1(\mathbf{pa}_1) \cdot \prod_{k \neq 1} \alpha_k(L) + \sum_{\mathbf{pa}_1} \sum_{\mathbf{pa}_2} \cdots \sum_{\mathbf{pa}_d} \sum_{j=2}^{d} h_j(\mathbf{pa}_j) \prod_i p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i)$$

By repeating this procedure for the remaining summands j, we get the final result

$$= \sum_i \left(\prod_{k \neq i} \alpha_k(L) \right) \cdot \sum_{\mathbf{pa}_i} p(\mathbf{pa}_i \mid L) \cdot w_i(\mathbf{pa}_i) \cdot h_i(\mathbf{pa}_i)$$

□

B. Implementation and Estimation Details

Our model is implemented in Python mainly relying on PyTorch (Ansel et al., 2024), and GPyTorch (Gardner et al., 2018) for GP inference.

B.1. Mechanism Model.

We follow Toth et al. (2022) and use two types of models for our mechanism.

For root nodes, i.e., causal variables without parents, we place a conjugate normal-inverse-gamma prior N-Γ^{-1} on the mean and variance of that node. Specifically,

$$\begin{aligned} p(f_i, \sigma_i^2 \mid \mathbf{Pa}_i = \varnothing) &= \text{N-}\Gamma^{-1}(f_i, \sigma_i^2 \mid \mu_0, \kappa_0^{-1}, \alpha_0, \beta_0) \\ &= \text{N}(f_i \mid \mu_0, \sigma_i^2 \cdot \kappa_0^{-1}) \cdot \Gamma^{-1}(\sigma_i^2 \mid \alpha_0, \beta_0) \end{aligned}$$

where $\mu_0, \kappa_0^{-1}, \alpha_0, \beta_0$ are fixed hyper-parameters. When sampling ground-truth SCMs we set $\mu_0 = 0$, $\kappa_0 = 1$, $\alpha_0 = 5$ and $\beta_0 = 10$, sample a mean and variance from the prior and keep them fixed thereafter. For the inference with ARCO-GP and DIBS-GP, we use $\mu_0 = 0$, $\kappa_0 = 1$, $\alpha_0 = 10$ and $\beta_0 = 10$. This yields a prior mean of 1 for the variance σ^2, which is sufficiently broad considering that we standardise all training data to zero mean and unit variance. Analytic expressions for the posterior marginal likelihood are found in (Murphy, 2007).

For non-root nodes we place a GP prior on the mechanisms. Specifically, we use rational quadratic (RQ) kernel

$$k_{\text{RQ}}(\mathbf{x_1}, \mathbf{x_2}) = \delta \cdot \left(1 + \frac{1}{2\gamma} (\mathbf{x_1} - \mathbf{x_2})^\top \lambda^{-2} (\mathbf{x_1} - \mathbf{x_2}) \right)^{-\gamma}$$

with scaling parameter δ, lengthscale parameter λ and mixing parameter γ to model non-linear mechanisms. For non-linear mechanisms we use an additive Gaussian noise likelihood with variance σ_i^2. We place Gamma priors $\Gamma(\delta \mid \alpha_\delta, \beta_\delta)$, $\Gamma(\lambda \mid \alpha_\lambda, \beta_\lambda)$, $\Gamma(\gamma \mid \alpha_\gamma, \beta_\gamma)$ and $\Gamma(\sigma_i^2 \mid \alpha_\sigma, \beta_\sigma)$ on these parameters.

When generating non-linear models we set $\alpha_\delta = 100$, $\beta_\delta = 10$, $\alpha_\lambda = 30 \cdot |\mathbf{Pa}_i|$, $\beta_\lambda = 30$, $\alpha_\gamma = 20$, $\beta_\gamma = 10$ and $\alpha_\sigma = 50$, $\beta_\sigma = 50$. For each GP we sample a set of hyper-parameters from their priors and keep them fixed thereafter. Additionally, we sample a function from the GP prior with 50 support points sampled uniform random in range [-10, 10]. For the inference with ARCO-GP and DIBS-GP, we set $\alpha_\delta = 100$, $\beta_\delta = 10$, $\alpha_\lambda = 30 \cdot |\mathbf{Pa}_i|$, $\beta_\lambda = 30$, $\alpha_\gamma = 20$, $\beta_\gamma = 10$ and $\alpha_\sigma = 2$, $\beta_\sigma = 88$ again considering that we standardise all training data to zero mean and unit variance. We train GP hyper-parameters for a maximum of 100 steps with the RMSprop (Hinton, 2012) optimiser with learning rate 0.05.

B.2. ARCO Model.

We use a normal prior $\mathcal{N}(\boldsymbol{\theta} \mid 0, \sigma^2 I)$ with $\sigma = 10$ over ARCO's neural network parameters $\boldsymbol{\theta}$. The neural network $g_{\boldsymbol{\theta}}$ uses a single hidden layer with 30 neurons and ReLU activation functions (see Section 3.1). We train ARCO for a maximum of

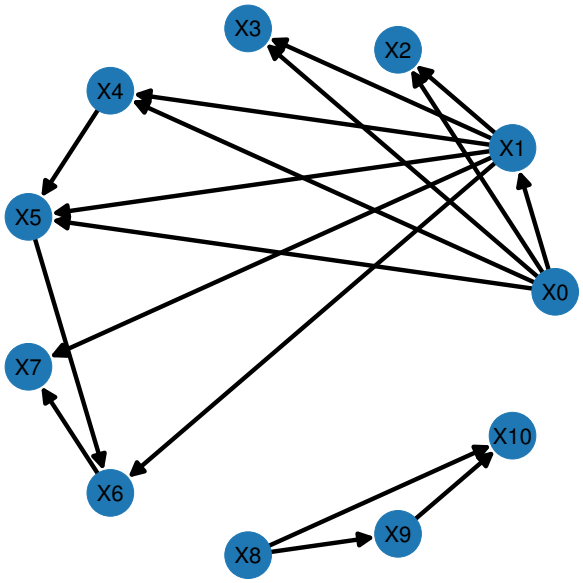

Figure 4. **Sachs Graph.** Consensus protein interaction graph from (Sachs et al., 2005). We relabeled nodes to avoid misinterpretation of our simulation results. Nodes X0 to X10 correspond to the original labels ['PKC', 'PKA', 'Jnk', 'P38', 'Raf', 'Mek', 'Erk', 'Akt', 'Plcg', 'PIP3', 'PIP2'].

400 gradient steps, using the ADAM (Kingma & Ba, 2015) optimiser with learning rate of 0.01. For the score-function gradient estimator in Equation (6) we use an exponential moving average baseline with decay rate 0.9. To estimate gradients and causal queries we sample 100 causal orders and 25 DAGs conditional on each order where necessary, resulting in 2500 DAG samples. Per default, we limit the parent set size to a maximum of two parents per node. These settings apply to all our simulations unless stated otherwise.

B.3. Kernel Density Estimates of Interventional Distributions.

To approximate the interventional distribution given our posterior samples (see Section 3, Algorithm 1, lines 6-11), we compute a kernel density estimate (KDE). We draw 100 causal orders, with 10 graphs per order and 10 samples per graph, equalling 10000 samples from the posterior interventional distribution in total. We use the KDE implementation provided by scikit-learn (Pedregosa et al., 2011), with a Gaussian kernel and a bandwidth of 0.2.

C. Experimental Setup

Our code is available at https://github.com/chritoth/bci-arco-gp.

Sampling of Ground-Truth SCMs. To sample ground truth SCMs we draw graph structures from either the Erdös-Rényi (ER) (Erdös & Rényi, 1959) or the scale-free (SF) (Barabási & Albert, 1999) random models commonly used in DAG structure learning benchmarks. We follow the setup of Lorch et al. (2021) and generate scale-free and Erdös-Rényi graphs with an expected degree of 2. Specifically, this will yield SF graphs with a maximum parent set size of 2, whereas no such restriction applies to ER graphs in general. We instantiate the causal mechanisms for non-linear additive noise models as described in Appendix B.

For each ground truth SCM, we sample a fixed set of N training data from the observational distribution. Importantly, Reisach et al. (2021) argue, that the causal order may strongly correlate with increasing marginal variance in simulated data, and therefore, benchmarks may be easy to game by exploiting this property. As this may be especially relevant to order-based inference methods, we follow their recommendation and normalise the training data for each endogenous variable to zero mean and unit variance.

Metrics. As metrics for learning the causal structure we report the *expected structural Hamming distance* (ESHD), as well as the *area under the receiver operating characteristic* (AUROC) and the *area under the precision recall curve* (AUPRC) w.r.t. posterior edge prediction as commonly reported metrics. To provide additional insight into the methods' strengths and weaknesses, we also report the expected number of edges (#Edges), the *true positive rate* (TPR) and the *true negative rate* (TNR) for the edge prediction task. We do **not** report the log-likelihood on held-out test data as is sometimes reported, because the evaluated methods implement different noise models and approximate inference schemes, which leads to the (marginal) log-likelihoods being uncalibrated and thus incomparable (cf. (Murphy, 2023)[Sec. 7.5] and references therein).

To evaluate the inferred causal structures w.r.t. their causal implications, Henckel et al. (2024) recently proposed a family of causal distances called *Adjustment Identification Distance* (AID). Briefly summarised, the AID counts the number of correctly identified adjustment sets for pairwise causal effects w.r.t. a target graph, where the adjustment sets are determined according to a chosen identification strategy. We adopt the three variants proposed by Henckel et al. (2024), namely *Ancestor* adjustment (A-AID), *Parent* adjustment (P-AID), and *Optimal* adjustment (OSET-AID). These metrics can also be applied between different graph types, e.g., comparing the AID between a predicted CPDAG and a reference DAG. Note that the (P-AID) computed between to DAGs is equivalent to the well-known Structural Identification Distance proposed by Peters & Bühlmann (2015).

Baselines. We compare our inference model (ARCO-GP) to a diverse set of nine different structure learning methods that we describe in the following.

- **BAYESDAG** (Annadani et al., 2023). BayesDAG utilises a mixture of MCMC to infer permutations and mechanism parameters, and Variational Inference (VI) to infer DAG edges given the permutations. We use the implementation provided by Annadani et al. (2023) at https://github.com/microsoft/Project-BayesDAG. We needed to adapt the sparsity regularisation hyper-parameter in order to get meaningful results and ran our experiments with the configurations in Listings 2-3. As the implementation runs a number of MCMC chains and only evaluates the best chain afterwards, we use only one MCMC chain to enable a fair comparison, as multiple chains would correspond to multiple runs of the other methods.
- **DAG-GNN** (Yu et al., 2019). DAG-GNN is a gradient-based structure learning approach combining graph neural networks an acyclicity constraint similar to Zheng et al. (2018). We use the implementation provided by Zhang et al. (2021)[Version 1.0.3] using default settings.
- **DDS** (Charpentier et al., 2022). DDS builds upon the permutation-based approach of (Cundy et al., 2021) and utilises differentiable permutation sampling and VI to infer a posterior over DAGs. We use the implementation provided by Charpentier et al. (2022) at https://github.com/sharpenb/Differentiable-DAG-Sampling. We needed to adapt the default hyper-parameters in order to get meaningful results and ran our experiments with the configuration displayed in Listing 1.
- **DIBS-GP** (Toth et al., 2022; Lorch et al., 2021). DIBS-GP is a Bayesian causal inference framework recently proposed by Toth et al. (2022) utilising the differential structure learning method by Lorch et al. (2021) employing a soft acyclicity constraint in line with Zheng et al. (2018); Yu et al. (2019). We use the implementation provided by Toth et al. (2022) at https://github.com/chritoth/active-bayesian-causal-inference. We use their standard parameters with 10 latent particles and constant hyper-parameters $\alpha = \beta = 1$ as described in (Lorch et al., 2021). For each latent particle we sample 100 graphs to estimate gradients and causal quantities.
- **GADGET** (Viinikka et al., 2020). GADGET is a Bayesian structure learning method for linear Gaussian models based on MCMC and structure marginalisation over capped-size parent sets. We use the implementation provided by (Viinikka et al., 2020) at https://github.com/Sums-of-Products/sumu and parameters as shown in Listing 4.
- **GES** (Chickering, 2003). GES is a well-known greedy score-based method for causal discovery using the BIC score. We use the implementation provided by Zhang et al. (2021)[Version 1.0.3] using default settings.
- **GOLEM** (Ng et al., 2020). GOLEM is a differentiable DAG structure learning similar to Zheng et al. (2018) but with a likelihood-based score function for linear models. We use the implementation provided by Zhang et al. (2021)[Version 1.0.3] using default settings.
- **GRASP** (Lam et al., 2022). GRASP is a recently proposed permutation-based approach to causal discovery. We use the implementation provided by Zheng et al. (2024) using default settings.

- **PC** (Spirtes et al., 2000). The PC algorithm is another standard causal discovery methods based on conditional independence tests. We use the implementation provided by Zhang et al. (2021)[Version 1.0.3] using default settings. It happens quite frequently that PC return a possibly cyclic PDAG, for which the AID metrics cannot computed and are thus omited in our experimental results.

Listing 1. DDS Parameter Set.

```
# Architecture parameters
'seed_model': 123,  # Seed to init model.
'ma_hidden_dims': [32, 32, 32],  # Output dimension.
'ma_architecture': 'linear',  # Output dimension.
'ma_fast': False,  # Output dimension.
'pd_initial_adj': 'Learned',  # Output dimension.
'pd_temperature': 1.0,  # Output dimension.
'pd_hard': True,  # Output dimension.
'pd_order_type': 'topk',  # Output dimension.
'pd_noise_factor': 1.0,  # Hidden dimensions.

# Training parameters
'max_epochs': 500,  # Maximum number of epochs for training
'patience': 150,  # Patience for early stopping.
'frequency': 2,  # Frequency for early stopping test.
'batch_size': 16,  # Batch size.
'ma_lr': 1e-3,  # Learning rate.
'pd_lr': 1e-2,  # Learning rate.
'loss': 'ELBO',  # Loss name. string
'regr': 1e-1,  # Regularization factor in Bayesian loss.
'prior_p': 1e-6  # Regularization factor in Bayesian loss.
```

Listing 2. BAYESDAG Non-Linear ER Parameter Set.

```
"model_hyperparams": {
    "num_chains": 1,
    "sinkhorn_n_iter": 3000,
    "scale_noise_p": 0.001,
    "scale_noise": 0.001,
    "VI_norm": true,
    "input_perm": false,
    "lambda_sparse": 10,
    "sparse_init": false
},
"training_hyperparams": {
    "learning_rate": 1e-3,
    "batch_size": 512,
    "stardardize_data_mean": false,
    "stardardize_data_std": false,
    "max_epochs": 500
}
```

Listing 3. BAYESDAG Non-linear SF Parameter Set.

```
    "model_hyperparams": {
        "num_chains": 1,
        "sinkhorn_n_iter": 3000,
        "scale_noise_p": 0.001,
        "scale_noise": 0.001,
        "VI_norm": true,
        "input_perm": false,
        "lambda_sparse": 10,
        "sparse_init": false
    },
    "training_hyperparams": {
        "learning_rate": 1e-3,
        "batch_size": 512,
        "stardardize_data_mean": false,
        "stardardize_data_std": false,
        "max_epochs": 500
    }
```

Listing 4. GADGET Parameter Set.

```
    "scoref": 'bge',
    "max_id": -1,
    "K": min(self.num_nodes - 1, 16),
    "d": 2,
    "cp_algo": 'greedy-lite',
    "mc3_chains": 16,
    "burn_in": 1000,
    "iterations": 1000,
    "thinning": 10,
```

D. Extended Experimental Results

Here we present additional experimental results and ablations with varying number of variables $d \in \{11, 20, 50\}$, as well as experiments on real-world data from Sachs et al. (2005). For these extended experiments we report an extended set of eight metrics as described in Appendix C.

Real-world dataset from Sachs et al. (2005). The dataset consists of 853 observational data. The target consensus graph has 11 nodes and 17 edges. We compare our ARCO-GP method in variants with maximum parent set size $k \in 2, 3$. All variants are competitive with the baselines.

Simulations on larger models with $d = 50$ variables. We report structure learning results for non-linear models with ER and SF graphs over $d = 50$ variables in Table 3. The results of ARCO-GP are qualitatively similar to the ones achieved on 20 node systems.

Influence of the maximum parent set size. We evaluate the influence of the maximum parent set size in range $k \in \{1, 2, 3, 4\}$ on the performance of our ARCO-GP method in Table 2. We distinguish the models variants by labels ARCO-Kk-GP. On non-linear benchmarks, $k = 1$ performs worst as expected, since each node can only have one parent. Not surprisingly, $k = 2$ performs best on scale-free graphs, whereas $k = 4$ performs best on Erdös-Rényi graphs, as the respective k better fit the stuctural properties of the true graphs.

Non-linear Average Causal Effect Estimation. In this simulation, we compare our ability to infer pairwise average causal effects (ACEs)

$$ACE(X_i \,|\, do(X_j = x_j)) := \mathbb{E}_{X_i \,|\, do(X_j = x_j), \mathcal{D}}\left[X_i\right]$$

to two baselines, namely DIBS-GP ((Toth et al., 2022; Lorch et al., 2021)) and GADET+BEEPS ((Viinikka et al., 2020)). While DIBS-GP can implements a non-linear additive noise model based on GPs, GADGET+BEEPS is designed for linear Gaussian models only. In particular, the implementation of GADGET+BEEPS does not naturally support the inference of average causal effects, but infers causal effects as the path-wise accumulation of the linear mechanism weights. Thus, we extend the implementation in the following way. A linear SCM can be written as

$$X = (G \odot B)^T \cdot X + \mu + \epsilon,$$

where X are the endogenous variables, G is the adjacency matrix of the causal DAG, B is the weight matrix of the linear edge weights, $\odot$ denotes the element-wise multiplication, μ denotes the mean vector of the variables and ϵ denotes the exogenous Gaussian noise. Solving for X yields

$$X = (I - (G \odot B)^T)^{-1}(\mu + \epsilon),$$

and taking the expectation w.r.t. ϵ yields

$$\mathbb{E}_\epsilon[X] = (I - (G \odot B)^T)^{-1}\mu. \tag{12}$$

We then estimate $ACE(X_i \,|\, do(X_j = x_j))$ by (i) drawing posterior DAGs G and weight matrices B using GADGET+BEEPS, (ii) performing the intervention by removing all parents of X_j in the sampled DAGs and setting the posterior mean $\mu_j = x_j$, and (iii) finally solving Equation (12) to get the desired ACE.

We generate 10 non-linear ground truth models with fixed graph as in Figure 4. We then report the mean absolute errors (MAEs) between for all pairwise ACEs in Table 4 for ARCO-GP, in Table 5 for DIBS-GP, and in Table 6 for GADGET+BEEPS.

ARCO-GP achieves better MAEs than DIBS-GP for almost all pairwise ACEs. Not surprisingly, ARCO-GP performs better than GADGET+BEEPS on non-linear models.

Table 1. **Real-world dataset from Sachs et al. (2005)).** The target graph has 11 nodes and 17 edges. We report means and 95% confidence intervals (CIs) across 20 different ground truth models. Arrows next to metrics indicate lower is better (↓) and higher is better (↑).

Model	#Edges	↓ A-AID	↓ P-AID	↓ OSET-AID
ARCO-K2-GP	7 ± 0	0.22 ± 0.01	0.48 ± 0.02	0.23 ± 0.01
ARCO-K3-GP	7 ± 1	0.22 ± 0.00	0.47 ± 0.02	0.22 ± 0.00
BAYESDAG	24 ± 1	0.33 ± 0.02	0.48 ± 0.02	0.33 ± 0.02
DAG-GNN	7 ± 0	0.23 ± 0.00	0.45 ± 0.00	0.23 ± 0.00
DDS	39 ± 1	0.36 ± 0.01	0.41 ± 0.01	0.36 ± 0.01
DIBS-GP	7 ± 1	0.22 ± 0.01	0.43 ± 0.04	0.22 ± 0.01
GADGET	9 ± 0	0.25 ± 0.00	0.50 ± 0.01	0.25 ± 0.00
GES	8 ± 0	0.28 ± 0.00	0.56 ± 0.00	0.28 ± 0.00
GOLEM	17 ± 0	0.40 ± 0.00	0.49 ± 0.00	0.39 ± 0.00
GRASP	8 ± 0	0.28 ± 0.00	0.56 ± 0.00	0.28 ± 0.00
PC	8 ± 0	–	–	–

(a) **AID Metrics.**

Model	↓ ESHD	↑ AUROC	↑ AUPRC	↑ TPR	↑ TNR
ARCO-K2-GP	17 ± 1	0.56 ± 0.02	0.30 ± 0.02	0.20 ± 0.03	0.96 ± 0.01
ARCO-K3-GP	17 ± 0	0.55 ± 0.03	0.31 ± 0.02	0.21 ± 0.02	0.96 ± 0.00
BAYESDAG	34 ± 2	0.49 ± 0.04	0.18 ± 0.03	0.22 ± 0.06	0.78 ± 0.01
DAG-GNN	18 ± 0	0.57 ± 0.00	0.37 ± 0.00	0.18 ± 0.00	0.96 ± 0.00
DDS	45 ± 2	0.48 ± 0.01	0.19 ± 0.02	0.33 ± 0.00	0.64 ± 0.02
DIBS-GP	16 ± 1	0.61 ± 0.02	0.36 ± 0.04	0.24 ± 0.04	0.97 ± 0.01
GADGET	20 ± 0	0.63 ± 0.03	0.27 ± 0.02	0.19 ± 0.01	0.94 ± 0.00
GES	19 ± 0	0.63 ± 0.00	0.44 ± 0.00	0.35 ± 0.00	0.91 ± 0.00
GOLEM	26 ± 0	0.55 ± 0.00	0.29 ± 0.00	0.24 ± 0.00	0.86 ± 0.00
GRASP	19 ± 0	0.63 ± 0.00	0.44 ± 0.00	0.35 ± 0.00	0.91 ± 0.00
PC	19 ± 0	0.63 ± 0.00	0.44 ± 0.00	0.35 ± 0.00	0.91 ± 0.00

(b) **Edge prediction metrics.**

Table 2. **Varying the maximal parent set size.** Ablation studies on simulated (non-)linear ground truth models with 20 nodes. We report means and 95% confidence intervals (CIs) across 20 different ground truth models. Arrows next to metrics indicate lower is better (↓) and higher is better (↑).

Model	#Edges	↓ A-AID	↓ P-AID	↓ OSET-AID	↓ A-AID (CO)
ARCO-K1-GP	10 ± 1	0.17 ± 0.02	0.82 ± 0.02	0.19 ± 0.02	0.24 ± 0.05
ARCO-K2-GP	32 ± 1	0.06 ± 0.02	0.13 ± 0.04	0.07 ± 0.02	0.06 ± 0.03
ARCO-K3-GP	35 ± 1	0.09 ± 0.04	0.15 ± 0.04	0.10 ± 0.04	0.09 ± 0.05
ARCO-K4-GP	37 ± 1	0.08 ± 0.03	0.16 ± 0.05	0.10 ± 0.03	0.08 ± 0.04

(a) **Scale-free Nonlinear (Part 1/2).**

Model	↓ ESHD	↑ AUROC	↑ AUPRC	↑ TPR	↑ TNR
ARCO-K1-GP	26 ± 1	0.86 ± 0.02	0.67 ± 0.02	0.28 ± 0.03	1.00 ± 0.00
ARCO-K2-GP	5 ± 2	0.97 ± 0.02	0.94 ± 0.03	0.88 ± 0.04	1.00 ± 0.00
ARCO-K3-GP	9 ± 2	0.94 ± 0.03	0.89 ± 0.04	0.87 ± 0.04	0.99 ± 0.00
ARCO-K4-GP	11 ± 2	0.94 ± 0.02	0.90 ± 0.04	0.86 ± 0.04	0.98 ± 0.00

(b) **Scale-free Nonlinear (Part 2/2).**

Model	#Edges	↓ A-AID	↓ P-AID	↓ OSET-AID	↓ A-AID (CO)
ARCO-K1-GP	8 ± 1	0.21 ± 0.03	0.50 ± 0.03	0.21 ± 0.02	0.36 ± 0.06
ARCO-K2-GP	20 ± 2	0.15 ± 0.02	0.32 ± 0.04	0.18 ± 0.03	0.23 ± 0.05
ARCO-K3-GP	29 ± 2	0.12 ± 0.02	0.23 ± 0.03	0.16 ± 0.03	0.19 ± 0.05
ARCO-K4-GP	32 ± 3	0.10 ± 0.03	0.21 ± 0.04	0.15 ± 0.04	0.18 ± 0.06

(c) **Erdös-Rényi Nonlinear (Part 1/2).**

Model	↓ ESHD	↑ AUROC	↑ AUPRC	↑ TPR	↑ TNR
ARCO-K1-GP	32 ± 3	0.79 ± 0.03	0.54 ± 0.04	0.19 ± 0.03	1.00 ± 0.00
ARCO-K2-GP	21 ± 3	0.84 ± 0.03	0.68 ± 0.05	0.49 ± 0.05	1.00 ± 0.00
ARCO-K3-GP	17 ± 2	0.87 ± 0.02	0.76 ± 0.04	0.65 ± 0.05	0.99 ± 0.00
ARCO-K4-GP	17 ± 3	0.88 ± 0.03	0.79 ± 0.04	0.70 ± 0.05	0.99 ± 0.00

(d) **Erdös-Rényi Nonlinear (Part 2/2).**

Table 3. **Benchmarks on systems with 50 variables.** Ablation studies on simulated non-linear ground truth models with 50 nodes and different DAG structures. We report means and 95% confidence intervals (CIs) across 20 different ground truth models.

Model	#Edges	↓ A-AID	↓ P-AID	↓ OSET-AID
ARCO-GP	91 ± 1	0.03 ± 0.01	0.06 ± 0.02	0.04 ± 0.02
BAYESDAG	237 ± 5	0.25 ± 0.02	0.57 ± 0.05	0.25 ± 0.02
DAG-GNN	37 ± 4	0.14 ± 0.01	0.94 ± 0.02	0.14 ± 0.01
DDS	1082 ± 26	0.39 ± 0.01	0.42 ± 0.01	0.39 ± 0.01
DIBS-GP	70 ± 7	0.14 ± 0.01	0.82 ± 0.04	0.14 ± 0.01
GADGET	176 ± 5	0.31 ± 0.02	0.75 ± 0.02	0.31 ± 0.02
GES	220 ± 11	0.38 ± 0.05	0.71 ± 0.05	0.39 ± 0.05
GOLEM	56 ± 5	0.16 ± 0.02	0.88 ± 0.03	0.17 ± 0.02
GRASP	154 ± 8	0.30 ± 0.02	0.71 ± 0.03	0.31 ± 0.02
PC	108 ± 3	–	–	–

(a) **Scale-free Nonlinear (Part 1/2).**

Model	↓ ESHD	↑ AUROC	↑ AUPRC	↑ TPR	↑ TNR
ARCO-GP	6 ± 2	0.98 ± 0.01	0.95 ± 0.03	0.94 ± 0.02	1.00 ± 0.00
BAYESDAG	220 ± 11	0.77 ± 0.03	0.40 ± 0.03	0.59 ± 0.04	0.92 ± 0.00
DAG-GNN	90 ± 4	0.61 ± 0.02	0.42 ± 0.03	0.22 ± 0.03	0.99 ± 0.00
DDS	1037 ± 25	0.79 ± 0.03	0.52 ± 0.05	0.73 ± 0.03	0.57 ± 0.01
DIBS-GP	123 ± 7	0.61 ± 0.02	0.27 ± 0.03	0.22 ± 0.03	0.98 ± 0.00
GADGET	191 ± 7	0.79 ± 0.02	0.30 ± 0.03	0.42 ± 0.02	0.94 ± 0.00
GES	229 ± 13	0.69 ± 0.02	0.34 ± 0.03	0.46 ± 0.05	0.92 ± 0.00
GOLEM	94 ± 6	0.65 ± 0.02	0.43 ± 0.04	0.30 ± 0.03	0.99 ± 0.00
GRASP	160 ± 9	0.72 ± 0.01	0.41 ± 0.02	0.49 ± 0.02	0.95 ± 0.00
PC	156 ± 5	0.61 ± 0.01	0.25 ± 0.02	0.25 ± 0.02	0.96 ± 0.00

(b) **Scale-free Nonlinear (Part 2/2).**

Model	#Edges	↓ A-AID	↓ P-AID	↓ OSET-AID
ARCO-GP	48 ± 3	0.09 ± 0.01	0.26 ± 0.02	0.09 ± 0.01
BAYESDAG	195 ± 4	0.17 ± 0.03	0.48 ± 0.06	0.18 ± 0.03
DAG-GNN	34 ± 3	0.14 ± 0.01	0.57 ± 0.04	0.14 ± 0.01
DDS	928 ± 30	0.37 ± 0.03	0.45 ± 0.03	0.38 ± 0.03
DIBS-GP	33 ± 8	0.12 ± 0.01	0.43 ± 0.04	0.13 ± 0.01
GADGET	121 ± 4	0.25 ± 0.02	0.50 ± 0.05	0.26 ± 0.02
GES	134 ± 7	0.25 ± 0.03	0.45 ± 0.04	0.27 ± 0.03
GOLEM	52 ± 5	0.16 ± 0.02	0.54 ± 0.03	0.16 ± 0.01
GRASP	92 ± 5	0.18 ± 0.02	0.40 ± 0.05	0.19 ± 0.02
PC	92 ± 3	–	–	–

(c) **Erdös-Rényi Nonlinear (Part 1/2).**

Model	↓ ESHD	↑ AUROC	↑ AUPRC	↑ TPR	↑ TNR
ARCO-GP	56 ± 4	0.77 ± 0.02	0.57 ± 0.04	0.46 ± 0.03	1.00 ± 0.00
BAYESDAG	213 ± 9	0.68 ± 0.02	0.29 ± 0.02	0.41 ± 0.04	0.93 ± 0.00
DAG-GNN	110 ± 4	0.55 ± 0.01	0.25 ± 0.03	0.12 ± 0.01	0.99 ± 0.00
DDS	900 ± 26	0.75 ± 0.03	0.44 ± 0.05	0.64 ± 0.04	0.63 ± 0.01
DIBS-GP	105 ± 5	0.56 ± 0.02	0.30 ± 0.04	0.13 ± 0.03	0.99 ± 0.00
GADGET	130 ± 4	0.84 ± 0.01	0.46 ± 0.03	0.45 ± 0.02	0.97 ± 0.00
GES	122 ± 9	0.77 ± 0.02	0.51 ± 0.03	0.58 ± 0.03	0.97 ± 0.00
GOLEM	111 ± 4	0.59 ± 0.01	0.31 ± 0.03	0.20 ± 0.02	0.99 ± 0.00
GRASP	80 ± 6	0.79 ± 0.01	0.61 ± 0.03	0.60 ± 0.03	0.98 ± 0.00
PC	112 ± 5	0.69 ± 0.01	0.43 ± 0.02	0.40 ± 0.02	0.98 ± 0.00

(d) **Erdös-Rényi Nonlinear (Part 2/2).**

Table 4. **Nonlinear Average Causal Effect Estimation (ARCO-GP).** We report the mean absolute errors and 95% confidence intervals (CIs) between predicted and true average causal effects $\mathbb{E}_{X_j \mid do(X_i=1)}[X_j]$. The interventions $do(X_i = 1)$ are listed per column and the targets X_j listed per row. The results stem from simulations on 10 different ground truth models with fixed graph (see Figure 4) and simulated nonlinear mechanisms.

	do(X0 = 1)	do(X1 = 1)	do(X2 = 1)	do(X3 = 1)	do(X4 = 1)	do(X5 = 1)
X0	0.00 ± 0.00	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02
X1	0.07 ± 0.02	0.00 ± 0.00	0.09 ± 0.02	0.08 ± 0.03	0.09 ± 0.03	0.09 ± 0.03
X2	0.18 ± 0.09	0.25 ± 0.08	0.00 ± 0.00	0.10 ± 0.04	0.12 ± 0.03	0.10 ± 0.04
X3	0.11 ± 0.03	0.44 ± 0.23	0.06 ± 0.02	0.00 ± 0.00	0.06 ± 0.02	0.06 ± 0.02
X4	0.13 ± 0.05	0.30 ± 0.12	0.08 ± 0.03	0.08 ± 0.02	0.00 ± 0.00	0.06 ± 0.02
X5	0.26 ± 0.06	0.44 ± 0.21	0.04 ± 0.01	0.04 ± 0.01	0.34 ± 0.11	0.00 ± 0.00
X6	0.22 ± 0.08	0.27 ± 0.12	0.06 ± 0.03	0.06 ± 0.03	0.14 ± 0.04	0.10 ± 0.03
X7	0.12 ± 0.04	0.25 ± 0.14	0.09 ± 0.03	0.09 ± 0.03	0.08 ± 0.02	0.09 ± 0.04
X8	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02
X9	0.07 ± 0.03	0.06 ± 0.03	0.07 ± 0.03	0.07 ± 0.03	0.07 ± 0.03	0.07 ± 0.03
X10	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.03	0.06 ± 0.03	0.07 ± 0.02	0.07 ± 0.02
AVG	0.12 ± 0.02	0.20 ± 0.03	0.06 ± 0.01	0.06 ± 0.01	0.10 ± 0.01	0.07 ± 0.01

	do(X6 = 1)	do(X7 = 1)	do(X8 = 1)	do(X9 = 1)	do(X10 = 1)
X0	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.02
X1	0.12 ± 0.03	0.10 ± 0.02	0.10 ± 0.03	0.09 ± 0.03	0.09 ± 0.03
X2	0.12 ± 0.03	0.11 ± 0.03	0.11 ± 0.03	0.11 ± 0.04	0.11 ± 0.04
X3	0.07 ± 0.02	0.07 ± 0.02	0.07 ± 0.02	0.07 ± 0.02	0.07 ± 0.02
X4	0.08 ± 0.03	0.08 ± 0.02	0.08 ± 0.02	0.08 ± 0.02	0.08 ± 0.02
X5	0.04 ± 0.01	0.04 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.04 ± 0.01
X6	0.00 ± 0.00	0.06 ± 0.03	0.07 ± 0.03	0.06 ± 0.03	0.06 ± 0.03
X7	0.15 ± 0.05	0.00 ± 0.00	0.09 ± 0.03	0.09 ± 0.03	0.09 ± 0.03
X8	0.06 ± 0.02	0.06 ± 0.02	0.00 ± 0.00	0.06 ± 0.02	0.06 ± 0.02
X9	0.07 ± 0.03	0.06 ± 0.03	0.26 ± 0.17	0.00 ± 0.00	0.07 ± 0.03
X10	0.06 ± 0.02	0.06 ± 0.03	0.15 ± 0.08	0.12 ± 0.07	0.00 ± 0.00
AVG	0.07 ± 0.01	0.06 ± 0.01	0.09 ± 0.02	0.07 ± 0.01	0.07 ± 0.01

Table 5. **Nonlinear Average Causal Effect Estimation (DIBS).** We report the mean absolute errors and 95% confidence intervals (CIs) between predicted and true average causal effects $\mathbb{E}_{X_j \mid do(X_i=1)}[X_j]$. The interventions $do(X_i = 1)$ are listed per column and the targets X_j listed per row. The results stem from simulations on 10 different ground truth models with fixed graph (see Figure 4) and simulated nonlinear mechanisms.

	do(X0 = 1)	do(X1 = 1)	do(X2 = 1)	do(X3 = 1)	do(X4 = 1)	do(X5 = 1)
X0	0.00 ± 0.00	0.30 ± 0.28	0.23 ± 0.27	0.20 ± 0.22	0.10 ± 0.10	0.06 ± 0.03
X1	0.40 ± 0.33	0.00 ± 0.00	0.19 ± 0.28	0.09 ± 0.05	0.12 ± 0.10	0.07 ± 0.04
X2	0.45 ± 0.22	0.34 ± 0.20	0.00 ± 0.00	0.12 ± 0.09	0.09 ± 0.05	0.09 ± 0.06
X3	0.23 ± 0.23	0.55 ± 0.38	0.17 ± 0.25	0.00 ± 0.00	0.07 ± 0.03	0.10 ± 0.06
X4	0.24 ± 0.11	0.49 ± 0.23	0.20 ± 0.19	0.10 ± 0.06	0.00 ± 0.00	0.14 ± 0.07
X5	0.38 ± 0.10	0.50 ± 0.38	0.14 ± 0.19	0.07 ± 0.04	0.32 ± 0.20	0.00 ± 0.00
X6	0.37 ± 0.21	0.59 ± 0.26	0.06 ± 0.04	0.05 ± 0.03	0.20 ± 0.13	0.42 ± 0.23
X7	0.27 ± 0.15	0.43 ± 0.22	0.11 ± 0.07	0.10 ± 0.06	0.12 ± 0.09	0.17 ± 0.11
X8	0.07 ± 0.04	0.07 ± 0.04	0.07 ± 0.05	0.07 ± 0.04	0.07 ± 0.04	0.08 ± 0.05
X9	0.07 ± 0.04	0.07 ± 0.05	0.08 ± 0.04	0.07 ± 0.04	0.08 ± 0.04	0.07 ± 0.05
X10	0.04 ± 0.04	0.05 ± 0.03	0.07 ± 0.05	0.06 ± 0.04	0.04 ± 0.03	0.05 ± 0.04
AVG	0.23 ± 0.08	0.31 ± 0.10	0.12 ± 0.10	0.08 ± 0.04	0.11 ± 0.02	0.11 ± 0.03

	do(X6 = 1)	do(X7 = 1)	do(X8 = 1)	do(X9 = 1)	do(X10 = 1)
X0	0.06 ± 0.03	0.07 ± 0.03	0.08 ± 0.04	0.07 ± 0.02	0.05 ± 0.03
X1	0.10 ± 0.05	0.08 ± 0.05	0.08 ± 0.05	0.07 ± 0.04	0.09 ± 0.04
X2	0.10 ± 0.05	0.10 ± 0.05	0.10 ± 0.06	0.10 ± 0.05	0.09 ± 0.05
X3	0.08 ± 0.03	0.08 ± 0.03	0.09 ± 0.04	0.08 ± 0.03	0.08 ± 0.03
X4	0.11 ± 0.08	0.09 ± 0.04	0.08 ± 0.04	0.07 ± 0.04	0.08 ± 0.05
X5	0.20 ± 0.16	0.06 ± 0.04	0.07 ± 0.03	0.07 ± 0.03	0.05 ± 0.03
X6	0.00 ± 0.00	0.10 ± 0.11	0.06 ± 0.04	0.04 ± 0.03	0.04 ± 0.02
X7	0.27 ± 0.13	0.00 ± 0.00	0.10 ± 0.06	0.09 ± 0.05	0.10 ± 0.06
X8	0.07 ± 0.04	0.07 ± 0.04	0.00 ± 0.00	0.15 ± 0.21	0.14 ± 0.19
X9	0.08 ± 0.04	0.07 ± 0.05	0.40 ± 0.27	0.00 ± 0.00	0.14 ± 0.12
X10	0.06 ± 0.04	0.07 ± 0.05	0.29 ± 0.16	0.44 ± 0.34	0.00 ± 0.00
AVG	0.10 ± 0.02	0.07 ± 0.02	0.12 ± 0.04	0.11 ± 0.04	0.08 ± 0.02

Table 6. **Nonlinear Average Causal Effect Estimation (GADGET + BEEPS).** We report the mean absolute errors and 95% confidence intervals (CIs) between predicted and true average causal effects $\mathbb{E}_{X_j \mid do(X_i=1)}[X_j]$. The interventions $do(X_i = 1)$ are listed per column and the targets X_j listed per row. The results stem from simulations on 10 different ground truth models with fixed graph (see Figure 4) and simulated nonlinear mechanisms.

	do(X0 = 1)	do(X1 = 1)	do(X2 = 1)	do(X3 = 1)	do(X4 = 1)	do(X5 = 1)
X0	0.00 ± 0.00	0.38 ± 0.17	0.27 ± 0.13	0.34 ± 0.17	0.24 ± 0.14	0.15 ± 0.12
X1	0.74 ± 0.30	0.00 ± 0.00	0.20 ± 0.13	0.15 ± 0.08	0.22 ± 0.13	0.26 ± 0.16
X2	0.49 ± 0.28	0.31 ± 0.20	0.00 ± 0.00	0.16 ± 0.11	0.14 ± 0.14	0.21 ± 0.10
X3	0.49 ± 0.23	0.87 ± 0.44	0.09 ± 0.03	0.00 ± 0.00	0.13 ± 0.08	0.15 ± 0.10
X4	0.43 ± 0.28	0.74 ± 0.57	0.14 ± 0.09	0.15 ± 0.06	0.00 ± 0.00	0.20 ± 0.12
X5	0.54 ± 0.27	0.47 ± 0.31	0.07 ± 0.04	0.10 ± 0.07	0.32 ± 0.20	0.00 ± 0.00
X6	0.38 ± 0.17	0.48 ± 0.12	0.13 ± 0.10	0.09 ± 0.04	0.23 ± 0.13	0.33 ± 0.16
X7	0.30 ± 0.23	0.43 ± 0.24	0.09 ± 0.06	0.13 ± 0.11	0.12 ± 0.09	0.17 ± 0.13
X8	0.06 ± 0.04	0.06 ± 0.03	0.06 ± 0.04	0.06 ± 0.04	0.06 ± 0.04	0.06 ± 0.04
X9	0.08 ± 0.05	0.07 ± 0.05	0.08 ± 0.05	0.07 ± 0.05	0.08 ± 0.05	0.08 ± 0.05
X10	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
AVG	0.32 ± 0.10	0.35 ± 0.11	0.11 ± 0.03	0.12 ± 0.02	0.14 ± 0.03	0.15 ± 0.05

	do(X6 = 1)	do(X7 = 1)	do(X8 = 1)	do(X9 = 1)	do(X10 = 1)
X0	0.08 ± 0.05	0.09 ± 0.05	0.06 ± 0.03	0.06 ± 0.03	0.06 ± 0.02
X1	0.18 ± 0.11	0.15 ± 0.07	0.10 ± 0.04	0.10 ± 0.04	0.10 ± 0.04
X2	0.09 ± 0.04	0.12 ± 0.06	0.09 ± 0.05	0.09 ± 0.05	0.08 ± 0.04
X3	0.10 ± 0.05	0.12 ± 0.06	0.08 ± 0.02	0.08 ± 0.03	0.08 ± 0.03
X4	0.12 ± 0.07	0.13 ± 0.05	0.09 ± 0.03	0.08 ± 0.03	0.08 ± 0.04
X5	0.17 ± 0.11	0.12 ± 0.07	0.04 ± 0.02	0.04 ± 0.02	0.04 ± 0.02
X6	0.00 ± 0.00	0.24 ± 0.15	0.05 ± 0.03	0.05 ± 0.03	0.05 ± 0.03
X7	0.32 ± 0.09	0.00 ± 0.00	0.08 ± 0.04	0.08 ± 0.04	0.08 ± 0.04
X8	0.06 ± 0.03	0.06 ± 0.03	0.00 ± 0.00	0.15 ± 0.08	0.15 ± 0.08
X9	0.07 ± 0.05	0.08 ± 0.05	0.51 ± 0.20	0.00 ± 0.00	0.26 ± 0.13
X10	0.03 ± 0.01	0.03 ± 0.01	0.32 ± 0.16	0.49 ± 0.29	0.00 ± 0.00
AVG	0.11 ± 0.03	0.10 ± 0.03	0.13 ± 0.03	0.11 ± 0.02	0.09 ± 0.01