Attention-Based Neural Causal Models Under Markov Equivalence

Adiba Ejaz* **Yushu Pan*** **Hongshuo Yang*** **Elias Bareinboim**
Causal Artificial Intelligence Lab
Columbia University
{adiba.ejaz,yushupan,yhs,eb}@cs.columbia.edu

Abstract

Neural causal models can simulate interventions in complex, high-dimensional settings, but typically require a known causal graph. Observational data, however, generally identifies only a Markov equivalence class of graphs, represented by a CPDAG, and causal queries may vary across DAGs in that class. We introduce *Masked Neural Causal Models* (NCMs), a provably expressive nonparametric framework for simulating and bounding interventional queries over all models compatible with a given CPDAG given observational data. We give an optimization objective in the space of masked NCMs that asymptotically recovers the true bounds of the causal effect. To enable this optimization in practice, we introduce an attention-based architecture and a novel optimization strategy that recovers highly accurate bounds in discrete nonparametric settings.

1 Introduction

Causal generative models aim to simulate interventions in the world from high-dimensional observational data [33, 34, 18, 11, 7]. Their applications include editing images [17], modeling cellular perturbations [10, 31], estimating wage disparities from text [28], and more. However, there is no free lunch for such models: observations underdetermine the effects of interventions, a corollary of the *Causal Hierarchy Theorem* [4]. To circumvent this barrier, they rely on the additional inductive bias of a *causal graph*. In this work, we ask: to what extent can interventions be simulated from observational data when the causal graph is only partially identified?

This question arises because, like causal effects, the true causal graph cannot be learned from observational data alone. Still, this graph leaves traces in the data through invariances such as conditional independencies. A profusion of causal discovery methods use these independencies to learn the *Markov equivalence class* (MEC) of causal graphs that describe the data [26]. A completed partially directed acyclic graph (CPDAG) represents such an MEC: directed edges are shared by all graphs in the class, while undirected edges indicate causal relationships whose direction is ambiguous from observational data. However, it remains relatively understudied how to perform generative modeling given an equivalence class.

Consider a drug discovery setting in which we wish to improve insulin sensitivity by modulating an insulin-signaling pathway. Suppose Y denotes a measured protein activity implicated in insulin resistance, but Y itself is difficult to target safely. Instead, we may be able to perturb other proteins X and Z in the same signaling network, and we have observational measurements of protein abundance. How do we know whether an *intervention* on X, Z, or both would be effective in changing Y? We are not merely interested in the observational probability $P(Y \mid X, Z)$: even if X and Z are predictive of Y, this association need not reflect what would happen if we actively perturbed them.

* Equal contribution.

Suppose causal discovery on these observational measurements identifies the CPDAG $X - Z - Y$. This tells us that there is a direct causal relationship between Z and Y but not between X and Y. This informs us that intervening on X and Z jointly is redundant; intervening on Z suffices, since it mediates any possible pathway from X to Y. However, both $X \leftarrow Z \rightarrow Y$ and $X \leftarrow Z \leftarrow A$ are compatible WITH the learned CPDAG, but imply different responses to the intervention $\mathrm{do}(X = x)$: only in the second graph can perturbing X change the activity of Y. Thus, a model that commits to a single DAG may produce an incorrect simulation for our interventional query. Beyond this example, MECs have been discovered in domains spanning biology, medicine, neuroscience, climate, and more [27, 8, 23, 25]. Different DAGs in these equivalence classes can give rise to different causal effects; for decision-making, it is useful to know the range.

In this paper, we develop an approach to causal generative modeling from Markov equivalence classes. Our contributions are threefold.

1. We define *Masked Neural Causal Models* (Def. 2), and show that they can express any structural causal model over a given set of variables (Thm. 1).
2. We formulate the causal query bounding from a CPDAG as a constrained optimization problem in the space of masked NCMs and show that its optima recover the minimum and maximum causal effects consistent with the CPDAG and data (Thm. 4).
3. To solve this optimization in practice, we introduce an attention-based approach that recovers causal bounds with high accuracy across various settings, including image data (Sec. 4, 5).

Due to limited space, proofs of all theoretical results are provided in Appendix B.

2 Background and Problem Statement

2.1 Related work

The IDA family. Existing approaches for estimating causal effects from CPDAGs, such as IDA (Intervention-calculus when the DAG is Absent) and its variants [14, 15, 16], compute possible effects by enumerating DAGs or possible parent sets and are primarily developed for linear-Gaussian data. Unlike generative models, they do not learn mechanisms from which one can sample observational, interventional, and counterfactual distributions. Additionally, IDA assumes single-variable interventions; joint IDA generalizes this to joint interventions, but requires enumerating potentially large connected components of the equivalence class, and is thus best suited for sparse settings.

Amortized causal inference. A more recent line of work based on *prior-fitted networks* bypasses the CPDAG altogether, instead aiming to estimate causal effects directly from observational data [2, 24]. These methods amortize causal effect inference using synthetic pretraining, and encode causal assumptions in the data-generating prior instead of in a causal graph. Causal-PFN generates synthetic data from a prior that guarantees that the causal effect is uniquely identified from the observational data [2]. On the other hand, Do-PFN generates synthetic data from SCMs with non-linear functions and additive noise, where, assuming no unobserved confounding, the true graph (and hence the causal effect) is uniquely identified from observational data [21].

2.2 Preliminaries

Notation. Capital letters (X) denote variables, $\mathrm{dom}(X)$ denotes their domains, small letters (x) denote values in their domains, and bold letters denote sets of variables $(\mathbf{X})$ and their values $(\mathbf{x})$. $P(\mathbf{X})$ denotes the probability distribution over a set of variables $\mathbf{X}$. We consistently use $P(\mathbf{x})$ to abbreviate probabilities $P(\mathbf{X} = \mathbf{x})$.

Structural causal models [20, 3]. A *structural causal model* (SCMs) is a formalism for data-generating processes. An SCM $\mathcal{M}$ is a four-tuple $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, F, P(\mathbf{U}) \rangle$ where $\mathbf{V}$ and $\mathbf{U}$ are sets of endogenous (observed) and exogenous (unobserved) variables respectively. F is a set of mechanisms: each $V_i \in \mathbf{V}$ takes the value $f_i(\mathbf{pa}_i, \mathbf{u}_i)$, a function of the values of its endogenous and exogenous parents, $\mathbf{Pa}_i \subseteq \mathbf{V}$ and $\mathbf{U}_i \subseteq \mathbf{U}$, respectively. $P(\mathbf{U})$ is a joint distribution over $\mathbf{U}$. We assume the variables in $\mathbf{U}$ are jointly independent, and that for any distinct $V_i, V_j \in \mathbf{V}$, their unobserved parents are disjoint. Such an SCM is said to be *Markovian*.

Every Markovian SCM induces a *causal DAG*, constructed as follows: (1) add a vertex for every $V \in \mathbf{V}$, and (2) add an edge $V_i \to V_j$ for every $V_i, V_j \in \mathbf{V}$ if $V_i \in \mathbf{Pa}_{V_j}$.

Neural causal models [33, 34]. A *neural causal model* (NCM) is a structural causal model with the following additonal properties: (i) each exogenous variable $U \in \mathbf{U}$ has domain $[0, 1]$ and distribution $U \sim \text{Unif}([0, 1])$, and (ii) each function $f_i(\mathbf{pa}_i, \mathbf{u}_i)$ is a feed-forward neural network mapping from $\text{dom}(\mathbf{Pa_i}) \times \text{dom}(\mathbf{U_i}) \to \text{dom}(V_i)$. Given a DAG $\mathcal{G}$, an NCM is said to be $\mathcal{G}$-constrained if for all $V_i \in \mathbf{V}$, the function $f_i(\mathbf{pa}_i, \mathbf{u}_i)$ respects $\mathbf{Pa}_i = \mathbf{Pa}_i^{\mathcal{G}}$, where $\mathbf{Pa}_i^{\mathcal{G}}$ contains parents of V_i in $\mathcal{G}$.

Markov equivalence classes [19, 26]. Two causal DAGs $\mathcal{G}, \mathcal{H}$ are said to be Markov equivalent if they encode exactly the same d-separations. The Markov equivalence class (MEC) of a DAG is the set of all graphs that are Markov equivalent to it. An MEC $\mathcal{M}$ is represented by a unique completed partially directed graph (CPDAG). We frequently refer to an MEC by its representative CPDAG. A CPDAG $\mathcal{E}$ for $\mathcal{M}$ has an undirected edge $X - Y$ if $\mathcal{M}$ contains two DAGs $\mathcal{G}_1, \mathcal{G}_2$ with $X \to Y$ in $\mathcal{G}_1$ and $Y \to X$ in $\mathcal{G}_2$. $\mathcal{E}$ has a directed edge $X \to Y$ if $X \to Y$ is in every DAG in $\mathcal{M}$. The adjacencies of a variable X in a CPDAG $\mathcal{E}$ comprise those variables connected by any edge to X. An *unshielded non-collider* in $\mathcal{E}$ is a structure $X - Z - Y$ where X, Y are non-adjacent. An *unshielded collider* in $\mathcal{E}$ is a structure $X \to Z \leftarrow Y$ where X, Y are non-adjacent. A DAG $\mathcal{G}$ belongs to the MEC represented by $\mathcal{E}$ if and only if it has the exact same adjacencies and unshielded colliders [26].

Now, we define the problem of partial causal identification from a CPDAG. We assume we are given observational data sampled from a true, unknown SCM $\mathcal{M}^*$. $\mathcal{M}^*$ induces a causal graph $\mathcal{G}$, to which $P(\mathbf{v})$ is faithful. We are also given the CPDAG $\mathcal{E}$ representing the MEC of $\mathcal{G}$.

Definition 1 (Optimal Interventional Bound (CPDAG) [36, Def. 2.1])**.** *For a CPDAG $\mathcal{E}$ and observational distribution $P(\mathbf{V})$, the optimal bound $[l, r]$ over an interventional probability $P(\mathbf{y_x})$ is defined as, respectively, the minimum and maximum of the following optimization problem:*

$$\min / \max_{\mathcal{M} \in \Omega(\mathcal{E})} P^{\mathcal{M}}(\mathbf{y_x})$$
$$\textit{s.t. } P^{\mathcal{M}}(\mathbf{V}) = P(\mathbf{V})$$

where $\Omega(\mathcal{E})$ is the set of all SCMs whose induced graph $\mathcal{G}$ is in the MEC represented by $\mathcal{E}$.

3 A Neural Parameterization of Equivalence Classes

3.1 Masked Neural Causal Model

In this section, we introduce the construct of a *masked* neural causal model. In a standard NCM, a neural network f_{θ_i} determines the value of a variable V_i. The set of inputs of this network are fixed: they are the causal parents of V_i. Masked NCMs relax this assumption, and enable learning not only the parameters of θ_i but also the causal parents of V_i that optimize our desired objective.

Definition 2 (Masked Neural Causal Model ($\mathcal{E}$-NCM))**.** *Let $\mathbf{V}$ be a set of n observed variables. A* masked neural causal model *is a 4-tuple $\mathcal{N} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ where*

- $\mathbf{V}$ *is a set of (discrete and finite) endogenous variables,*
- $\mathbf{U}$ *is a set of exogenous variables following the distribution $P(\mathbf{U})$, given by $U_i \sim \text{Unif}([0, 1])$ for each $V_i \in \mathbf{V}$,*
- $\mathbf{m} \in [0, 1]^{n \times n}$ *is a mask, and*
- $\mathcal{F}_{\theta,\mathbf{m}}$ *is a collection of functions $f_i : dom(\mathbf{V}_{-i}) \times dom(U_i) \to dom(V_i)$ for each V_i, jointly represented by a neural network architecture with parameters θ, and satisfying the following mask invariance. Let $\mathbf{Pa}_i \subseteq \mathbf{V}_{-i}$ comprise variables V_j such that $\mathbf{m}_{ji} > 0$. For any values $\mathbf{v}_{-i}, \mathbf{v}'_{-i} \in dom(\mathbf{V}_{-1})$ that agree on $\mathbf{Pa}_i$, and any $u_i \in dom(U_i)$, we have*

$$f_{\theta_i}(\mathbf{v}_{-i}, u_i) = f_{\theta_i}(\mathbf{v}'_{-i}, u_i).$$

We make two comments on the above definition. First, the functions may be parameterized by a separate feed-forward network for each variable, or a shared Transformer, as we propose in Sec. 4.

Second, the functions of a standard SCM give a recursive (acyclic) mapping from its exogenous variables $\mathbf{U}$ to its endogenous variables $\mathbf{V}$. However, this is not the case in a masked NCM, which has cyclic dependencies between its variables, requiring a new notion of valuation.l

Definition 3 (Synchronous masked NCM valuation)**.** *Consider a masked NCM* $\mathcal{N} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ *and a nonnegative integer* d*. The* synchronous depth-d valuation $\mathbf{V}^{(d)}(\mathbf{u})$ *for any exogenous assignment* $\mathbf{u} \in dom(\mathbf{U})$ *is defined as the unique value obtained by evaluating, in order* $t = 0, \ldots, d$*,*

$$\mathbf{V}^{(0)}(\mathbf{u}) = \mathbf{0}$$
$$\mathbf{V}^{(t)}(\mathbf{u}) = (f_n(\mathbf{V}_{-1}^{(t-1)}, \mathbf{u}_1), \ldots, f_1(\mathbf{V}_{-n}^{(t-1)}, \mathbf{u}_n))$$

The interventional valuation, for a set of variables $\mathbf{X}$ *and assignment* $\mathbf{x} \in dom(\mathbf{X})$*, is defined by substituting* f_i *with the constant function* $\mathbf{x}_i$ *for any variable* $X_i \in \mathbf{X}$*.*

Given this valuation, we can define the distributions induced by a masked NCM.

Definition 4 (Masked NCM induced distributions)**.** *Consider a masked NCM* $\mathcal{N} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ *and a nonnegative integer* d*. Let* $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ *be sets of observed variables, and* $\mathbf{x} \in dom(\mathbf{X}), \mathbf{y} \in dom(\mathbf{Y})$ *be assignments to their values. The induced distribution of depth* d *is defined as*

$$P(\mathbf{y}_{\mathbf{x}}^{(d)}) = \sum_{\mathbf{u} \in dom(\mathbf{U})} \mathbf{1}[\mathbf{Y}_{\mathbf{x}}^{(d)}(\mathbf{u}) = \mathbf{y}] P(\mathbf{u})$$

A mask assignment $\mathbf{m}$ naturally induces a directed graph over the endogenous variables, which contains an edge $i \to j$ whenever $\mathbf{m}_{ij} > 0$. We say a mask *acyclic* if the graph that it induces is acyclic. When a masked NCM is parameterized by an *acyclic* mask, we can show that the familiar computation according to a topological order recovers the masked NCM valuation (Def. 6, Prop. 2).

In the next result, we establish that masked NCMs with acyclic mask assignments can simulate the observational and interventional distributions of any standard SCM, and vice-versa.

Theorem 1. *[Equivalence between masked NCMs and standard SCMs] Consider a masked NCM* $\mathcal{N}$ *over variables with an acyclic mask inducing a DAG* $\mathcal{G}$*. Then, there exists an SCM* $\mathcal{M}$ *over* $\mathbf{V}$ *such that, for any sets of variables* $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ *and their values* $\mathbf{x}, \mathbf{y}$*,*

$$P^{\mathcal{N}}(\mathbf{y}_{\mathbf{x}}^{(d)}) = P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}})$$

where d *is the length of the longest directed path in* $\mathcal{G}$*. The converse also holds: for any SCM* $\mathcal{M}$ *over* $\mathbf{V}$ *inducing a causal graph* $\mathcal{G}$*, there exists a masked NCM* $\mathcal{N}$ *over* $\mathcal{G}$ *whose mask induces the graph* $\mathcal{G}$ *such that for any sets of variables* $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ *and their values* $\mathbf{x}, \mathbf{y}$*,*

$$P^{\mathcal{N}}(\mathbf{y}_{\mathbf{x}}^{(d)}) = P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}}).$$

This result shows that masked NCMs with acyclic masks constitute a sufficiently expressive space to describe observational and interventional distributions induced by any SCM over the given variables. Additionally, it shows that this space is not *too* expressive–it captures *exactly* the latter distributions.

3.2 Constraining Masked NCMs with a Markov Equivalence Class

Masked NCMs can express any SCM over a given set of variables and allow for gradient-based optimization over the neural network parameters and mask. Next, we consider how this space can be restricted using a given Markov equivalence class while preserving this property.

Recall that a given DAG $\mathcal{G}$ is in the MEC ($\mathcal{E}$ if and only if it has the same adjacencies and unshielded colliders as $\mathcal{E}$ (Sec. 2). To restrict the space of masks to acyclic masks, we can use the following well-known result.

Theorem 2. *[Acyclicity constraint ([37, adapted from Thm. 1])] A mask* $\mathbf{m} \in [0, 1]^{n \times n}$ *induces a DAG if and only if*

$$h(\mathbf{m}) = \operatorname{tr}(e^{\mathbf{m}}) - n = 0$$

where $e^{\mathbf{m}}$ *is the matrix exponential of* $\mathbf{m}$*. Furthermore,* $h(\mathbf{m})$ *is smooth.*

Next, we need to ensure the mask contains no unshielded colliders not already in the given CPDAG.

Proposition 1. *[Non-collider constraint] Given a CPDAG $\mathcal{E}$, let $\mathcal{T}_{\mathcal{E}}$ be the set of unshielded non-colliders. A matrix $\mathbf{m} \in [0,1]^{n \times n}$ represents a graph in $\mathcal{E}$ only if*

$$c(\mathbf{m}) = \sum_{(i,j,k) \in \mathcal{T}_{\mathcal{E}}} m_{ij} m_{kj} = 0.$$

Furthermore, $c(\mathbf{m})$ is smooth.

Let $\langle \Theta, \mathbf{M} \rangle$ be the space of all masked NCMs over variables $\mathbf{V}$. Given a CPDAG $\mathcal{E}$, we can define a subspace of $\Theta, \mathbf{M}$ consistent with the constraints encoded in $\mathcal{E}$.

Definition 5 ($\mathcal{E}$-Subspace of Masked NCMs)**.** *Let $\langle \Theta, \mathbf{M} \rangle$ be the space of all masked NCMs over variables $\mathbf{V}$. The $\mathcal{E}$-*subspace *of $\langle \Theta, \mathbf{M} \rangle$, denoted $\mathcal{E}(\langle \Theta, \mathbf{M} \rangle)$ is defined as the set of all $\theta \in \Theta, \mathbf{m} \in \mathbf{M}$ such that (i) $h(\mathbf{m}) = 0$, (ii) $c(\mathbf{m}) = 0$, (iii) $\mathbf{m}_{ij} = 1$ for any directed edge $V_i \to V_j$ in $\mathcal{E}$, and (iv) $\mathbf{m}_{ij} = \mathbf{m}_{ji} = 0$ for any non-adjacent variables V_i, V_j in $\mathcal{E}$.*

Each constraint in the definition of an $\mathcal{E}$-subspace of masked NCMs is a smooth equality constraint. Note that there may be mask values in this subspace that do not exactly induce a graph in $\mathcal{E}$, since we do not enforce the adjacencies in $\mathcal{E}$. As it turns out, this constraint is not needed in order to derive optimal interventional bounds from CPDAGs (Def. 1) via masked NCMs.

Theorem 3. *[Partial identification via $\mathcal{E}$-masked NCMs.] Consider a CPDAG $\mathcal{E}$ over n variables $\mathbf{V}$, an observational distribution $P(\mathbf{V})$, and an interventional query $P(\mathbf{y_x})$. Let $\mathcal{E}(\langle \Theta, \mathbf{M} \rangle)$ be the $\mathcal{E}$-subspace of the space of all masked NCMs over $\mathbf{V}$ The optimal interventional bounds (Def. 1 of $P(\mathbf{y_x})$ from $\mathcal{E}$ and $P(\mathbf{V})$ can be derived by solving the following optimization problem:*

$$\min / \max_{\theta, \mathbf{m} \in \mathcal{E}(\langle \Theta, \mathbf{M} \rangle)} P^{\mathcal{N}(\theta, \mathbf{m})}(\mathbf{y_x}) \text{ such that } P^{\mathcal{N}(\theta, \mathbf{m})}(\mathbf{V}^{(n)}) = P(\mathbf{V})$$

4 Architecture and Optimization

4.1 A Relaxed Training Objective

In the previous section, we characterized, theoretically, how optimization in the space of masked NCMs can recover interventional bounds from a given CPDAG and data. In this section, we propose a practical implementation to solve this challenging, non-convex optimization problem.

Fitting the observational distribution. First, while the optimization is formulated in terms of the population $P(\mathbf{v})$, in practice we have a finite dataset of i.i.d. samples $\mathcal{D} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$. Let $\hat{P}(\mathbf{v})$ be the empirical distribution and $\mathbb{D}(\cdot||\cdot)$ be some divergence function between empirical distributions, e.g., KL-divergence, negative log-likelihood, or max-mean discrepancy.

Optimizing the interventional query. Given a CPDAG $\mathcal{E}$ we learn masked NCM parameters θ and a mask $\mathbf{m}$ as follows. By design, we fix the mask parameters $m_{ij} = 1$ whenever $\mathcal{E}$ contains a directed edge $i \to j$. Then, extending the objective of standard neural causal models [33] to our case, to optimize a query $P(\mathbf{y_x})$), we minimize the loss

$$\mathcal{L}(\theta, \mathbf{m}) = \mathbb{D}(\hat{P}^{\mathcal{N}(\theta, \mathbf{m})}(\mathbf{V}^n)||\hat{P}(\mathbf{V})) + \lambda_h \cdot h(\mathbf{m}) + \lambda_c \cdot c(\mathbf{m}) \pm \lambda_q \cdot \mathbb{D}(\hat{P}^{\mathcal{N}(\theta, \mathbf{m})}(\mathbf{y}_{\mathbf{x}}^{(n)})||\hat{P}(\mathbf{y_x})))$$

where $\hat{P}(\mathbf{y_x})$ is a synthetic empirical distribution, e.g., a sample with all $\mathbf{y} = 1$ if we want to maximize $P(\mathbf{y} = 1 \mid do(\mathbf{x}))$. The interventional term is an artificial direction added to (respectively, subtracted from) the observational term in order to maximize (respectively, minimize) the query. The hyperparameter λ_q balances the interventional term during training; see App. D.4.2 for an ablation.

Example of masking. A basic feed-forward network that respects the non-parent invariance of Def. 2 can be obtained by taking an element-wise product. For simplicity, assume the endogenous variables do not contain $\mathbf{0}$ in their domain. In Def. 2, for a variable V_i, we have

$$V_i \leftarrow f_i(\mathbf{v}_{-i}, u_i) \tag{1}$$

To guarantee that f_i does not depend on masked values v_j in $\mathbf{v}_i$ where $m_{ji} = 0$, we can apply an explicit element-wise product, feeding $\mathbf{m}_{:i} \odot \mathbf{v}_{-i}$ into f_i instead of the vector $\mathbf{v}_i$.

We evaluate such an approach in practice, with implementation details in Sec. C.1 and results in Sec. 5. Next, we provide a more accurate attention-based optimization strategy.

4.2 Attention-based architecture for $\mathcal{E}$-NCM

We instantiate the structural functions of an $\mathcal{E}$-NCM by a masked-attention architecture. At a high level, the architecture takes the exogenous variables $\mathbf{U}$ and the endogenous variables $\mathbf{V}$ as token inputs and produces predictions for all endogenous $\mathbf{V}$ as token outputs. At each position i, the input is the exogenous draw U_i and the corresponding output is the endogenous prediction V_i; in this sense the attention block serves as the structural function $\hat{f}_i$ in (1), mapping U_i to V_i. The masked attention mechanism is what enforces the structural constraint: U_i is prevented from attending to any token V_j for which the indicator $m_{ji} = 0$, i.e. for which V_j is not a parent of V_i in the underlying DAG. We describe this construction step by step.

Tokens and embeddings. The input sequence is ordered as $(U_1, \ldots, U_n, V_1, \ldots, V_n)$ and is mapped into a sequence of $2n$ embedded tokens $\mathbf{T} \in \mathbb{R}^{2n \times D}$ via a pair of position-wise embedding maps. Throughout the description of the architecture we use $a, b \in \{1, \ldots, 2n\}$ to index *token positions* in the sequence, while $i, j \in \{1, \ldots, n\}$ continue to index *variables* as in the rest of the paper; concretely, the token at position a corresponds to U_a when $a \leq n$ and to V_{a-n} otherwise. The embedding map is then

$$\mathbf{T}_a \;=\; \begin{cases} \mathrm{Embed}_a^{(u)}(U_a), & 1 \leq a \leq n, \\ \mathrm{Embed}_{a-n}^{(v)}(V_{a-n}), & n+1 \leq a \leq 2n. \end{cases} \tag{2}$$

Because each U_i is a continuous uniform latent vector (Def. 2), $\mathrm{Embed}_a^{(u)} \colon \mathbb{R}^{D_U} \to \mathbb{R}^D$ is implemented as a multi-layer perceptron. For the endogenous V_i, the embedding is a learned lookup table $\mathrm{Embed}_a^{(v)} \colon \{0, 1\} \to \mathbb{R}^D$. Importantly, the embedding parameters are not shared across variables nor across the latent / endogenous sides: each of the $2n$ positions has its own embedding, which lets the network learn variable-specific token representations.

Masked attention. After embedding, the $2n$ tokens $\mathbf{T} \in \mathbb{R}^{2n \times D}$ are processed by a stack of L Transformer blocks. Each block first projects every token $\mathbf{t}_a$ into a query, key, and value via learned linear maps,

$$\mathbf{q}_a \;=\; W_Q\, \mathbf{t}_a, \qquad \mathbf{k}_a \;=\; W_K\, \mathbf{t}_a, \qquad \mathbf{v}_a \;=\; W_V\, \mathbf{t}_a, \qquad a = 1, \ldots, 2n, \tag{3}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{D \times D}$ are the projection matrices shared across positions. The $\mathcal{E}$-mask $\widetilde{\mathbf{m}} \in \{0, 1\}^{2n \times 2n}$ is injected directly into the attention scores before the softmax,

$$s_{ab} \;=\; \frac{\mathbf{q}_a^\top \mathbf{k}_b}{\sqrt{D}} \;+\; \log \widetilde{m}_{ba}, \qquad \alpha_{ab} \;=\; \frac{\exp(s_{ab})}{\sum_{b'=1}^{2n} \exp(s_{ab'})}, \tag{4}$$

where s_{ab} is the attention logit and α_{ab} is the post-softmax attention weight assigned by the query at position a to the key at position b. Notice that the augmented mask entries $\tilde{m}_{ba}$ will make α_{ab} as 0 (since $\log 0 = -\infty$) if $\tilde{m}_{ba} = 0$. Specifically, $\tilde{m}_{ba}$ is constructed as follows:

- **Self-attention everywhere.** $\widetilde{m}_{aa} = 1$ for every position $a \in \{1, \ldots, 2n\}$, so each token may always attend to itself. For latent positions $a \leq n$, this self-loop encodes the structural fact that U_i is, by construction, the exogenous parent of V_i (Def. 2); the final output V_i is produced from this token's hidden state. For value positions $a > n$ the self-loop has a different purpose: although the value tokens V_j are never decoded into outputs and do not appear in the training objective directly, allowing them to refine themselves through repeated self-attention lets the network build expressive representations of V_j that the latent tokens can subsequently attend to.
- **No cross-attention within U.** $\widetilde{m}_{ba} = 0$ for every pair of distinct latent positions $a \neq b$ with $a, b \leq n$. By Def. 2 the exogenous noises are mutually independent, so U_j is never a parent of V_i for $j \neq i$.
- **No cross-attention within V.** $\widetilde{m}_{ba} = 0$ for every pair of distinct value positions $a \neq b$ with $a, b > n$. As noted above, value tokens are not decoded as outputs; their representations contribute to the model output only through interacting with former token $\mathbf{U}$. Allowing them to mix with each other would not respect the structure imposed by the mask.

- **Cross U and V governed by the mask.** A latent token at position $a = i$ attends to a value token at position $b = n + j$ *iff* the $\mathcal{E}$-mask declares V_j to be a parent of V_i; that is, $\widetilde{m}_{n+j,\,i} = m_{ji}$, where m_{ji} is the $\mathcal{E}$ mask entry. When $m_{ji} = 0$ the score $s_{i,\,n+j}$ is driven to $-\infty$ and the corresponding weight $\alpha_{i,\,n+j}$ vanishes after the softmax; when $m_{ji} = 1$ the weight is a learned positive value. All remaining cross entries (the $\mathbf{U} \to \mathbf{V}$ direction) are zero, so the value tokens never attend back to the latents.

Concretely: when V_j is not a parent of V_i, the score $s_{i,\,n+j}$ is driven to $-\infty$ and the corresponding weight $\alpha_{i,\,n+j}$ vanishes after the softmax; when $V_j \in \mathrm{pa}(V_i)$, the weight is a learned positive value. The output of the attention block at the latent position $a = i$ is therefore the convex combination

$$\mathrm{Attn}(\mathbf{T})_i \;=\; \alpha_{i,i}\,\mathbf{v}_i \;+\; \sum_{j \,\in\, \mathrm{pa}(V_i)} \alpha_{i,\,n+j}\,\mathbf{v}_{n+j}, \tag{5}$$

mixing the latent's own value with the value tokens of its DAG parents only, which encodes the structure carried by the $\mathcal{E}$-mask continuously through the attention scores of Eq. 4. To illustrate this design, observe that the query $\mathbf{q}_i$ is a learned function of U_i alone (by Eq. 3 and the embedding map of Eq. 2); since every weight $\alpha_{i,b}$ depends on $\mathbf{q}_i$ through the dot product $\mathbf{q}_i^\top \mathbf{k}_b$, U_i enters the computation of every attention coefficient at position i, in particular through the self-loop term $\alpha_{i,i}\,\mathbf{v}_i$ that is guaranteed to be non-zero by $\widetilde{m}_{ii} = 1$. The mask therefore, restricts only which value tokens are mixed in alongside U_i, never whether U_i itself participates in producing the latent's output. Following the masked attention, each block applies a position-wise feed-forward sublayer with residual connections and layer normalization in the standard pre-norm Transformer arrangement; the architecture stacks L such blocks, and the final hidden state at each latent position $a = i$ is fed to the per-variable output head to output V_i[1].

4.3 Reinforcement learning over the space of masks

The differentiable-mask formulation searches over a continuous relaxation of adjacency matrices, and generates samples without hard thresholding of these masks. While this is desirable for gradient-based optimization, it may come at a cost of accuracy. We next present an alternative search strategy over masks that leverages thresholding in the attention-based architecture.

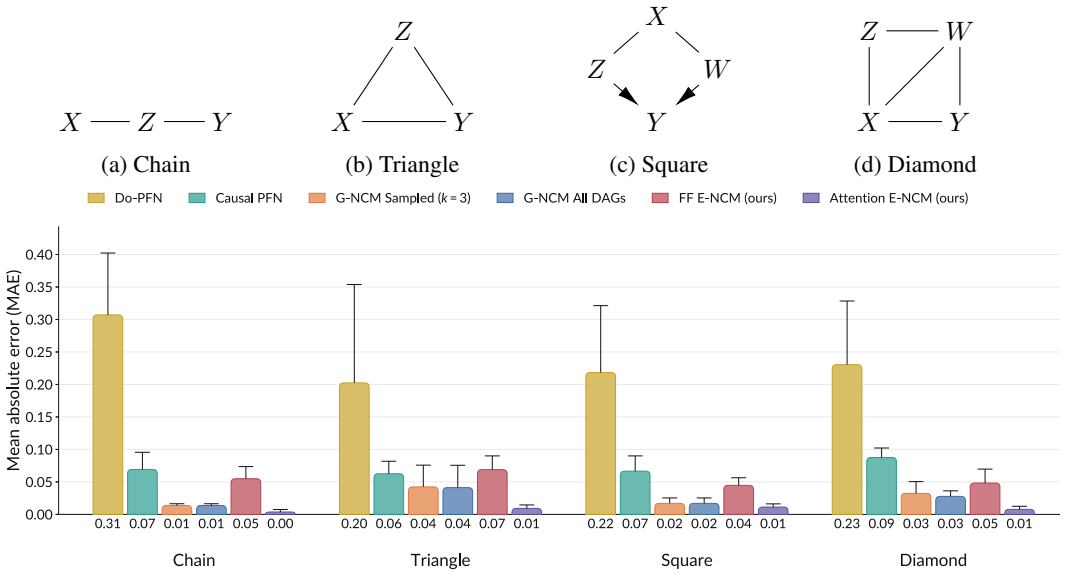
At a high-level, we indirectly sample from the space of masks by sampling from the space of topological orderings over n nodes. We parameterize a distribution over orders using Plackett–Luce scores [22, 13]: a higher score for a variable V_i makes it more likely to appear earlier in the sampled topological order. Given a sampled order π, we sample a binary mask m from π as follows. Each edge allowed by $\mathcal{E}$ that points forward in π is included with its own learnable probability, and all other entries of m are zero (App. C.2). We then evaluate the $\mathcal{E}$-NCM under that mask. Since all included edges point forward in π, acyclicity is guaranteed by construction; only the collider constraints remain to be enforced. We optimize the ordering distribution with REINFORCE [32], using a reward that combines the query value with data-fit and collider constraints, while the attention parameters are trained on the data likelihood alone. RL search over variable orderings has also been used for causal discovery [30, 35]; here we use it to optimize a query within the equivalence class of a given CPDAG. Details and pseudocode can be found in Sec. C.2.

5 Experiments

5.1 Partial identification accuracy

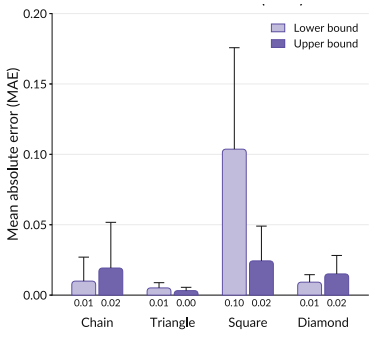
Data generation. We consider four distinct MECs, whose CPDAGs are shown in Fig. 1. We fix a true causal graph from each MEC, and consider $\{1, 8\}-$dimensional variables with binary domains. We sample five datasets with 10^4 samples for the 1D setting and 10^5 for 8D from an expressive regional canonical model following [33]. The true MEC $\mathcal{E}$ is given as input to all relevant methods. To compute the gold-standard bounds, we manually derive possible causal estimands for graphs in $\mathcal{E}$, and compute these using empirical frequencies in the generated data.

[1]The structure here is illustrated with one-layer attention with one head. But it can easily be extended into a multi-head.

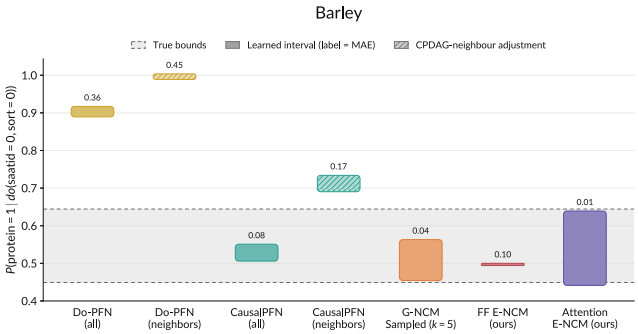


(a) Chain (b) Triangle (c) Square (d) Diamond

(e) Mean absolute error of various methods.



(f) Attention $\mathcal{E}$-NCM accuracy on 8-dimensional covariates.

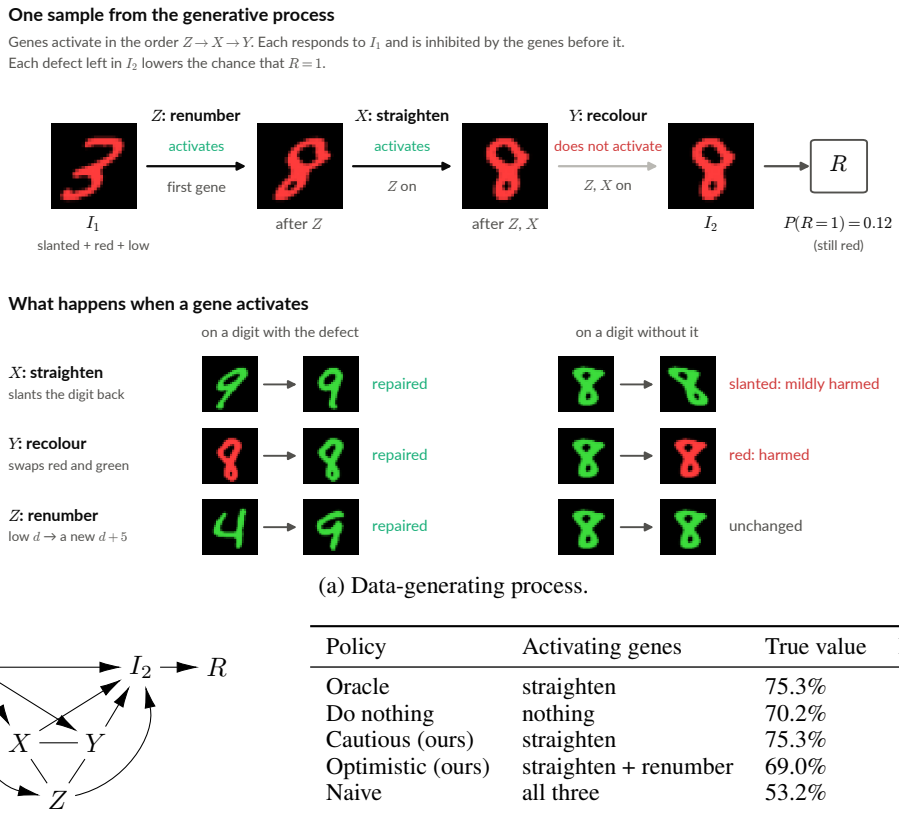


(g) Estimated bounds for the Barley network (48 nodes).

Figure 1: Graphs and results of Exps. 5.1, 5.2. Error bars show one standard deviation across 5 random datasets and 3 training seeds (except (g), which shows one trial). The Attention $\mathcal{E}$-NCM consistently outperforms baselines in accuracy, showing near-exact recovery of upper and lower bounds of interventional queries across graphs. See Fig. 3 for more details.

Methods and baselines. We evaluate six methods in total: (1) a $\mathcal{G}$-NCM baseline which enumerates all DAGs in the given MEC, and trains a $\mathcal{G}$-NCM for each; (2) a $\mathcal{G}$-NCM baseline which samples $k = 3$ DAGs from the MEC, following [9]; (3) Do-PFN [24], with 10^3 posterior samples to estimate the $[2.5\%, 97.5\%]$ confidence interval, used as a proxy for the lower/upper bound; (4) Causal-PFN [2], with similar posterior sampling; (5) a feed-forward $\mathcal{E}$-NCM using soft masks following Thm. 1, and (6) an Attention $\mathcal{E}$-NCM trained with reinforcement learning. We rerun each neural method thrice, varying the random initialization.

Results. Results for the 1D setting are shown in Fig. 1e, and 8D in Fig. 1f. The Attention $\mathcal{E}$-NCM consistently outperforms baselines in accuracy, showing near-exact recovery of upper and lower bounds of interventional queries across graphs. Do-PFN is consistently the least accurate, in two ways: it outputs very similar upper/lower confidence estimates despite non-identifiability, and these estimates lie significantly outside the true bounds (Fig. 3). This may be explained by a mismatch between our data-generating process (discrete, non-parametric) and Do-PFN's pre-training prior (continuous, additive noise). The sampling baseline is competitive in cases where sampling exhausts the entire MEC, but its performance is less consistent than that of the FF $\mathcal{E}$-NCM; the high variance highlights the sensitivity of this approach to graph sampling. Sec. D provides further details.



(a) Data-generating process.

(b) Input graph $\mathcal{E}$.

Policy	Activating genes	True value	Regret
Oracle	straighten	75.3%	—
Do nothing	nothing	70.2%	5.1
Cautious (ours)	straighten	75.3%	**0.0**
Optimistic (ours)	straighten + renumber	69.0%	6.3
Naive	all three	53.2%	22.1

(c) Decisions and regret.

Figure 2: Exp. 5.3. (a) One sample from the data-generating process (top), and the effect of each gene on a digit with and without its defect (bottom). (b) The input partially directed graph; the true causal graph uses the order $Z \prec X \prec Y$. (c) Intervention selected by each policy, the resulting % of cells labelled healthy, and its regret relative to the best intervention under the true DAG (oracle).

5.2 Real-world Barley network

Setting. The Barley directed graph [12] is a real-world Bayesian network from the Bayesian Network Repository containing 48 nodes and 84 edges.[2] It was originally designed as a decision-support model for growing malting barley without pesticides. We consider the query: what is the combined causal effect of the barley variety (sort) and sowing time (saatid) of barley on its protein content, $P(protein = 1 \mid do(sort = 0, saatid = 0))$? We consider using the Barley graph's MEC $\mathcal{E}$, which contains 96 DAGs, to bound this query. To ensure known ground-truth, we generate a binary dataset of 10^6 samples, and evaluate the methods presented in Exp. 5.1 (increasing to $k = 5$ for the $\mathcal{G}$-sampling baseline) where feasible to bound the query. For Do-PFN and Causal-PFN, we consider two possible adjustment sets: the set of *all* other variables and the set of neighbors of the treatment.

Results. The Attention $\mathcal{E}$-NCM recovers the ground truth bounds exactly, despite the high dimensionality. The FF $\mathcal{E}$-NCM, on the other hand, mistakenly learns the same value for the upper and lower bounds, though this value is within the true bounds. The sampling baseline correctly learns the lower bound, but its upper bound is off by ≈ 0.1, suggesting it is unable to find the extrema of the query for this large MEC. Do-PFN's estimates lie outside the true bounds for both adjustment sets.

5.3 Decision-making from images

We now consider a setting motivated by decision-making in biology. Say we have a microscopy image I_1 of a possibly diseased cell, and measurements of the expression of three genes X, Y, and Z that

[2] https://www.bnlearn.com/bnrepository/

respond to the cell state and regulate one another, in directions that are unknown. The active genes alter the cell, which is imaged again as I_2 and finally scored as 'healthy' or 'unhealthy', $R \in \{0, 1\}$. The experimenter does not know which features of the cell the genes respond to, nor what the genes change. Given observational samples of (I_1, X, Y, Z, I_2, R), the goal is to decide which genes to activate, for instance via CRISPR, to maximize the fraction of healthy cells. We emulate this task using MNIST digits in place of cells, which gives us access to the ground truth.

Data-generating process. Fig. 2 depicts the causal mechanisms and causal graph. Each initial 'cell' I_1 is a colored MNIST digit that may have up to three defects: it may be slanted, colored red rather than green, or show a low number (0 to 4) rather than a high one (5 to 9). Each gene X, Y, and Z tends to activate when its own defect is present in I_1, and the active genes edit I_1 to produce I_2. The genes can also activate on healthy cells and regulate one another. Fig. 2a elaborates the causal mechanisms. Importantly, genes X and Y can harm healthy digits while Z never does.

Policies and results. We consider interventions on various subsets of genes $\mathbf{S} \subseteq \{X, Y, Z\}$ (possibly empty) with the goal of maximizing $P(R = 1 \mid do(\mathbf{S} = 1))$. Since $P(R = 1 \mid do(\mathbf{S} = 1))$ is not point identifiable, we compare three policies for choosing $\mathbf{S}$. The *cautious* policy picks the gene set with the largest lower bound, and the *optimistic* policy the one with the largest upper bound. The naive policy picks the gene set with the largest observed rate $P(R = 1 \mid S = 1)$, i.e. the genes whose activation co-occurs most often with healthy cells.

Results are shown in Table 2c. The cautious policy activates X (straighten) and matches the oracle. The naive policy activates all three genes: among cells in which all three happened to be active, 95% end up healthy, since genes mostly activate on digits that need them. Forcing them on in every cell, however, also harms the digits without defects, and only 53.2% of cells end up healthy. Sec. D.3 reports all learned bounds, and a robustness study over randomly drawn mechanisms.

5.4 Ablation studies

We report ablation studies on misspecified CPDAGs and the hyperparameters of the training objective. Results for misspecified CPDAGs are shown below; the latter are reported in App. D.4.

Controlled misspecification. We tested four types of misspecification for the Square graph: (i) addition of $Z - W$, (ii) addition of $X \to Y$, (iii) deletion of $X - W$, and (iv) misoriented CPDAG $X - Z$, $Z - Y$, $X \to W$, $W \to Y$. Results are shown in Table 4. The accuracy of the Attention $\mathcal{E}$-NCM changes very little in (i) and (ii). In (iii) and (iv), it degrades more sharply from MAE 0.0098 to MAE ≈ 0.042. This asymmetry is expected, since an overspecified model may still fit the observational data, whereas an underspecified or misoriented model cannot. The Attention $\mathcal{E}$-NCM remains more accurate than the $\mathcal{G}$-NCM baseline in each setting.

PC-learned graphs. We also evaluated an end-to-end pipeline in which the supplied CPDAG is estimated using PC [26] for the Chain and Square settings. We use the PC implementation in `causal-learn` [38], a chi-squared conditional-independence test, and $\alpha = 0.05$, at discovery sample sizes $n = 500$ and $10{,}000$. PC recovers the chain MEC correctly in all except one $n = 500$ trial, where it incorrectly learns a collider graph. PC never recovers the true square MEC, although increasing sample size improves mean Structural Hamming Distance (SHD) from 3.20 to 2.60. Results in Table 5 show that at $n = 500$, the Attention $\mathcal{E}$-NCM's MAE increases approximately three-fold relative to the oracle-input setting for both Chain and Square. Attention $\mathcal{E}$-NCM nevertheless remains more accurate than the $\mathcal{G}$-NCM baseline. At $n = 10{,}000$, Attention $\mathcal{E}$-NCM's errors return to approximately the oracle-CPDAG levels, despite errors in CPDAG recovery.

6 Conclusions

In this work, we introduced a novel characterization of the space of structural causal models over a given set of variables via *masked neural causal models*, capable of expressing any such SCM (Def. 2, Thm. 1). We showed that masked NCMs enable optimization in the space of SCMs consistent with a learned Markov equivalence class and can thus be used to simulate and bound the effects of interventions given such a Markov equivalence class and observational data (Thm. 4). Based on this result, we developed an attention-based architecture and optimization strategy that yields highly accurate estimates of causal bounds in practice (Sec. 5) compared to baselines.

Acknowledgments and Disclosure of Funding

This research is supported in part by the NSF, ONR, AFOSR, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation. We thank the anonymous reviewers for their valuable feedback.

References

[1] Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[2] Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Junwei Ma, Bingru Li, Jesse C. Cresswell, and Rahul Krishnan. CausalPFN: Amortized causal effect estimation via in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[3] Elias Bareinboim. *Causal Artificial Intelligence: A Roadmap for Building Causally Intelligent Systems*. 2025. Draft version (October 11, 2025).

[4] Elias Bareinboim, Juan D Correa, Duligur Ibeling, and Thomas Icard. On pearl's hierarchy and the foundations of causal inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1st edition, 2022.

[5] Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[6] Daniel C. Castro, Jeremy Tan, Bernhard Kainz, Ender Konukoglu, and Ben Glocker. Morpho-mnist: Quantitative assessment and diagnostics for representation learning. *Journal of Machine Learning Research*, 20(178):1–29, 2019.

[7] Fabio De Sousa Ribeiro, Tian Xia, Miguel Monteiro, Nick Pawlowski, and Ben Glocker. High fidelity image counterfactuals with probabilistic causal models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7390–7425. PMLR, 23–29 Jul 2023.

[8] Julien Dubois, Hiroyuki Oya, J. Michael Tyszka, Matthew Howard, Frederick Eberhardt, and Ralph Adolphs. Causal mapping of emotion networks in the human brain: Framework and initial findings. *Neuropsychologia*, 145:106571, 2020. The Neural Basis of Emotion.

[9] Amir Emad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Counting and sampling from markov equivalent dags using clique trees. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[10] Chujun He, Jiaqi Zhang, Munther Dahleh, and Caroline Uhler. Morph predicts the single-cell outcome of genetic perturbations across conditions and data modalities. *bioRxiv*, 2025.

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

[12] Kristian Kristensen and I Rasmussen. A decision support system for mechanical weed control in malting barley. In *Proceedings of the First European Conference on Information Technology in Agriculture*, pages 447–452, 1997.

[13] R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

[14] Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

[15] Daniel Malinsky and Peter Spirtes. Estimating causal effects with ancestral graph Markov models. In Alessandro Antonucci, Giorgio Corani, and Cassio Polpo Campos, editors, *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume 52 of *Proceedings of Machine Learning Research*, pages 299–309, Lugano, Switzerland, 06–09 Sep 2016. PMLR.

[16] Preetam Nandy, Marloes H. Maathuis, and Thomas S. Richardson. Estimating the effect of joint interventions from observational data in sparse high-dimensional settings. *The Annals of Statistics*, 45(2):647 – 674, 2017.

[17] Yushu Pan and Elias Bareinboim. Counterfactual image editing with disentangled causal latent space. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[18] Nick Pawlowski, Daniel C. Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[19] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[20] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, 2nd edition, 2009.

[21] Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(58):2009–2053, 2014.

[22] Robin L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

[23] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3:121–129, 2017.

[24] Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-PFN: In-context learning for causal effect estimation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[25] Jakob Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075310, 07 2018.

[26] P Spirtes, C N Glymour, and R Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

[27] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, Peer Bork, Lars J Jensen, and Christian von Mering. The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Research*, 51(D1):D638–D646, 01 2023.

[28] Keyon Vafa, Susan Athey, and David M. Blei. Estimating wage disparities using foundation models. *Proceedings of the National Academy of Sciences*, 122(22):e2427298122, 2025.

[29] Thomas S. Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 255–270. Elsevier, 1990.

[30] Xiaoqiang Wang, Yali Du, Shengyu Zhu, Liangjun Ke, Zhitang Chen, Jianye Hao, and Jun Wang. Ordering-based causal discovery with reinforcement learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3566–3573, 2021.

[31] Eli N. Weinstein, Elizabeth B. Wood, and David M. Blei. Estimating the causal effects of t cell receptors, 2024.

[32] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

[33] Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: expressiveness, learnability, and inference. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[34] Kevin Muyuan Xia, Yushu Pan, and Elias Bareinboim. Neural causal models for counterfactual identification and estimation. In *The Eleventh International Conference on Learning Representations*, 2023.

[35] Dezhi Yang, Guoxian Yu, Jun Wang, Zhengtian Wu, and Maozu Guo. Reinforcement causal structure learning on order graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10737–10744, 2023.

[36] Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26548–26558. PMLR, 17–23 Jul 2022.

[37] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc.

[38] Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

Appendices

Contents

A Extended Background and Definitions **14**

A.1 Further Definitions 14

B Further Results and Proofs **14**

B.1 Masked NCMs 14

B.2 CPDAG-Constrained Masked NCMs 17

C Architecture and Algorithms **19**

C.1 FFN Masked NCM architecture and hyperparameters 19

C.2 Attention Masked NCM: RL Architecture and Details 19

D Experiments **22**

D.1 Details on data generation 22

D.2 Partial identification accuracy: detailed experiemental results 22

D.3 Decision-making from images 22

D.4 Additional ablation studies 25

D.4.1 Detailed results of CPDAG misspecification 25

D.4.2 Query weight λ_q 26

D.5 Runtime 26

A Extended Background and Definitions

A.1 Further Definitions

Definition 6 (Masked NCM recursive valuation)**.** *Consider a masked NCM* $\mathcal{N} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ *with an acyclic mask* $\mathbf{m}$ *inducing a DAG* $\mathcal{G}$*. Let* $\prec$ *be a topological ordering of* $\mathbf{V}$ *according to* $\mathcal{G}$*. For each* $V_i \in \mathbf{V}$*, let* $\mathbf{Pa}_i := \{V_j \in \mathbf{V} : m_{ji} > 0\}$*. The* recursive valuation $\mathbf{V}^{\prec}(\mathbf{u})$ *for an exogenous assignment* $\mathbf{u} \in$ *dom*$(\mathbf{U})$ *is the unique assignment obtained by evaluating, in topological order,*

$$V_i(\mathbf{u}) = f_i\left(\mathbf{v}_{-i}^{\mathbf{Pa}_i}, u_i\right),$$

where $\mathbf{v}_{-i}^{\mathbf{Pa}_i}$ *denotes the vector in dom*$(\mathbf{V}_{-i})$ *whose* j*-th coordinate is*

$$\left(\mathbf{v}_{-i}^{\mathbf{Pa}_i}\right)_j = \begin{cases} V_j(\mathbf{u}), & V_j \in \mathbf{Pa}_i, \\ 0, & V_j \notin \mathbf{Pa}_i. \end{cases}$$

Equivalently, $V_i(\mathbf{u})$ *is computed by applying* f_i *to the previously computed parent values and setting all masked-out inputs to* 0*.*

The interventional recursive valuation*, for an intervention* $\mathrm{do}(\mathbf{X} = \mathbf{x})$*, is defined in the same way in the intervened model obtained by replacing* f_i *with the constant function* x_i *for every* $X_i \in \mathbf{X}$*.*

B Further Results and Proofs

B.1 Masked NCMs

Proposition 2 (Equivalence of recursive and synchronous valuations for acyclic masks)**.** *Let* $\mathcal{N} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ *be a masked NCM whose mask* $\mathbf{m}$ *induces an acyclic graph* $\mathcal{G}$*. Let* $\prec$ *be any*

topological ordering of $\mathcal{G}$, and let $\mathbf{V}(\mathbf{u})$ denote the recursive valuation computed according to $\prec$. Then, for every exogenous assignment $\mathbf{u} \in dom(\mathbf{U})$, the synchronous valuation of depth d for any $d \geq n$ agrees with the recursive valuation:

$$\mathbf{V}^{(d)}(\mathbf{u}) = \mathbf{V}(\mathbf{u}).$$

More generally, if $\mathcal{N}_{\mathrm{do}(\mathbf{X}=\mathbf{x})}$ denotes the masked NCM obtained by replacing f_i with the constant function x_i for each $X_i \in \mathbf{X}$, then

$$\mathbf{V}^{(n)}_{\mathrm{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}) = \mathbf{V}_{\mathrm{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}).$$

Proof. We prove the observational claim; the interventional claim follows by the same argument after replacing each intervened structural function f_i by the corresponding constant function.

Let the variables be indexed according to a topological ordering $\prec$, so that if $V_j \in \mathbf{Pa}_i$, then $j < i$. Since the mask is acyclic, every parent of V_i precedes V_i in this ordering.

For each i, let ℓ_i denote the length of the longest directed path in $\mathcal{G}$ ending at V_i. Thus $\ell_i = 0$ exactly when V_i has no parents, and otherwise

$$\ell_i = 1 + \max_{V_j \in \mathbf{Pa}_i} \ell_j.$$

Because $\mathcal{G}$ has n vertices and is acyclic, $\ell_i \leq n - 1$ for all i.

We show by induction on ℓ_i that

$$V_i^{(t)}(\mathbf{u}) = V_i(\mathbf{u}) \qquad \text{for all } t \geq \ell_i + 1.$$

First suppose $\ell_i = 0$. Then V_i has no parents. By the masking invariance, f_i depends only on u_i, with all non-parent endogenous inputs set to 0. Hence the first synchronous update gives

$$V_i^{(1)}(\mathbf{u}) = f_i(\mathbf{0}_{-i}, u_i) = V_i(\mathbf{u}),$$

which proves the base case.

Now suppose the claim holds for all variables whose longest-parent-path length is at most k, and let $\ell_i = k + 1$. Every parent $V_j \in \mathbf{Pa}_i$ satisfies $\ell_j \leq k$. Therefore, by the induction hypothesis, for every $t \geq k + 2$,

$$V_j^{(t-1)}(\mathbf{u}) = V_j(\mathbf{u}) \qquad \text{for all } V_j \in \mathbf{Pa}_i.$$

All non-parent inputs to f_i are masked out, equivalently fixed to 0, in both the synchronous and recursive evaluations. Hence, for every $t \geq k + 2 = \ell_i + 1$,

$$\begin{aligned} V_i^{(t)}(\mathbf{u}) &= f_i\Big(\mathbf{V}_{-i}^{(t-1)}(\mathbf{u}), u_i\Big) \\ &= f_i \mid_{\mathbf{v}_{-i} \setminus \mathbf{pa}_i = \mathbf{0}} \left(\mathbf{V}_{\mathbf{pa}_i}(\mathbf{u}), u_i\right) \\ &= V_i(\mathbf{u}). \end{aligned}$$

This proves the induction step.

Since $\ell_i \leq n - 1$ for every i, we have $V_i^{(n)}(\mathbf{u}) = V_i(\mathbf{u})$ for every coordinate i. Therefore, $\mathbf{V}^{(n)}(\mathbf{u}) = \mathbf{V}(\mathbf{u})$. □

Theorem 1. *[Equivalence between masked NCMs and standard SCMs] Consider a masked NCM $\mathcal{N}$ over variables with an acyclic mask inducing a DAG $\mathcal{G}$. Then, there exists an SCM $\mathcal{M}$ over $\mathbf{V}$ such that, for any sets of variables $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ and their values $\mathbf{x}, \mathbf{y}$,*

$$P^{\mathcal{N}}(\mathbf{y}_{\mathbf{x}}^{(d)}) = P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}})$$

where d is the length of the longest directed path in $\mathcal{G}$. The converse also holds: for any SCM $\mathcal{M}$ over $\mathbf{V}$ inducing a causal graph $\mathcal{G}$, there exists a masked NCM $\mathcal{N}$ over $\mathcal{G}$ whose mask induces the graph $\mathcal{G}$ such that for any sets of variables $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ and their values $\mathbf{x}, \mathbf{y}$,

$$P^{\mathcal{N}}(\mathbf{y}_{\mathbf{x}}^{(d)}) = P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}}).$$

Proof. **Forward direction (masked NCM to SCM).** Consider a masked NCM $\mathcal{N} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}_{\theta,\mathbf{m}}, P(\mathbf{U}) \rangle$ with acyclic mask $\mathbf{m}$ inducing graph $\mathcal{G}$. Construct a standard SCM $\mathcal{M} = \langle \mathbf{U}', \mathbf{V}', \mathcal{F}', P(\mathbf{U}') \rangle$ as follows. Let $\mathbf{U}', \mathbf{V}'$ and $P(\mathbf{U}')$ be as in $\mathcal{N}$. Let $f_i \in \mathcal{F}$ be the function determining V_i in $\mathcal{N}$, having the form

$$V_i \leftarrow f_i(\mathbf{V}_{-i}, U_i).$$

For each $V_i \in \mathbf{V}$, let $\mathbf{Pa}_i := \{V_j \mid \mathbf{m}_{ji} > 0\}$. Then, define the functions in the SCM $\mathcal{M}$ as follows. Let each $f_i' : \text{dom}(\mathbf{Pa}_i) \times \text{dom}(U_i) \to \text{dom}(V_i)$ be a mapping defined as

$$f_i'(\mathbf{pa}_i, u_i) = f_i(\mathbf{v}_{-i}, U_i).$$

where $\mathbf{v}_{i-1}$ is the vector of values whose j-th component is $\mathbf{pa}_{i,j}$ if $V_i \in \mathbf{Pa}_i$ and 0 otherwise. By construction, $\mathcal{M}$ induces the graph $\mathcal{G}$. Next, we claim for any $\mathbf{u} \in \text{dom}(\mathbf{U})$, $\mathbf{X} \subseteq \mathbf{V}$, and $\mathbf{x} \in \text{dom}(\mathbf{X})$ the masked NCM synchronous depth-d valuation agrees with the SCM valuation. Consider a topological order consistent with $\mathcal{G}$; this is consistent with both the masked NCM and the constructed SCM. By Prop. 2, the synchronous masked NCM valuation equals the order-based valuation: we have $\mathbf{V}_{\mathbf{x}}^{(d)}(\mathbf{u}) = \mathbf{V}^{\prec}(\mathbf{u})$. We construct f_i' to match f_i with non-parent inputs set to zero; the topological evaluation of $\mathbf{V}^{\prec}(\mathbf{u})$ is analogous, setting non-parent inputs to zero (Def. 6). Therefore, $\mathbf{V}^{\prec}(\mathbf{u}) = \mathbf{V}'^{\prec}(\mathbf{u})$, where the latter is the SCM valuation. Finally, this implies

$$P^{\mathcal{N}}(\mathbf{v}_{\mathbf{x}}^{(d)}) = \sum_{\mathbf{u} \in \text{dom}(\mathbf{U})} \mathbf{1}[\mathbf{V}^{\prec}(\mathbf{u}) = \mathbf{v}] P(\mathbf{u}) = \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \mathbf{1}[\mathbf{V}'^{\prec}(\mathbf{u}') = \mathbf{v}] P(\mathbf{u}) = P^{\mathcal{M}}(\mathbf{v}_{\mathbf{x}})$$

and we are done.

Backward direction (SCM to masked NCM).

Without loss of generality, we can assume that $\mathcal{M}$ is an SCM with discrete and finite $\text{dom}(\mathbf{U})$ [33, Def. 10, Lemma 2]. Let $\mathcal{M}$ induce the graph $\mathcal{G}$. First, construct the mask assignment $\mathbf{m}$ as follows. For every V_i, V_j such that $V_i \to V_j$ in $\mathcal{G}$, set $m_{ij} = 1$. Otherwise, set $m_{ij} = 0$. Thus, $\mathbf{m}$ also induces $\mathcal{G}$. Since $\mathcal{G}$ is acyclic, so is the mask.

Next, we will construct $\mathcal{N} = \langle \mathbf{V}', \mathbf{U}', \mathcal{F}', P(\mathbf{U}') \rangle$ as follows.

1. Let $\mathbf{V}' = \mathbf{V}$.
2. Let $\mathbf{U}'$ contain a variable $U_i \sim \mathcal{U}([0, 1])$, as in Def. **??**.
3. For each $V_i \in \mathbf{V}$, we construct two MLPs.
   - First, let $\mathbf{U}_i \subseteq \mathbf{U}$ be the set of exogenous variables in $\mathcal{M}$ affecting V_i. By [33, Lemma 5], there exists an MLP $f_i^R : \text{dom}(U_i) \to \text{dom}(\mathbf{U}_i)$ mapping $\mathcal{U}([0, 1])$ to $P(\mathbf{U}_i)$, that is, for any $\mathbf{u} \in \text{dom}(\mathbf{U}_i)$, we have $P^{\mathcal{N},\mathbf{m}}(f_i^p(u_i) = \mathbf{u}_i) = P^{\mathcal{M}}(\mathbf{u})$. Then, for any $\mathbf{u} \in \text{dom}(\mathbf{U})$, we have

$$\begin{aligned} P^{\mathcal{M}}(\mathbf{U} = \mathbf{u}) &= \prod_{i=1,\ldots,n} P(\mathbf{U}_i = \mathbf{u}_i) && (\mathbf{U}_i \text{ are jointly independent}) \\ &= \prod_{i=1,\ldots,n} P^{\mathcal{M}}(\mathbf{U}_i = \mathbf{u}_i) \\ &= \prod_{i=1,\ldots,n} P^{\mathcal{N},\mathbf{m}}(f_i^p(U_i) = \mathbf{u}_i) && (\text{By construction of } f_i^p) \end{aligned}$$

   - Next, we want to construct an MLP $f_i^H : \text{dom}(\mathbf{V}_{-i}) \times \text{dom}(\mathbf{U}_i) \to \text{dom}(V_i)$ to simulate $f_i : \text{dom}(\mathbf{Pa}_i) \times \text{dom}(\mathbf{U}_i) \to \text{dom}(V_i)$. Since $\text{dom}(\mathbf{V})$ and $\text{dom}(\mathbf{U}_i)$ are both discrete and finite, by [33, Lemma 4], there exists an MLP

$$f_i'^H : \text{dom}(\mathbf{Pa}_i) \times \text{dom}(\mathbf{U}_i) \to \text{dom}(V_i)$$

     agreeing with f_i. To construct the MLP f_i^H, simply define

$$f_i^H(\mathbf{v}_{-i}, \mathbf{u}_i) = f_i'^H(\mathbf{pa}_i, \mathbf{u}_i)$$

     where $\mathbf{pa}_i$ denotes the values assigned to $\mathbf{Pa}_i$ in $\mathbf{v}_{-i}$. This satisfies the non-parent invariance condition of Def. 2, since f_i^H does not depend on values $\mathbf{v}_{-i} \setminus \mathbf{pa}_i$.

- Finally, we define the MLP $f'_i : \text{dom}(\mathbf{V}_{-i}) \times \text{dom}(U_i)$ as the composition of MLPs

$$f'_i(\mathbf{v}_{-i}, u_i) = f_i^H(\mathbf{v}_{-i}, f_i^R(u_i)) \tag{6}$$

Fix a topological order $\prec$ consistent with the graph $\mathcal{G}$ It remains to show that $P^{\mathcal{N}}(\mathbf{V}_{\mathbf{x}}^{\prime(d)}) = P^{\mathcal{M}}(\mathbf{V}_{\mathbf{x}})$ for any $\mathbf{X} \subseteq \mathbf{V}$ and assignment $\mathbf{x}$.

Since $\mathbf{m}$ is acyclic, it follows from Prop. 2 that for any $\mathbf{u} \in \text{dom}(\mathbf{U})$, we have $\mathbf{V}_{\mathbf{x}}^{\prime(d)}(\mathbf{u}) = \mathbf{V}_{\mathbf{x}}^{\prime\prec}(\mathbf{u})$ and hence $P^{\mathcal{N}}(\mathbf{v}_{\mathbf{x}}^{(d)}) = P^{\mathcal{N}}(\mathbf{v}_{\mathbf{x}}^{\prec})$. Then,

$$\begin{aligned}
P^{\mathcal{N}}(\mathbf{v}_{\mathbf{x}}^{\prec}) &= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \mathbf{1}[\mathbf{V}_{\mathbf{x}}^{\prec}(\mathbf{u}) = \mathbf{v}] P(\mathbf{U}' = \mathbf{u}) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f'_i \mid_{\mathbf{v} \setminus \mathbf{pa}_i = \mathbf{0}} (\mathbf{v}_{-i}, u_i) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{N}, \mathbf{m}}(\mathbf{U}' = \mathbf{u}) \\
&\qquad \text{(By Def. ??, where } \mathbf{Pa_i} \text{ contains } V_j \text{ s.t. } \mathbf{m}_{ji} \neq 0) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i^H(\mathbf{v}_{-i}, f_i^R(u_i)) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{N}, \mathbf{m}}(\mathbf{U}' = \mathbf{u}) \\
&\qquad \text{(By construction of } f'_i) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i'^H(\mathbf{pa}_i, f_i^R(u_i)) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{N}, \mathbf{m}}(\mathbf{U}' = \mathbf{u}) \\
&\qquad \text{(By construction of } f_i^H) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U}')} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i(\mathbf{pa}_i, f_i^R(u_i)) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{N}, \mathbf{m}}(\mathbf{U}' = \mathbf{u}) \\
&\qquad \text{(By construction of } f_i'^H) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U})} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i(\mathbf{pa}_i, \mathbf{u}_i) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{N}, \mathbf{m}}(f^R(\mathbf{U}') = \mathbf{u}) \\
&\qquad \text{(By construction of } f_i^R \text{ and change-of-variables)} \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U})} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i(\mathbf{pa}_i, \mathbf{u}_i) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) \prod_{i=1,\ldots,n} P^{\mathcal{N}, \mathbf{m}}(f_i^R(U_i) = \mathbf{u}_i) \\
&\qquad (U'_i \in \mathbf{U}' \text{ jointly independent}) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U})} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i(\mathbf{pa}_i, \mathbf{u}_i) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) \prod_{i=1,\ldots,n} P^{\mathcal{M}}(\mathbf{U}_i = \mathbf{u}_i) \\
&\qquad \text{(by construction of each } f_i^R) \\
&= \sum_{\mathbf{u} \in \text{dom}(\mathbf{U})} \left(\prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \mathbf{1}[f_i(\mathbf{pa}_i, \mathbf{u}_i) = v_i] \right) \left(\prod_{V_i \in \mathbf{X}} \mathbf{1}[V_i = x_i] \right) P^{\mathcal{M}}(\mathbf{U} = \mathbf{u}) \\
&\qquad (U_i \in \mathbf{U} \text{ jointly independent}) \\
&= P^{\mathcal{M}}(\mathbf{v}_{\mathbf{x}})
\end{aligned}$$

□

B.2 CPDAG-Constrained Masked NCMs

Lemma 1 (MEC of DAGs [29, 1])**.** *Two DAGs are Markov equivalent if and only if they have the same skeleton and the same unshielded colliders.*

Theorem 2. *[Acyclicity constraint ([37, adapted from Thm. 1])] A mask $\mathbf{m} \in [0,1]^{n\times n}$ induces a DAG if and only if*

$$h(\mathbf{m}) = \operatorname{tr}(e^{\mathbf{m}}) - n = 0$$

where $e^{\mathbf{m}}$ is the matrix exponential of $\mathbf{m}$. Furthermore, $h(\mathbf{m})$ is smooth.

Proof. This follows from [37, Thm. 1]. Since our masks are non-negative by construction, the Hadamard product $m \circ m$ is not needed in our case. □

Proposition 1. *[Non-collider constraint] Given a CPDAG $\mathcal{E}$, let $\mathcal{T}_{\mathcal{E}}$ be the set of unshielded non-colliders. A matrix $\mathbf{m} \in [0,1]^{n\times n}$ represents a graph in $\mathcal{E}$ only if*

$$c(\mathbf{m}) = \sum_{(i,j,k)\in\mathcal{T}_{\mathcal{E}}} m_{ij}m_{kj} = 0.$$

Furthermore, $c(\mathbf{m})$ is smooth.

Proof. Let $\mathbf{m} \in [0,1]^{n\times n}$ represent a DAG in the Markov equivalence class encoded by $\mathcal{E}$. Then every unshielded triple in $\mathcal{T}_{\mathcal{E}}$ must remain a non-collider. Hence, for every $(i,j,k) \in \mathcal{T}_{\mathcal{E}}$, the DAG cannot contain both $i \to j$ and $k \to j$. Therefore, $m_{ij}m_{kj} = 0$, and the sum over all such triples $c(\mathbf{m})$ is also zero.

Smoothness follows because $c(W)$ is a finite sum of polynomial functions of the entries of W.

For a single triple (i,j,k), let

$$c_{ijk}(W) = m_{ij}m_{kj}.$$

Then

$$\frac{\partial c_{ijk}}{\partial m_{ij}} = m_{kj}, \qquad \frac{\partial c_{ijk}}{\partial m_{kj}} = m_{ij},$$

with all other partial derivatives equal to zero. Summing over all triples in $\mathcal{T}_{\mathcal{E}}$ gives

$$\frac{\partial c(\mathbf{m})}{\partial m_{ij}} = \sum_{k:(i,j,k)\in\mathcal{T}_{\mathcal{E}}} W_{kj} + \sum_{l:(l,j,i)\in\mathcal{T}_{\mathcal{E}}} W_{lj}.$$

□

Proposition 3. *Let $\mathcal{G}_*$ be the true DAG, let $\mathcal{E}$ be its CPDAG, and suppose $P(\mathbf{V})$ is faithful to $\mathcal{G}_*$. Consider a masked NCM $\mathcal{N}(\theta,\mathbf{m})$ learned by the optimization in Thm. 4, where $\mathbf{m}$ is an acyclic mask satisfying the constraints of the $\mathcal{E}$-subspace. If*

$$P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}) = P(\mathbf{V}),$$

then the DAG $\mathcal{G}_{\mathbf{m}}$ induced by the nonzero entries of $\mathbf{m}$ belongs to the Markov equivalence class represented by $\mathcal{E}$.

Proof. Let $\mathcal{G}_{\mathbf{m}}$ denote the DAG induced by the nonzero entries of $\mathbf{m}$. We show that $\mathcal{G}_{\mathbf{m}} \in \mathcal{E}$.

First, by Def. 5, $\mathcal{G}_{\mathbf{m}}$ cannot contain any adjacency absent from $\mathcal{E}$. Therefore, the skeleton of $\mathcal{G}_{\mathbf{m}}$ is a subgraph of the skeleton of $\mathcal{E}$.

Suppose, for contradiction, that this containment is strict. Then there exist variables $X, Y \in \mathbf{V}$ that are adjacent in $\mathcal{E}$, but not adjacent in $\mathcal{G}_{\mathbf{m}}$. Since $\mathcal{G}_* \in (\mathcal{E})$, X and Y are adjacent in $\mathcal{G}_*$. By faithfulness of $P(\mathbf{V})$ to $\mathcal{G}_*$, this implies that X and Y are not conditionally independent given any subset of $\mathbf{V} \setminus \{X, Y\}$ under $P(\mathbf{V})$.

On the other hand, since X and Y are non-adjacent in the DAG $\mathcal{G}_{\mathbf{m}}$, there exists a set $\mathbf{Z} \subseteq \mathbf{V}\setminus\{X, Y\}$ that d-separates X and Y in $\mathcal{G}_{\mathbf{m}}$. By Thm. 1, the masked NCM $\mathcal{N}(\theta,\mathbf{m})$ is matched by a standard SCM inducing $\mathcal{G}_{\mathbf{m}}$. Hence $P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V})$ is Markov with respect to $\mathcal{G}_{\mathbf{m}}$, and so

$$X \perp\!\!\!\perp Y \mid \mathbf{Z} \qquad \text{under } P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}).$$

Since $P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}) = P(\mathbf{V})$, this implies

$$X \perp\!\!\!\perp Y \mid \mathbf{Z} \qquad \text{under } P(\mathbf{V}),$$

contradicting faithfulness of $P(\mathbf{V})$ to $\mathcal{G}_*$. Therefore, $\mathcal{G}_\mathbf{m}$ and $\mathcal{G}_*$ have the same skeleton.

It remains to show that $\mathcal{G}_\mathbf{m}$ and $\mathcal{G}_*$ have the same unshielded colliders. By construction of the $\mathcal{E}$-subspace, every unshielded collider in $\mathcal{G}_*$ is also an unshielded collider in $\mathcal{G}_\mathbf{m}$. The converse also holds due to the non-collider constraint (Prop. 1). Therefore, $\mathcal{G}_\mathbf{m}$ and $\mathcal{G}_*$ have the same unshielded colliders.

Since two DAGs are Markov equivalent if and only if they have the same skeleton and the same unshielded colliders, we conclude that

$$\mathcal{G}_\mathbf{m} \in \mathcal{E}.$$

□

Theorem 4. *[Partial identification via $\mathcal{E}$-masked NCMs.] Consider a CPDAG $\mathcal{E}$ over n variables $\mathbf{V}$, an observational distribution $P(\mathbf{V})$, and an interventional query $P(\mathbf{y_x})$. Let $\mathcal{E}(\langle\Theta, \mathbf{M}\rangle)$ be the $\mathcal{E}$-subspace of the space of all masked NCMs over $\mathbf{V}$ The optimal interventional bounds (Def. 1 of $P(\mathbf{y_x})$ from $\mathcal{E}$ and $P(\mathbf{V})$ can be derived by solving the following optimization problem:*

$$\min_{\theta, \mathbf{m} \in \mathcal{E}(\langle\Theta, \mathbf{M}\rangle)} / \max \; P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{y_x}) \text{ such that } P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}^{(n)}) = P(\mathbf{V})$$

Proof. Let $[l, r]$ be the true bounds and $[\hat{l}, \hat{r}]$ be the NCM bounds. Consider any NCM and mask assignment $\mathbf{m}$ satisfying $h(\mathbf{m}) = 0,\ nc^{\mathcal{E}}(\mathbf{m}) = 0,$ and $P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}) = P(\mathbf{V})$. Say $\theta, \mathbf{m}$ induce a value $q \in [\hat{l}, \hat{r}]$ for the query $P(\mathbf{y}_)$.

Let $\mathcal{G}$ denote the DAG induced by $\mathbf{m}$. Under the faithfulness assumption, when the NCM fits the observational data $P^{\mathcal{N}(\theta,\mathbf{m})}(\mathbf{V}) = P(\mathbf{V})$, its induced DAG must fall under the Markov equivalence class of models sharing $P(\mathbf{V})$ (Prop. 3). By Lem. 1 and the correctness of the non-collider constraints, we have that $\mathbf{m}$ induces a valid DAG $\mathcal{G}$ in the MEC represented by $\mathcal{E}$. By Thm. 1, there exists an SCM M inducing $\mathcal{G}$ such that $\mathcal{N}(\theta, \mathbf{m})$ and $\mathcal{M}$ agree on all interventional distributions. Since the query value induced by M is accounted for in the true bounds, we have $q \in [l, r]$. Therefore, $[\hat{l}, \hat{r}] \subseteq [l, r]$.

Next, consider any SCM $M \in \Omega(\mathcal{E})$ inducing graph $\mathcal{G}$, distributions $P(\mathbf{V})$, and query value $q \in [l, r]$. Again, by Thm. 1, there exists an $\mathcal{E}$-NCM $\mathcal{N}$ and an $\mathcal{E}$-admissible mask assignment $\mathbf{m}$ such that $N(\theta, \mathbf{m})$ agree with $\mathcal{M}$ on all interventional distributions. Since $N(\theta, \mathbf{m})$ is accounted for in the neural bounds, we have $q \in [\hat{l}, \hat{r}]$. Therefore, $[l, r] \subseteq [\hat{l}, \hat{r}]$ and we are done. □

C Architecture and Algorithms

C.1 FFN Masked NCM architecture and hyperparameters

We give the pseudocode for the feed-forward masked NCM training in Alg. 1 and hyperparameters in Table 1. A key design choice is using an alternating optimization over the mask $\mathbf{m}$ and the neural network parameters θ. Since any interventional query is uniquely identified given a known graph in the Markovian setting, we update θ based only on fit to the data, and not based on the query. The mask $\mathbf{m}$ is updated on all loss terms: data fit, acyclicity, colliders, and query. Finally, we consistently apply a post-processing step to switch learned min/max values in any runs where thier order is flipped.

C.2 Attention Masked NCM: RL Architecture and Details

Notation. Throughout this section, $m \in \{0, 1\}^{n \times n}$ denotes a hard mask, and $P_{\theta,m}$ is shorthand for the distributions $P^{\mathcal{N}(\theta,m)}$ induced by the attention $\mathcal{E}$-NCM with parameters θ and mask m, including the interventional distribution $P_{\theta,m}(\mathbf{y_x})$. We write $\sigma(z) = 1/(1 + e^{-z})$ for the logistic function, $[z]_+ = \max\{z, 0\}$, $\mathbb{1}[\cdot]$ for the indicator function, and $\mathrm{Bern}(b; p) = p^b(1-p)^{1-b}$ for $b \in \{0, 1\}$ and $p \in [0, 1]$.

Policy over masks. Let $\mathcal{A}(\mathcal{E})$ be the set of ordered pairs (i, j), meaning $V_i \to V_j$, that are allowed by $\mathcal{E}$: both (i, j) and (j, i) for every undirected edge $V_i - V_j$, and only (i, j) for every directed edge $V_i \to V_j$. The structure policy has order scores $\rho \in \mathbb{R}^n$ and an edge logit $\psi_{ij} \in \mathbb{R}$ for every

Algorithm 1 Masked FF-NCM bound optimization

Input: Dataset $\mathcal{D}$, CPDAG $(\mathcal{E})$, query Q, variables V
Input: Learning rates η_θ, η_M, acyclicity weight λ_h, collider weight λ_c query weights schedule $\lambda_q(t)$,
Initialize a neural network f_j for each $X_j \in V$
Each f_j takes candidate inputs $\{X_i : i \neq j\}$ and node-specific noise U_j
Initialize soft mask $M \in [0,1]^{|V|\times|V|}$
for all directed edges $i \to j \in E_{\text{dir}}$ **do**
    Set $M_{ij} = 1$ and $M_{ji} = 0$
end for
for all non-skeleton pairs $\{i, j\}$ **do**
    Set $M_{ij} = M_{ji} = 0$
end for
for all undirected edges $\{i, j\} \in E_{\text{undir}}$ **do**
    Introduce one learned coupling parameter α_{ij}
    Set $M_{ij} = \sigma(\alpha_{ij})$ and $M_{ji} = 1 - \sigma(\alpha_{ij})$
end for
for all bound directions $b \in \{\min, \max\}$ **do**
    Initialize neural parameters θ and mask parameters ϕ
    for $t = 1, \ldots, T$ **do**
        Theta phase:
        Sample minibatches from $\mathcal{D}$ and from the masked FF-NCM
        Using synchronous sampling, compute

$$\mathcal{L}_{\text{fit}} = \frac{1}{|\mathcal{I}|} \sum_{do \in \mathcal{I}} \text{MMD}\Big(\mathcal{D}_{do}, \widehat{\mathcal{D}}_{do}(\theta, M(\phi))\Big).$$

        Update θ using $\nabla_\theta \mathcal{L}_{\text{fit}}$
        Mask phase:
        Recompute $\mathcal{L}_{\text{fit}}$ with current θ and $M(\phi)$
        Estimate query loss $\mathcal{L}_Q$ by Monte Carlo samples from the masked FF-NCM
        Compute acyclicity penalty $h(M) = \text{tr}(\exp(M)) - |V|$
        Compute any equivalence-class non-collider penalty $\mathcal{L}_{\text{nc}}$
        Update ϕ using

$$\nabla_\phi \left[\mathcal{L}_{\text{fit}} + \lambda_Q(t)\mathcal{L}_Q + \lambda_h h(\mathbf{m})) + \lambda_c c(\mathbf{m}))\right].$$

    end for
    Freeze $M(\phi)$ and optionally refine θ with mask fixed
end for
return trained min/max masked FF-NCMs and their query estimates

$(i, j) \in \mathcal{A}(\mathcal{E})$. A topological order is a permutation π of $\{1, \ldots, n\}$, where $\pi(p)$ is the index of the variable at position p. It is drawn from the Plackett–Luce distribution [22, 13]

$$q_\rho(\pi) = \prod_{p=1}^{n} \frac{\exp(\rho_{\pi(p)})}{\sum_{p'=p}^{n} \exp(\rho_{\pi(p')})},$$

under which a larger ρ_i places V_i earlier. Let $r_\pi(i)$ be the position of V_i in π, so that $\pi(r_\pi(i)) = i$, and let $\mathcal{A}(\pi) = \{(i, j) \in \mathcal{A}(\mathcal{E}) : r_\pi(i) < r_\pi(j)\}$ be the allowed pairs that point forward in π. Given π, each m_{ij} with $(i, j) \in \mathcal{A}(\pi)$ is drawn independently from $\text{Bern}(\cdot\,; \sigma(\psi_{ij}))$, and all other entries of m are zero. Every edge of m points forward in π, so m is acyclic. The log-probability of the sampled pair (π, m) under the policy is

$$\log q_{\rho,\psi}(\pi, m) = \log q_\rho(\pi) + \sum_{(i,j)\in\mathcal{A}(\pi)} \log \text{Bern}\big(m_{ij}; \sigma(\psi_{ij})\big).$$

Table 1: Masked FF-NCM hyperparameters.

Hyperparameter	1-D setting	8-D setting
Endogenous dimension	$d = 1$	$d = 8$
Scalar variables	all variables	X, Y scalar
High-dimensional variables	none	W, Z with dimension 8
Exogenous noise dimension	1	8
Number of samples	10,000	100,000
Data batch size	1,000	1,000
NCM batch size	1,000	1,000
Hidden layers	2	2
Hidden width	128	128
Optimizer	AdamW	AdamW
Neural learning rate η_θ	4×10^{-3}	4×10^{-3}
Mask learning rate η_M	0.1	0.1
Masking rule	multiply	multiply
Mask initialization	Uniform$(0.1, 0.9)$	Uniform$(0.1, 0.9)$
Acyclicity penalty	NOTEARS [37]	NOTEARS [37]
DAG penalty weight λ_{DAG}	0.1	0.1
Mask ℓ_1 weight	0	0
Non-collider penalty weight	0.1	0.1
Theta:Mask update ratio	1:1	1:1
Query-gradient target	mask only	mask only
Query weight schedule	$10^{-2} \to 10^{-4}$	$10^{-2} \to 10^{-4}$
Max epochsquery iterations	1000	1000
Theta-only refinement	50 epochs	50 epochs

Fit and query under a sampled mask. In the attention $\mathcal{E}$-NCM, each U_i is uniform on $[0,1]^{D_U}$ and is embedded by a per-variable MLP (Sec. 4.2).[3] For a binary V_i, the output head returns the probability $P_{\theta,m}(V_i = 1 \mid \mathbf{v}_{-i}, u_i)$, which, by mask invariance, depends on $\mathbf{v}_{-i}$ only through the parents of V_i under m; vector-valued variables use one such output per coordinate. This is consistent with the deterministic mechanisms of Def. 2: if U_i is extended by one more uniform coordinate ξ_i, the function $f_i(\mathbf{v}_{-i}, u_i, \xi_i) = \mathbb{1}\big[\xi_i < P_{\theta,m}(V_i = 1 \mid \mathbf{v}_{-i}, u_i)\big]$ is deterministic and induces the same conditional distribution. For a minibatch $\mathcal{B} \subseteq \mathcal{D}$, the observational negative log-likelihood integrates ξ_i out in closed form and averages over N_u independent draws $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(N_u)} \sim P(\mathbf{U})$:

$$\mathrm{NLL}_{\theta,m} = -\frac{1}{|\mathcal{B}|} \sum_{\mathbf{v} \in \mathcal{B}} \log\Big(\frac{1}{N_u} \sum_{l=1}^{N_u} \prod_{i=1}^{n} P_{\theta,m}\big(v_i \mid \mathbf{v}_{-i}, u_i^{(l)}\big) \Big).$$

The query value $Q_{\theta,m} = P_{\theta,m}(\mathbf{y}_\mathbf{x})$ is computed in the intervened model by summing exactly over the remaining endogenous variables and averaging over N_u draws of $\mathbf{U}$. It enters training only through the reward below and is never differentiated.

Reward and loss. Let $s = +1$ when computing the upper bound and $s = -1$ when computing the lower bound. The reward of a sampled mask is

$$R(m) = \lambda_q \, s \log Q_{\theta,m} - \lambda_{\mathrm{nll}} \big[\mathrm{NLL}_{\theta,m} - \tau - \epsilon\big]_+ - \lambda_c \, c_{\mathrm{hard}}(m).$$

Here $\lambda_q, \lambda_{\mathrm{nll}}, \lambda_c \geq 0$ are weights, and the first two are adapted during training (see below). The NLL target $\tau = \hat{H}(\mathcal{D}) + \delta$ is the empirical entropy $\hat{H}(\mathcal{D}) = -\sum_{\mathbf{v}} \hat{P}(\mathbf{v}) \log \hat{P}(\mathbf{v})$ of the training data plus a margin $\delta > 0$, and $\epsilon \geq 0$ is a slack. The count $c_{\mathrm{hard}}(m)$ is the number of collider constraints of $\mathcal{E}$ that m violates: unshielded non-colliders of $\mathcal{E}$ that m orients as colliders, which is the hard counterpart of the penalty c of Prop. 1 weighted by λ_c in Sec. 4, plus unshielded colliders of $\mathcal{E}$ that m does not orient as colliders. We keep a baseline b, initialized to the first reward and then updated as $b \leftarrow \kappa b + (1 - \kappa) R(m)$ with a decay $\kappa \in [0, 1)$. Treating the centred reward $\tilde{R}(m) = R(m) - b$

[3] Any continuous noise distribution gives the same model class, since the embedding MLP can approximate the required change of variables; our implementation draws Gaussian noise.

as a constant, the loss is

$$\mathcal{L} = \mathrm{NLL}_{\theta,m} \;-\; \tilde{R}(m) \log q_{\rho,\psi}(\pi, m) \;+\; \lambda_h \, h\big(\sigma(\psi)\big) \;-\; \beta_H \, H(q_\rho).$$

The second term is the REINFORCE score-function estimator [32]. In the third term, $\sigma(\psi) \in [0,1]^{n\times n}$ is the matrix with entries $\sigma(\psi_{ij})$ for $(i,j) \in \mathcal{A}(\mathcal{E})$ and zero elsewhere, and h is an acyclicity function weighted by $\lambda_h \geq 0$, as in Sec. 3.2; we use the log-determinant characterization of [5] in place of the trace-exponential form given there. In the last term, $H(q_\rho) = -\mathbb{E}_{\pi\sim q_\rho}[\log q_\rho(\pi)]$ is the entropy of the order distribution, and its weight $\beta_H \geq 0$ decreases linearly to zero over the first T_H epochs. Because the sampled mask is held fixed in the forward pass, $\mathrm{NLL}_{\theta,m}$ updates only θ, and the remaining terms update only (ρ, ψ). Hence θ is fitted to the data alone, and the query influences training only through the choice of structure.

Adaptive weights. After every step, λ_{nll} takes a projected dual-ascent step with step size $\eta_{\mathrm{nll}} > 0$,

$$\lambda_{\mathrm{nll}} \leftarrow \min\Big\{\lambda_{\mathrm{nll}}^{\max},\, \max\big\{\lambda_{\mathrm{nll}}^{\min},\, \lambda_{\mathrm{nll}} + \eta_{\mathrm{nll}}\big(\mathrm{NLL}_{\theta,m} - \tau - \epsilon\big)\big\}\Big\},$$

where $0 < \lambda_{\mathrm{nll}}^{\min} \leq \lambda_{\mathrm{nll}}^{\max}$ are bounds; λ_{nll} therefore grows while the fit constraint $\mathrm{NLL}_{\theta,m} \leq \tau + \epsilon$ is violated and shrinks otherwise. Once per epoch, λ_q is updated from an exponential moving average $\overline{\mathrm{NLL}}$ of the training NLL. If $\overline{\mathrm{NLL}} \leq \tau + \epsilon$, it grows geometrically with rate $g > 0$, $\lambda_q \leftarrow \min\{(1+g)\lambda_q, \lambda_q^{\max}\}$; otherwise it decreases with step size $\eta_q > 0$, $\lambda_q \leftarrow \max\{\lambda_q - \eta_q(\overline{\mathrm{NLL}} - \tau - \epsilon), \lambda_q^{\min}\}$, where $0 < \lambda_q^{\min} \leq \lambda_q^{\max}$ are bounds. Query pressure therefore increases only while the model fits the data. In the ablation of App. D.4.2, λ_q and λ_{nll} are instead held constant.

Freezing the structure. At the end of each epoch, we decode the MAP mask $\hat{m}$: the order that sorts ρ in decreasing order, with each forward allowed pair included if and only if $\sigma(\psi_{ij}) > 1/2$. We call $\hat{m}$ *eligible* if (i) at least $T_{\min}$ epochs have passed; (ii) $\hat{m}$ has not changed over the last w epochs; (iii) every forward allowed pair has confidence $\max\{\sigma(\psi_{ij}), 1 - \sigma(\psi_{ij})\} \geq p_{\min}$, for a threshold $p_{\min} \in [1/2, 1)$; and (iv) the training NLL stayed below $\tau + \epsilon_f$ over the last w epochs, for a freezing slack $\epsilon_f \geq 0$. Every eligible mask is stored, together with its current query value $Q_{\theta,\hat{m}}$, in a candidate set $\mathcal{C}$. The first time the current MAP mask is eligible, or after $T_{\max}$ epochs, we freeze the structure to $m^\star = \arg\max_{m\in\mathcal{C}} s\, Q_{\theta,m}$, i.e., the stored mask with the largest query value when maximizing and the smallest when minimizing; if $\mathcal{C}$ is empty, $m^\star = \hat{m}$. This selection uses only the model's own query estimates, never the true bounds.

Refinement and readout. After freezing, the policy terms vanish, and θ is trained on $\mathrm{NLL}_{\theta,m^\star}$ alone for at most T_{post} epochs, with a smaller learning rate that is decayed on plateaus. We keep the checkpoint whose observational distribution has the smallest KL divergence to the empirical distribution $\hat{P}$ of the data. The reported bound is $P_{\theta,m^\star}(\mathbf{y}_\mathbf{x})$, computed exactly over the endogenous variables with 10^4 Monte Carlo draws of $\mathbf{U}$. Running the procedure with $s = +1$ and with $s = -1$ gives the upper and lower bounds. Alg. 2 summarizes the procedure.

D Experiments

D.1 Details on data generation

In all generated datasets, we enforce that there exists a gap between the true upper and lower bounds of the query. We use a regional canonical model with `c2-scale` 2.0. For an extended description of such models, see [34, Sec. B.3].

D.2 Partial identification accuracy: detailed experiemental results

In Figs. 3, we provide detailed bound recovery results for the various methods, whose MAE is summarized in Fig. 1.

D.3 Decision-making from images

Data generation. To sample I_1, we draw an MNIST training digit, from classes $\{0,\dots,4\}$ with probability p_Z and from $\{5,\dots,9\}$ otherwise, and standardise it to an upright posture and a stroke

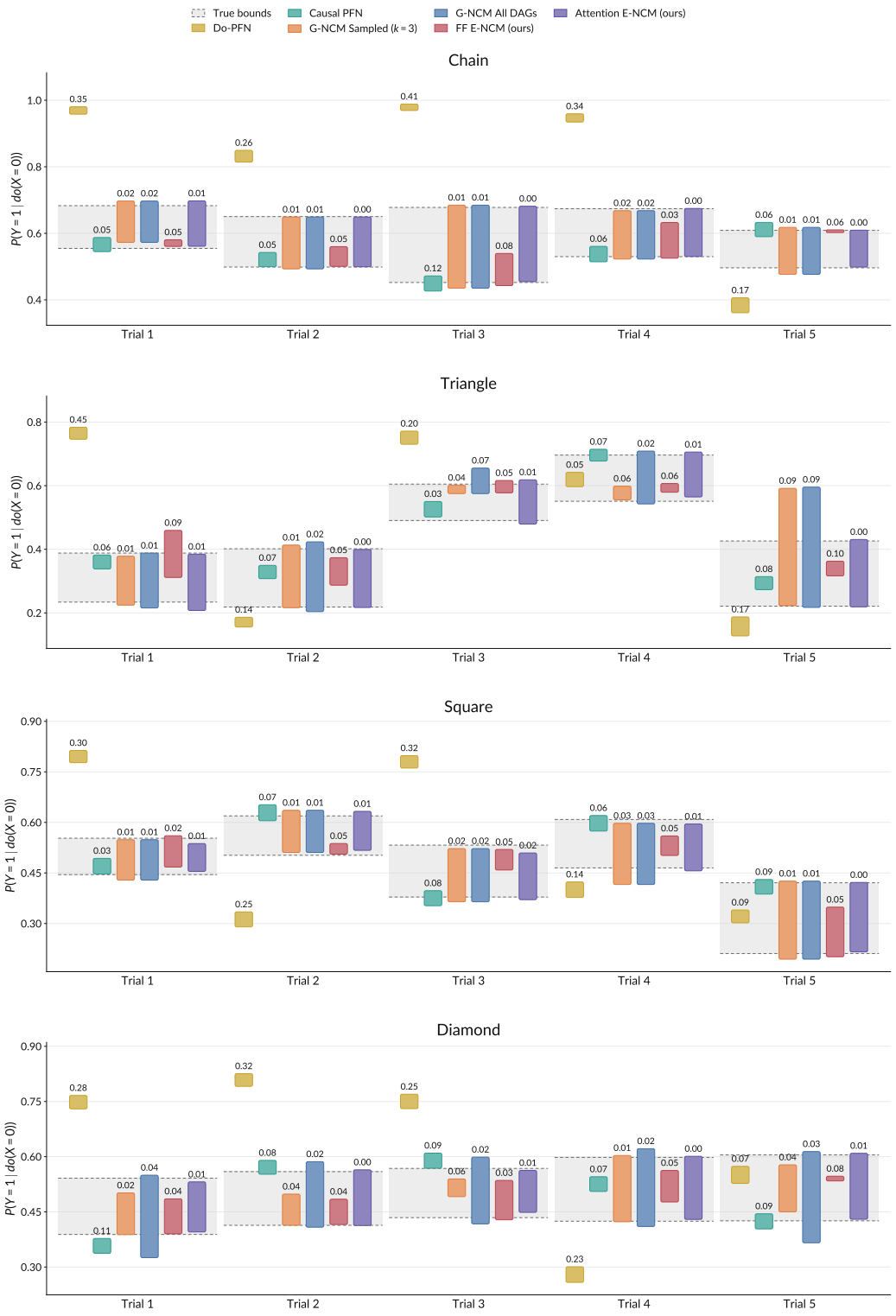


Figure 3: Bounds recovered by various methods in Exp. 5.1. The Attention $\mathcal{E}$-NCM shows near-exact recovery of the ground truth bounds.

Algorithm 2 Attention $\mathcal{E}$-NCM bound optimization with RL structure search

Input: Data $\mathcal{D}$, CPDAG $\mathcal{E}$, query $P(\mathbf{y_x})$, direction $s \in \{+1, -1\}$

Initialize θ, ρ, ψ, λ_q, λ_{nll}; set $\mathcal{C} \leftarrow \emptyset$ and $\tau \leftarrow \hat{H}(\mathcal{D}) + \delta$
for each epoch, until the structure is frozen **do**
  for each minibatch $\mathcal{B} \subseteq \mathcal{D}$ **do**
    Sample $\pi \sim q_\rho$, then $m_{ij} \sim \text{Bern}(\cdot\,; \sigma(\psi_{ij}))$ for $(i,j) \in \mathcal{A}(\pi)$ ▷ m is acyclic
    Compute $\text{NLL}_{\theta,m}$ on $\mathcal{B}$, and $Q_{\theta,m}$ without gradient
    $R \leftarrow \lambda_q s \log Q_{\theta,m} - \lambda_{\text{nll}}[\text{NLL}_{\theta,m} - \tau - \epsilon]_+ - \lambda_c c_{\text{hard}}(m)$
    $b \leftarrow \kappa b + (1-\kappa)R$; $\tilde{R} \leftarrow R - b$
    Gradient step on $\text{NLL}_{\theta,m} - \tilde{R} \log q_{\rho,\psi}(\pi, m) + \lambda_h h(\sigma(\psi)) - \beta_H H(q_\rho)$
    Update λ_{nll} by projected dual ascent
  end for
  Update λ_q from $\overline{\text{NLL}}$, and decrease β_H
  Decode the MAP mask $\hat{m}$; if it is eligible, add $(\hat{m}, Q_{\theta,\hat{m}})$ to $\mathcal{C}$
  if $\hat{m}$ is eligible **or** $T_{\max}$ epochs have passed **then**
    Freeze $m^\star \leftarrow \arg\max_{m \in \mathcal{C}} s\, Q_{\theta,m}$ ▷ $m^\star \leftarrow \hat{m}$ if $\mathcal{C} = \emptyset$
  end if
end for
Train θ on $\text{NLL}_{\theta,m^\star}$ alone for at most T_{post} epochs; keep the checkpoint closest in KL to $\hat{P}$
return $P_{\theta,m^\star}(\mathbf{y_x})$

thickness of 3.2 pixels using Morpho-MNIST [6]. We then independently slant it by 0.5 rad with probability p_X, and draw it in red rather than green with probability p_Y. The mechanisms downstream of I_1 respond to defects measured from the pixels of an image I: the slant from its second moments, the color from its channel intensities, and the number from a convolutional classifier (98.4% accuracy on the MNIST test set). We write $b_X(I), b_Y(I), b_Z(I) \in \{0,1\}$ for these measured defects. Given I_1, the genes $G_k \in \{X, Y, Z\}$ are sampled in the order $Z \to X \to Y$, with

$$P(G_k = 1 \mid I_1, \text{earlier genes}) = \sigma\Big(\beta_0 + \beta_1\, b_k(I_1) + \gamma \sum_{j \prec k} G_j\Big), \qquad \beta_0 = -2,\ \beta_1 = 4.5,\ \gamma = -1.5.$$

A gene therefore activates on 92% of the digits with its defect when no earlier gene is active (73% and 38% with one or two active earlier genes), and on 12% of the digits without it. The active genes then edit the pixels of I_1 to produce I_2. X applies a fixed counter-slant, which makes a slanted digit upright, but makes an upright digit lean the other way; we write $\ell(I_2) = 1$ for such an over-corrected digit. Y swaps red and green, so a green digit that did not need it becomes red. Z replaces a low digit d by a freshly drawn MNIST digit of class $d + 5$ with the same slant and color, and leaves high digits unchanged. Finally,

$$P(R = 1 \mid I_2) = 0.03 + 0.94 \prod_{k \in \{X,Y,Z\}} (1 - s_k)^{b_k(I_2)}\,(1 - h)^{\ell(I_2)},$$

where s_k is the cost of an uncorrected defect and h the cost of leaning back. We use $(p_X, p_Y, p_Z) = (0.8, 0.5, 0.4)$, $(s_X, s_Y, s_Z) = (0.9, 0.9, 0.5)$, and $h = 0.1$, which gives $P(R = 1) = 0.700$. These parameters were selected as a case in which the cautious and optimistic policies make different choices; the robustness study below shows that the qualitative conclusions hold for randomly drawn mechanisms. Since every variable after I_1 is a function of the pixels and of independent noise (including the fresh digit drawn by Z), the true model is Markovian.

Image encoding and model training. I_1 and I_2 enter the model as 16-bit binary codes from a convolutional autoencoder: two strided convolutions and two dense layers map an image to 16 bits by straight-through thresholding, and a mirrored decoder reconstructs the image. We train the autoencoder on both reconstruction of the image and the observed variables X, Y, Z, R. We use the attention $\mathcal{E}$-NCM trained with reinforcement learning, with one attention layer with four heads and 64-dimensional tokens, and 24-dimensional exogenous noise embedded by an MLP with two hidden layers of width 256.

Learned bounds. Table 2 reports the learned and true bounds for every gene set, together with the true value and the observed rate used by the naive policy. For 10 of the 12 runs whose query depends

on the gene ordering, the frozen mask corresponds to an ordering attaining the true bound; the two exceptions concern straightening, whose true bounds are only 0.031 apart, and miss by 0.002 and 0.010. The learned bounds have a mean absolute error of 0.034, and are biased upwards for gene sets that include X or Y (mean signed error $+0.037$, maximum 0.070), while the bounds for Z alone, which cannot harm, are accurate to within 0.010. We hypothesise that this reflects limited support: X and Y are active on only 8% and 3% of the digits without their defect, so the harm they cause is rarely observed. The bias does not change any decision: the gene sets with the largest errors remain well below doing nothing, and the error for straightening, the cautious choice, is at most 0.016.

Table 2: Learned and true bounds on $P(R = 1 \mid do(S = 1))$ for every gene set S, the true value under the true DAG, and the observed rate $P(R = 1 \mid S = 1)$ used by the naive policy. Doing nothing has value 0.702. For $S = \{X, Y, Z\}$ the query is identified.

S	Learned bounds	True bounds	True value	Observed rate
$\{X\}$	$[0.767, 0.776]$	$[0.751, 0.782]$	0.753	0.773
$\{Y\}$	$[0.463, 0.505]$	$[0.441, 0.474]$	0.474	0.785
$\{Z\}$	$[0.592, 0.704]$	$[0.602, 0.713]$	0.602	0.601
$\{X, Y\}$	$[0.579, 0.591]$	$[0.510, 0.524]$	0.524	0.917
$\{X, Z\}$	$[0.711, 0.795]$	$[0.690, 0.794]$	0.690	0.703
$\{Y, Z\}$	$[0.428, 0.522]$	$[0.382, 0.482]$	0.443	0.692
$\{X, Y, Z\}$	0.602	0.532	0.532	0.949

Robustness across mechanisms. We also evaluate the policies on $3{,}000$ mechanisms drawn at random from the same family, using exact bounds rather than learned ones. We draw $p_k \sim U(0.2, 0.8)$, $s_k \sim U(0.3, 0.9)$, $\beta_0 \sim U(-2.5, -1.5)$, $\beta_1 \sim U(4, 5.5)$, and a separate inhibition strength $\gamma \sim U(-2.5, 0)$ for each pair of genes. In the *color* family, X harms a digit without its defect by $h \sim U(0, 0.3)$, Y by s_Y, and Z not at all, as above. In the *general* family, every gene k harms by an amount drawn from $U(0, s_k)$. Table 3 shows the results. With exact bounds, the cautious policy is never worse than doing nothing, since the value of doing nothing is known. In the color family it has near-zero regret, whereas the optimistic policy and a policy that trusts a single DAG drawn at random from $\mathcal{E}$ choose an intervention worse than doing nothing in 46% and 24% of the mechanisms; the naive policy does so in 88%, while overestimating the value of its choice by 27 points on average. In the general family, where every gene can harm, the cautious policy is more conservative than necessary: it activates nothing in 69% of the mechanisms, against 57% for the oracle, and its regret exceeds that of the optimistic policy. The naive policy remains the worst in both families.

Table 3: Policies evaluated with exact bounds on $3{,}000$ random mechanisms per family: share of mechanisms in which the chosen intervention is worse than doing nothing, and mean regret relative to the oracle, in percentage points.

	Color family		General family	
Policy	Worse than nothing	Regret	Worse than nothing	Regret
Do nothing	0.0%	1.76	0.0%	2.06
Cautious (ours)	0.0%	**0.05**	0.0%	0.35
Optimistic (ours)	46.3%	1.65	8.2%	**0.18**
Random DAG in $\mathcal{E}$	24.1%	0.82	4.2%	0.26
Naive	87.9%	17.23	83.7%	15.00

D.4 Additional ablation studies

D.4.1 Detailed results of CPDAG misspecification

In Tables 4 and 5, we provide detailed results for the CPDAG misspecification experiments discussed in Sec. 5.4.

Table 4: Average of the lower- and upper-bound MAE ($\pm$ one standard deviation across runs), under controlled CPDAG misspecification on the Square graph; lower is better.

CPDAG setting	$\mathcal{G}$-NCM	Attention $\mathcal{E}$-NCM
oracle	0.017 ± 0.017	**0.0098** ± 0.0047
(i) added $Z - W$	0.018 ± 0.008	**0.010** ± 0.005
(ii) added $X \to Y$	0.029 ± 0.026	**0.0075** ± 0.0065
(iii) deleted $X - W$	0.055 ± 0.042	**0.041** ± 0.045
(iv) misorientation	0.045 ± 0.039	**0.043** ± 0.044

Table 5: Average of the lower- and upper-bound MAE ($\pm$ one standard deviation across runs), with oracle and PC-learned CPDAGs, where n is the sample size used by PC; lower is better.

CPDAG setting	**Graph**	$\mathcal{G}$-NCM	Attention $\mathcal{E}$-NCM
oracle	Chain	0.014 ± 0.006	**0.0037** ± 0.0037
	Square	0.017 ± 0.017	**0.0098** ± 0.0047
PC $n = 500$	Chain	0.030 ± 0.033	**0.018** ± 0.030
	Square	0.034 ± 0.041	**0.031** ± 0.039
PC $n = 10{,}000$	Chain	0.014 ± 0.006	**0.0037** ± 0.0037
	Square	0.014 ± 0.007	**0.0094** ± 0.0048

D.4.2 Query weight λ_q

The reward used by the structure policy (App. C.2) combines a query term, $\lambda_q s \log Q_{\theta,m}$, with an observational-fit term, $-\lambda_{\text{nll}} \left[\text{NLL}_{\theta,m} - \tau - \epsilon\right]_+$, with notation as in App. C.2. To test sensitivity to the balance between the two, we fix $\lambda_{\text{nll}} = 10$ and sweep $\lambda_q \in \{0, 0.001, 0.01, 0.05, 0.1, 0.5, 1, 5\}$ on the Diamond graph, keeping all other hyperparameters at their default values. Results are reported in Table 6.

Table 6: Sensitivity of Attention $\mathcal{E}$-NCM to the query weight λ_q on the Diamond graph, with $\lambda_{\text{nll}} = 10$. We report the mean absolute error (MAE) of the lower and upper bounds and their average. (Numbers to be updated.)

λ_q	0	0.001	0.01	0.05	0.1	0.5	1	5
Lower MAE	0.0829	0.1645	0.1294	0.0022	0.0011	0.0014	0.0014	0.0006
Upper MAE	0.0870	0.0055	0.0063	0.0054	0.0056	0.0064	0.0061	0.0064
Average MAE	0.0849	0.0850	0.0679	0.0038	0.0033	0.0039	0.0037	0.0035

For $\lambda_q \leq 0.01$, the query signal is too weak to recover both extrema reliably; at $\lambda_q = 0$ the maximization and minimization problems coincide, and the method no longer separates the two bounds. A sharp transition occurs at $\lambda_q = 0.05$, where the average MAE falls below 0.004 and each endpoint MAE is below 0.006. Performance then remains stable over the range $[0.05, 5]$. Attention $\mathcal{E}$-NCM is therefore insensitive to the choice of λ_q, provided it exceeds a threshold.

D.5 Runtime

Fig. 4 reports the average time per epoch and the average total training time of the three neural approaches. The sampled $\mathcal{G}$-NCM baseline is the fastest on the four small graphs. On the 48-node Barley network it is the slowest: training its $k = 5$ networks takes 25.1 h, compared with 9.4 h for FF $\mathcal{E}$-NCM and 16.7 h for Attention $\mathcal{E}$-NCM, and it is also less accurate than Attention $\mathcal{E}$-NCM in this setting (Fig. 1g). Attention $\mathcal{E}$-NCM has the highest cost per epoch on Barley but converges in fewer epochs.

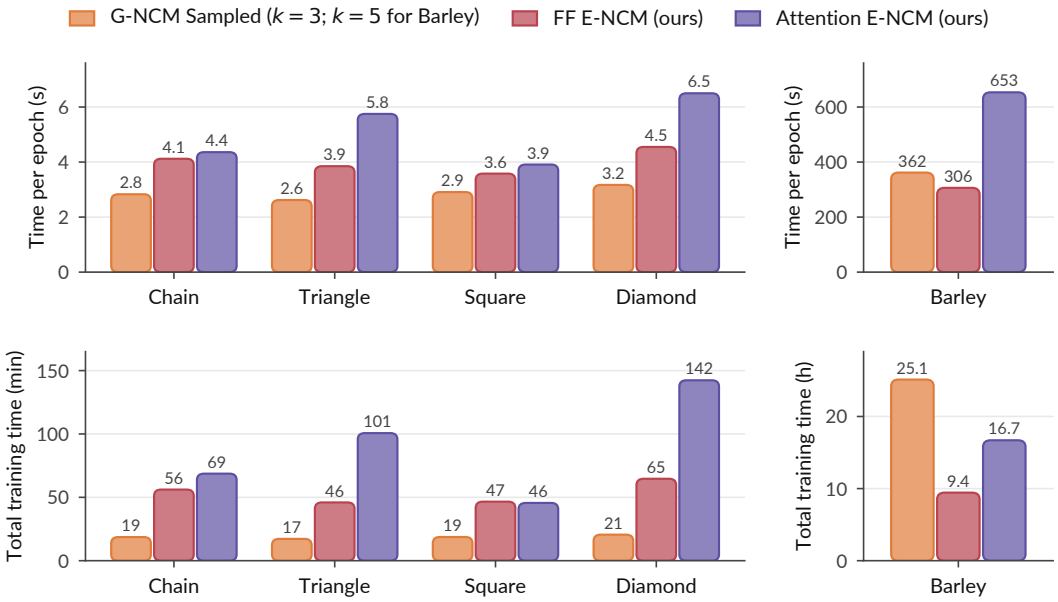


Figure 4: Average time per epoch (top) and average total training time (bottom) on the four small graphs (left) and the 48-node Barley network (right). Each bar is labelled with its value.