The Causal-Neural Connection: Expressiveness, Learnability, and Inference

Kevin Xia
CausalAI Lab
Columbia University
kmx2000@columbia.edu

Kai-Zhan Lee
Bloomberg L.P.
Columbia University
kl2792@columbia.edu

Yoshua Bengio
MILA
Université de Montréal
yoshua.bengio@mila.quebec

Elias Bareinboim
CausalAI Lab
Columbia University
eb@cs.columbia.edu

Abstract

One of the central elements of any causal inference is an object called structural causal model (SCM), which represents a collection of mechanisms and exogenous sources of random variation of the system under investigation (Pearl, 2000). An important property of many kinds of neural networks is *universal approximability*: the ability to approximate any function to arbitrary precision. Given this property, one may be tempted to surmise that a collection of neural nets is capable of learning any SCM by training on data generated by that SCM. In this paper, we show this is not the case by disentangling the notions of expressivity and learnability. Specifically, we show that the causal hierarchy theorem (Thm. 1, Bareinboim et al., 2020), which describes the limits of what can be learned from data, still holds for neural models. For instance, an arbitrarily complex and expressive neural net is unable to predict the effects of interventions given observational data alone. Given this result, we introduce a special type of SCM called a neural causal model (NCM), and formalize a new type of inductive bias to encode structural constraints necessary for performing causal inferences. Building on this new class of models, we focus on solving two canonical tasks found in the literature known as causal identification and estimation. Leveraging the neural toolbox, we develop an algorithm that is both sufficient and necessary to determine whether a causal effect can be learned from data (i.e., causal identifiability); it then estimates the effect whenever identifiability holds (causal estimation). Simulations corroborate the proposed approach.

1 Introduction

One of the most celebrated and relied upon results in the science of intelligence is the universality of neural models. More formally, universality says that neural models can approximate any function (e.g., boolean, classification boundaries, continuous valued) with arbitrary precision given enough capacity in terms of the depth and breadth of the network [14, 26, 44, 50]. This result, combined with the observation that most tasks can be abstracted away and modeled as input/output – i.e., as functions – leads to the strongly held belief that under the right conditions, neural networks can solve the most challenging and interesting tasks in AI. This belief is not without merits, and is corroborated by ample evidence of practical successes, including in compelling tasks in computer vision [40], speech recognition [22], and game playing [51]. Given that the universality of neural nets is such a compelling proposition, we investigate this belief in the context of causal reasoning.

To start understanding the causal-neural connection – i.e., the non-trivial and somewhat intricate relationship between these modes of reasoning – two standard objects in causal analysis will be instrumental. First, we evoke a class of generative models known as the *Structural Causal Model* (SCM, for short) [55, Ch. 7]. In words, an SCM $\mathcal{M}^*$ is a representation of a system that includes a collection of mechanisms and a probability distribution over the exogenous conditions (to be formally defined later on). Second, any fully specified SCM $\mathcal{M}^*$ induces a collection of distributions known as the *Pearl Causal Hierarchy* (PCH) [5, Def. 9]. The importance of the PCH is that it formally delimits

Preprint. Under review.

distinct cognitive capabilities (also known as layers; not to be confused with neural nets layers) that can be associated with the human activities of "seeing" (layer 1), "doing" (2), and "imagining" (3) [56, Ch. 1]. [1] Each of these layers can be expressed as a distinct formal language and represents queries that can help to classify different types of inferences [5, Def. 8]. Together, these layers form a strict containment hierarchy [5, Thm. 1]. We illustrate these notions in Fig. 1(a) (left side), where SCM $\mathcal{M}^*$ induces layers L_1^*, L_2^*, L_3^* of the PCH.

Even though each possible statement within these capabilities has well-defined semantics given the true SCM $\mathcal{M}^*$ [55, Ch. 7], a challenging inferential task arises when one wishes to recover part of the PCH when $\mathcal{M}^*$ is only partially observed. This situation is typical in the real world aside from some special settings in physics and chemistry where the laws of nature are understood with high precision.

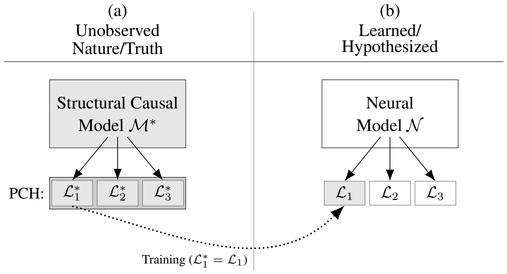


Figure 1: The l.h.s. contains the unobserved true SCM $\mathcal{M}^*$ that induces the three layers of the PCH. The r.h.s. contains an NCM that is trained to match in layer 1. The matching shading indicates that the two models agree w.r.t. L_1 while not necessarily agreeing w.r.t. layers 2 and 3.

For concreteness, consider the setting where one needs to make a statement about the effect of a new intervention (i.e., about layer 2), but only has observational data from layer 1, which is passively collected.[2] Going back to the causal-neural connection, one could try to learn a neural model $\mathcal{N}$ using the observational dataset (layer 1) generated by the true SCM $\mathcal{M}^*$, as illustrated in Fig. 1(b). Naturally, a basic consistency requirement is that $\mathcal{N}$ should be capable of generating the same distributions as $\mathcal{M}^*$; in this case, their layer 1 predictions should match (i.e., $L_1 = L_1^*$). Given the universality of neural models, it is not hard to believe that these constraints can be satisfied in the large sample limit. The question arises of whether the learned model $\mathcal{N}$ can act as a proxy, having the capability of predicting the effect of interventions that matches the L_2 distribution generated by the true (unobserved) SCM $\mathcal{M}^*$. [3] The answer to this question cannot be ascertained in general, as will become evident later on (Corol. 1). The intuitive reason behind this result is that there are multiple neural models that are equally consistent w.r.t. the L_1 distribution of $\mathcal{M}^*$ but generate different L_2-distributions. [4] Even though $\mathcal{N}$ may be expressive enough to fully represent $\mathcal{M}^*$ (as discussed later on), generating one particular parametrization of $\mathcal{N}$ consistent with L_1 is insufficient to provide any guarantee regarding higher-layer inferences, i.e., about predicting the effects of interventions (L_2) or counterfactuals (L_3).

The discussion above entails two tasks that have been acknowledged in the literature, namely, causal effect identification and estimation. The first – causal identification – has been extensively studied, and general solutions have been developed, such as Pearl's celebrated do-calculus [54]. Given the impossibility described above, the ingredient shared across current non-neural solutions is to represent assumptions about the unknown $\mathcal{M}^*$ in the form of causal diagrams [55, 62, 7] or their equivalence classes [28, 57, 29, 67]. The task is then to decide whether there is a unique solution for the causal query based on such assumptions. There are no neural methods today focused on solving this task.

The second task – causal estimation – is triggered when effects are determined to be identifiable by the first task. Whenever identifiability is obtained through the backdoor criterion/conditional ignorability [55, Sec. 3.3.1], deep learning techniques can be leveraged to estimate such effects with impressive practical performance [60, 49, 45, 30, 65, 66, 34, 61, 15, 25]. For effects that are identifiable through causal functionals that are not necessarily of the backdoor-form (e.g., frontdoor,

[1] This structure is named after Judea Pearl and is a central topic in his *Book of Why (BoW)*, where it is also called the "Ladder of Causation" [56]. For a more technical discussion on the PCH, we refer readers to [5].

[2] The full inferential challenge is, in practice, more general since an agent may be able to perform interventions and obtain samples from a subset of the PCH's layers, while its goal is to make inferences about some other parts of the layers [7, 43, 5]. This situation is not uncommon in RL settings [63, 17, 41, 42]. Still, for the sake of space and concreteness, we will focus on two canonical and more basic tasks found in the literature.

[3] We defer a more formal discussion on how neural models could be used to assess the effect of interventions to Sec. 2. Still, this is neither attainable in all universal neural architectures nor trivially implementable.

[4] Pearl shared a similar observation in the *BoW* [56, p. 32]: "Without the causal model, we could not go from rung (layer) one to rung (layer) two. This is why deep-learning systems (as long as they use only rung-one data and do not have a causal model) will never be able to answer questions about interventions (...)".

napkin), other optimization/statistical techniques can be employed that enjoy properties such as double robustness and debiasedness [31, 32, 33]. Each of these approaches optimizes a particular estimand corresponding to one specific target interventional distribution.

Despite all the great progress achieved so far, it is still largely unknown how to perform the tasks of causal identification and estimation in arbitrary settings using neural networks as a generative model, acting as a proxy for the true SCM $\mathcal{M}^*$. It is our goal here to develop a general causal-neural framework that has the potential to scale to real-world, high-dimensional domains while preserving the validity of its inferences, as in traditional symbolic approaches. In the same way that the causal diagram encodes the assumptions necessary for the do-calculus to decide whether a certain query is identifiable, our method encodes the same invariances as an inductive bias while being amenable to gradient-based optimization, allowing us to perform both tasks in an integrated fashion (in a way, addressing Pearl's concerns alluded to in Footnote 4). Specifically, our contributions are as follows:

1. [Sec. 2] We introduce a special yet simple type of SCM that is amenable to gradient descent called a *neural causal model* (NCM). We prove basic properties of this class of models, including its universal expressiveness and ability to encode an inductive bias representing certain structural invariances (Thm. 1-3). Notably, we show that despite the NCM's expressivity, it still abides by the Causal Hierarchy Theorem (Corol. 1).

2. [Sec. 3] We formalize the problem of neural identification (Def. 8) and prove a duality between identification in causal diagrams and in neural causal models (Thm. 4). We introduce an operational way to perform inferences in NCMs (Corol. 2-3) and a sound and complete algorithm to jointly train and decide effect identifiability for an NCM (Alg. 1, Corol. 4).

3. [Sec. 4] Building on these results, we develop a gradient descent algorithm to jointly identify and estimate causal effects (Alg. 2).

There are multiple ways of grounding these theoretical results. In Sec. 5, we perform experiments based on one possible implementation which support the feasibility of the proposed approach.

1.1 Preliminaries

In this section, we provide the necessary background to understand this work, following the presentation in [55]. An uppercase letter X indicates a random variable, and a lowercase letter x indicates its corresponding value; bold uppercase $\mathbf{X}$ denotes a set of random variables, and lowercase letter $\mathbf{x}$ its corresponding values. We use $\mathcal{D}_X$ to denote the domain of X and $\mathcal{D}_{\mathbf{X}} = \mathcal{D}_{X_1} \times \cdots \times \mathcal{D}_{X_k}$ for $\mathbf{X} = \{X_1, \dots, X_k\}$. We denote $P(\mathbf{X})$ as a probability distribution over a set of random variables $\mathbf{X}$ and $P(\mathbf{X} = \mathbf{x})$ as the probability of $\mathbf{X}$ being equal to the value of $\mathbf{x}$ under the distribution $P(\mathbf{X})$. For simplicity, we will mostly abbreviate $P(\mathbf{X} = \mathbf{x})$ as simply $P(\mathbf{x})$. The basic semantic framework of our analysis rests on *structural causal models* (SCMs) [55, Ch. 7], which are defined below.

Definition 1 (Structural Causal Model (SCM))**.** An SCM $\mathcal{M}$ is a 4-tuple $\langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$, where $\mathbf{U}$ is a set of exogenous variables (or "latents") that are determined by factors outside the model; $\mathbf{V}$ is a set $\{V_1, V_2, \dots, V_n\}$ of (endogenous) variables of interest that are determined by other variables in the model – that is, in $\mathbf{U} \cup \mathbf{V}$; $\mathcal{F}$ is a set of functions $\{f_{V_1}, f_{V_2}, \dots, f_{V_n}\}$ such that each f_i is a mapping from (the respective domains of) $\mathbf{U}_{V_i} \cup \mathbf{Pa}_{V_i}$ to V_i, where $\mathbf{U}_{V_i} \subseteq \mathbf{U}$, $\mathbf{Pa}_{V_i} \subseteq \mathbf{V} \setminus V_i$, and the entire set $\mathcal{F}$ forms a mapping from $\mathbf{U}$ to $\mathbf{V}$. That is, for $i = 1, \dots, n$, each $f_i \in \mathcal{F}$ is such that $v_i \leftarrow f_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}_{V_i})$; and $P(\mathbf{u})$ is a probability function defined over the domain of $\mathbf{U}$. ■

Each SCM $\mathcal{M}$ induces a causal diagram G where every $V_i \in \mathbf{V}$ is a vertex, there is a directed arrow $(V_j \rightarrow V_i)$ for every $V_i \in \mathbf{V}$ and $V_j \in Pa(V_i)$, and there is a dashed-bidirected arrow $(V_j \dashleftarrow\dashrightarrow V_i)$ for every pair $V_i, V_j \in \mathbf{V}$ such that $\mathbf{U}_{V_i}$ and $\mathbf{U}_{V_j}$ are not independent. For further details on this construction, see [5, Def. 13/16, Thm. 4]. The exogenous $\mathbf{U}_{V_i}$'s are not assumed independent (i.e. Markovianity does not hold). We consider here *recursive* SCMs, which implies acyclic diagrams.

We show next how an SCM $\mathcal{M}$ gives values to the PCH's layers; for details on the semantics, see [5, Sec. 1.2]. Superscripts are omitted when unambiguous.

Definition 2 (Layers 1, 2 Valuations)**.** An SCM $\mathcal{M}$ induces layer $L_2(\mathcal{M})$, a set of distributions over $\mathbf{V}$, one for each intervention $\mathbf{x}$. For each $\mathbf{Y} \subseteq \mathbf{V}$,

$$P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}}) = \sum_{\{\mathbf{u} \mid \mathbf{Y}_{\mathbf{x}}(\mathbf{u}) = \mathbf{y}\}} P(\mathbf{u}), \tag{1}$$

where $\mathbf{Y}_{\mathbf{x}}(\mathbf{u})$ is the solution for $\mathbf{Y}$ after evaluating $\mathcal{F}_{\mathbf{x}} := \{f_{V_i} : V_i \in \mathbf{V} \setminus \mathbf{X}\} \cup \{f_X \leftarrow x : X \in \mathbf{X}\}$. The specific distribution $P(\mathbf{V})$, where $\mathbf{X}$ is empty, is defined as layer $L_1(\mathcal{M})$. ■

In words, an external intervention forcing a set of variables $\mathbf{X}$ to take values $\mathbf{x}$ is modeled by replacing the original mechanism f_X for each $X \in \mathbf{X}$ with its corresponding value in $\mathbf{x}$. This operation is represented formally by the do-operator, $do(\mathbf{X} = \mathbf{x})$, and graphically as the *mutilation* procedure.

2 Neural Causal Models and the Causal Hierarchy Theorem

In this section, we aim to resolve the tension between expressiveness and learnability (Fig. 1). To that end, we define a special class of SCMs based on neural nets that is amenable to optimization and has the potential to act as a proxy for the true, unobserved SCM $\mathcal{M}^*$.

Definition 3 (NCM)**.** A Neural Causal Model (for short, NCM) $\widehat{M}(\boldsymbol{\theta})$ over variables $\mathbf{V}$ with parameters $\boldsymbol{\theta} = \{\theta_{V_i} : V_i \in \mathbf{V}\}$ is an SCM $\langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ such that

- $\widehat{\mathbf{U}} \subseteq \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \subseteq \mathbf{V}\}$, where each $\widehat{U}$ is associated with some subset of variables $\mathbf{C} \subseteq \mathbf{V}$, and $\mathcal{D}_{\widehat{U}} = [0, 1]$ for all $\widehat{U} \in \widehat{\mathbf{U}}$. (Unobserved confounding is present whenever $|\mathbf{C}| > 1$.)
- $\widehat{\mathcal{F}} = \{\hat{f}_{V_i} : V_i \in \mathbf{V}\}$, where each $\hat{f}_{V_i}$ is a feedforward neural network parameterized by $\theta_{V_i} \in \boldsymbol{\theta}$ mapping values of $\mathbf{U}_{V_i} \cup \mathbf{Pa}_{V_i}$ to values of V_i for some $\mathbf{Pa}_{V_i} \subseteq \mathbf{V}$ and $\mathbf{U}_{V_i} = \{\widehat{U}_{\mathbf{C}} : \widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}}, V_i \in \mathbf{C}\}$;
- $\widehat{P}(\widehat{\mathbf{U}})$ is defined s.t. $\widehat{U} \sim \text{Unif}(0, 1)$ for each $\widehat{U} \in \widehat{\mathbf{U}}$. ■

Some remarks are worth making at this point. First, by definition, all NCMs are SCMs, so they have the capability of generating any distribution associated with the PCH's layers. Second, not all SCMs are NCMs, since Def. 3 dictates that $\widehat{\mathbf{U}}$ follows uniform distributions in the unit interval and $\widehat{\mathcal{F}}$ are feedforward neural networks. [5] Further note that between any two variables V_i and V_j, $\mathbf{U}_{V_i}$ and $\mathbf{U}_{V_j}$ might share an input from $\widehat{\mathbf{U}}$, which will play a critical role in causality, not ruling out *a priori* the possibility of unobserved confounding (and violations of Markovianity). To compare the expressiveness of NCMs and SCMs, we formalize next the notion of consistency.

Definition 4 ($\text{P}^{(L_i)}$-Consistency)**.** Consider two SCMs, $\mathcal{M}_1$ and $\mathcal{M}_2$. $\mathcal{M}_2$ is said to be $\text{P}^{(L_i)}$-consistent (for short, L_i-consistent) w.r.t. $\mathcal{M}_1$ if $L_i(\mathcal{M}_1) = L_i(\mathcal{M}_2)$. ■

This definition applies to NCMs since they are also SCMs. As shown below, NCMs can not only approximate the collection of functions of the true SCM $\mathcal{M}^*$, but they can perfectly *represent* all the observational, interventional, and counterfactual distributions. This property is, in fact, special and not enjoyed by many neural models. (For examples and discussion, see Appendix C and D.1.)

Theorem 1 (NCM Expressiveness)**.** *For any SCM $\mathcal{M}^* = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$, there exists an NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ s.t. $\widehat{M}$ is L_3-consistent w.r.t. $\mathcal{M}^*$.* ■

Due to space constraints, proofs are provided in Appendix A. Thm. 1 ascertains that there is no loss of expressive power using NCMs despite the constraints imposed over its form, i.e., NCMs are as expressive as SCMs. One might be tempted to surmise, therefore, that an NCM can be trained on the observed data and act as a proxy for the true SCM $\mathcal{M}^*$, and inferences about other quantities of $\mathcal{M}^*$ can be done through computation directly in $\widehat{\mathcal{M}}$. Unfortunately, this is almost never the case: [6]

Corollary 1 (Neural Causal Hierarchy Theorem (N-CHT))**.** *Let Ω^* and Ω be the sets of all SCMs and NCMs, respectively. We say that Layer j of the causal hierarchy for NCMs* collapses *to Layer i ($i < j$) relative to $\mathcal{M}^* \in \Omega^*$ if $L_i(\mathcal{M}^*) = L_i(\widehat{M})$ implies that $L_j(\mathcal{M}^*) = L_j(\widehat{M})$ for all $\widehat{M} \in \Omega$. Then, with respect to the Lebesgue measure over (a suitable encoding of L_3-equivalence classes of) SCMs, the subset in which Layer j of NCMs collapses to Layer i has measure zero.* ■

This corollary highlights the fundamental challenge of performing inferences across the PCH layers even when the target object (NCM $\widehat{\mathcal{M}}$) is a suitable surrogate for the underlying SCM $\mathcal{M}^*$, in terms of expressiveness and capability of generating the same observed distribution. That is, expressiveness does not mean that the learned object has the same empirical content as the generating model.

[5] We note that feedforward networks are universal approximators [14, 26] (see also [19]), and any probability distribution can be generated by the uniform one (e.g., see *probability integral transform* [1]). This suggests that the pair $\langle \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ may be expressive enough for modeling $\mathcal{M}^*$'s mechanisms $\mathcal{F}$ and distribution $P(\mathbf{U})$ w.l.o.g.. These particular modeling choices were made for the sake of explanation, and the results discussed here still hold for other, more arbitrary classes of functions and probability distributions, as shown in Appendix D.

[6] Multiple examples of this phenomenon are discussed in Appendix C.1 and [5, Sec. 1.2]

2.1 A Family of Neural-Interventional Constraints (Inductive Bias)

In this section, we investigate constraints about $\mathcal{M}^*$ that will narrow down the hypothesis space and possibly allow for valid cross-layer inferences. One well-studied family of structural constraints comes in the form of a pair comprised of a collection of interventional distributions $\mathcal{P}$ and causal diagram $\mathcal{G}$, known as a *causal bayesian network* (CBN) (Def. 15; see also [5, Thm. 4])). The diagram $\mathcal{G}$ encodes constraints over the space of interventional distributions $\mathcal{P}$ which are useful to perform cross-layer inferences (for details, see Appendix C.2). For simplicity, we focus on interventional inferences from observational data. To compare the constraints entailed by distinct SCMs, we define the following notion of consistency:

Definition 5 ($\mathcal{G}$-Consistency)**.** Let $\mathcal{G}$ be the causal diagram induced by SCM $\mathcal{M}^*$. For any SCM $\mathcal{M}$, we say that $\mathcal{M}$ is $\mathcal{G}$-consistent (w.r.t. $\mathcal{M}^*$) if $\mathcal{G}$ is a CBN for $L_2(\mathcal{M})$. ■

In the context of NCMs, this means that $\mathcal{M}$ would impose the same constraints over $\mathcal{P}$ as the true SCM $\mathcal{M}^*$ (since $\mathcal{G}$ is also a CBN for $L_2(\mathcal{M}^*)$ by [5, Thm. 4]). Whenever the corresponding diagram $\mathcal{G}$ is known, one should only consider NCMs that are $\mathcal{G}$-consistent. [7] We provide below a systematic way of constructing $\mathcal{G}$-consistent NCMs.

Definition 6 (C^2-Component)**.** For a causal diagram $\mathcal{G}$, a subset $\mathbf{C} \subseteq \mathbf{V}$ is a complete confounded component (for short, C^2-component) if any pair $V_i, V_j \in \mathbf{C}$ is connected with a bidirected arrow in $\mathcal{G}$ and is maximal (i.e. there is no C^2-component $\mathbf{C}'$ for which $\mathbf{C} \subset \mathbf{C}'$.) ■

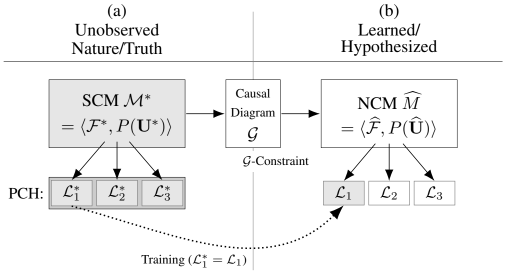


Figure 2: The l.h.s. contains the true SCM $\mathcal{M}^*$ that induces PCH's three layers. The r.h.s. contains an NCM that is trained with layer 1 data. The matching shading indicates that the two models agree with respect to L_1 while not necessarily agreeing in layers 2 and 3. The causal diagram $\mathcal{G}$ entailed by $\mathcal{M}^*$ is used as an inductive bias for $\widehat{M}$.

Definition 7 ($\mathcal{G}$-Constrained NCM (constructive))**.** Let $\mathcal{G}$ be the causal diagram induced by SCM $\mathcal{M}^*$. Construct NCM $\widehat{M}$ as follows. **(1)** Choose $\widehat{\mathbf{U}}$ s.t. $\widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}}$ if and only if $\mathbf{C}$ is a C^2-component in $\mathcal{G}$. **(2)** For each variable $V_i \in \mathbf{V}$, choose $\mathbf{Pa}_{V_i} \subseteq \mathbf{V}$ s.t. for every $V_j \in \mathbf{V}$, $V_j \in \mathbf{Pa}_{V_i}$ if and only if there is a directed edge from V_j to V_i in $\mathcal{G}$. Any NCM in this family is said to be $\mathcal{G}$-constrained. ■

Note that this represents a family of NCMs, not a unique one, since $\boldsymbol{\theta}$ (the parameters of the neural networks) are not yet specified by the construction, only the scope of the function and independence relations among the sources of randomness ($\widehat{\mathbf{U}}$). In contrast to SCMs where both $\langle \mathcal{F}, P(\mathbf{u}) \rangle$ can freely vary, the degrees of freedom within NCMs come from $\boldsymbol{\theta}$. [8]

We show next that an NCM constructed following the procedure dictated by Def. 7 encodes all the constraints of the original causal diagram.

Theorem 2 (NCM $\mathcal{G}$-Consistency)**.** *Any $\mathcal{G}$-constrained NCM $\widehat{M}(\boldsymbol{\theta})$ is $\mathcal{G}$-consistent.* ■

We show next the implications of imposing the structural constraints embedded in the causal diagram.

Theorem 3 (L_2-$\mathcal{G}$ Representation)**.** *For any SCM $\mathcal{M}^*$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-constrained NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ that is L_2-consistent w.r.t. $\mathcal{M}^*$.* ■

The importance of this result stems from the fact that despite constraining the space of NCMs to those compatible with $\mathcal{G}$, the resultant family is still expressive enough to represent the entire Layer 2 of the original, unobserved SCM $\mathcal{M}^*$.

[7] Otherwise, the causal diagram can be learned through structural learning algorithms from observational data [62, 58] or experimental data [39, 38, 27]. See the next footnote for a neural take on this task.

[8] There are some works in the literature that model SCMs using neural networks as functions, but which differ in nature and scope to our work. [21] attempts to learn the entire SCM from observational data under the Markov assumptions. This entails strong constraints over $P(U)$, which in the context of identification means all effects are always identifiable; see Corol. 3. [8, 10] also use experimental (L_2) data to learn the causal diagram $\mathcal{G}$.

Fig. 2 provides a mental picture useful to understand the results discussed so far. The true SCM $\mathcal{M}^*$ generates the three layers of the causal hierarchy (left side), but in many settings only observational data (layer 1) is visible. An NCM $\widehat{M}$ trained with this data is capable of perfectly representing L_1 (right side). For almost any generating $\mathcal{M}^*$ sampled from the space Ω^*, there exists an NCM $\widehat{M}$ that exhibits the same behavior with respect to observational data ($\widehat{M}$ is L_1-consistent) but exhibits a different behavior with respect to interventional data. In other words, L_1 underdetermines L_2. (Similarly, L_1 and L_2 underdetermine L_3 [5, Sec. 1.3].) Still, the true SCM $\mathcal{M}^*$ also induces a causal diagram $\mathcal{G}$ that encodes constraints over the interventional distributions. If we use this collection of constraints as an inductive bias, imposing G-consistency in the construction of the NCM, $\widehat{M}$ may agree with those of the true $\mathcal{M}^*$ under some conditions, which we will investigate in the next section.

3 The Neural Identification Problem

We now investigate the feasibility of causal inferences in the class of $\mathcal{G}$-constrained NCMs. [9] The first step is to refine the notion of identification [55, pp. 67] to inferences within this class of models.

Definition 8 (Neural Effect Identification)**.** Let Ω^* be the set of all SCMs, and Ω the set of NCMs. The causal effect $P(\mathbf{y} \mid do(\mathbf{x}))$ is said to be neural identifiable from the set of $\mathcal{G}$-constrained NCMs, $\Omega(\mathcal{G})$, and observational data $P(\mathbf{v})$ if and only if $P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x})) = P^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$ for every pair of models $\mathcal{M}^* \in \Omega^*$ and $\widehat{M} \in \Omega(\mathcal{G})$ s.t. $\mathcal{M}^*$ induces $\mathcal{G}$ and $P(\mathbf{v}) = P^{\mathcal{M}^*}(\mathbf{v}) = P^{\widehat{M}}(\mathbf{v})$. ■

In the context of graphical identifiability [55, Def. 3.2.4] and do-calculus, an effect is identifiable if any SCM in Ω^* compatible with the observed causal diagram and capable of generating the observational distribution matches the interventional query. If we constrain our attention to NCMs, identification in the general class would imply identification in NCMs, naturally, since it needs to hold for all SCMs. On the other hand, it would be insufficient to constrain identification within the NCM class since it is conceivable that the effect could match within the class (perhaps in a not very expressive neural architecture) while there still exists an SCM that generates the same observational distribution and induces the same diagram, but does not agree in the interventional query; see Example 7 in Appendix C. Accordingly, Def. 8 relates the solution space of these two classes of models and requires the solution within the neural class to match the solution within the SCM class; for an illustration, see Fig. 3. The next result makes the relationship between these classes more explicit.

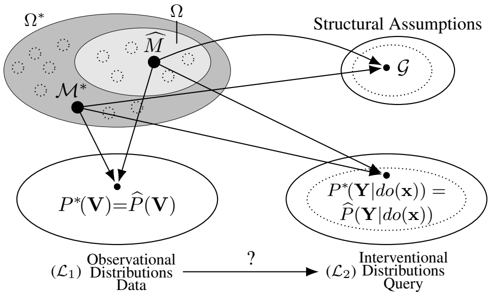

Ω^*
Ω
$\widehat{M}$
$\mathcal{M}^*$
Structural Assumptions
$\mathcal{G}$
$P^*(\mathbf{V})=\widehat{P}(\mathbf{V})$
$P^*(\mathbf{Y}|do(\mathbf{x})) = \widehat{P}(\mathbf{Y}|do(\mathbf{x}))$
($\mathcal{L}_1$) Observational Distributions Data
?
($\mathcal{L}_2$) Interventional Distributions Query

Figure 3: $P(\mathbf{Y} \mid do(\mathbf{x}))$ is identifiable from $P(\mathbf{V})$ and NCM $\widehat{\mathcal{M}} \in \Omega$ if for any SCM $\mathcal{M}^* \in \Omega^*$ (top left), $\widehat{\mathcal{M}}, \mathcal{M}^*$ match in $P(\mathbf{V})$ (bottom left) and $\mathcal{G}$ (top right), they also match in $P(\mathbf{Y} \mid do(\mathbf{x}))$ (bottom right).

Theorem 4 (Graphical-Neural Equivalence (Dual ID))**.** *Let Ω^* be the set of all SCMs and Ω the set of NCMs. Consider the true SCM $\mathcal{M}^*$ and the corresponding causal diagram $\mathcal{G}$. Let $Q = P(\mathbf{y} \mid do(\mathbf{x}))$ be the query of interest and $P(\mathbf{v})$ the observational distribution. Then, Q is neural identifiable from $\Omega(\mathcal{G})$ and $P(\mathbf{v})$ if and only if it is identifiable from $\mathcal{G}$ and $P(\mathbf{v})$.* ■

Theorem 4 says that the identification status of a query is preserved across settings. For instance, if an effect is identifiable from the combination of a causal graph $\mathcal{G}$ and $P(\mathbf{v})$, it will also be identifiable from $\mathcal{G}$-constrained NCM (and the other way around). This is encouraging since our goal is to perform inferences directly through neural causal models, avoiding the symbolic nature of do-calculus computation; the theorem guarantees that this is achievable *in principle*.

Corollary 2 (Neural Mutilation (Operational ID))**.** *Consider the true SCM $\mathcal{M}^* \in \Omega^*$, causal diagram $\mathcal{G}$, the observational distribution $P(\mathbf{v})$, and a target query Q equal to $P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$. Let $\widehat{\mathcal{M}} \in \Omega(\mathcal{G})$ be a $\mathcal{G}$-constrained NCM that is L_1-consistent with $\mathcal{M}^*$. If the effect is identifiable from*

[9] This is akin to what happens with the non-neural CHT [5, Thm. 1] and the subsequent use of causal diagrams to encode the necessary inductive bias, and in which the do-calculus allows for cross-layer inferences directly from the graphical representation [5, Sec. 1.4].

$\mathcal{G}$ and $P(\mathbf{v})$, then Q is computable through a mutilation process on a proxy NCM $\widehat{\mathcal{M}}$, i.e., for each $X \in \mathbf{X}$, replacing the equation f_x with a constant x $(Q = \text{PROC-MUTILATION}(\widehat{M}; \mathbf{X}, \mathbf{Y}))$. ■

Following the duality stated by Thm. 4, this result provides a practical, operational way of evaluating queries in NCMs: inferences may be carried out through the process of mutilation, which gives semantics to queries in the generating SCM $\mathcal{M}^*$ (via Def. 2). What is interesting here is that the proposition provides conditions under which this process leads to valid inferences, even when $\mathcal{M}^*$ is unknown, or when the mechanisms $\mathcal{F}$ and unobserved noise $P(\mathbf{U})$ of $\mathcal{M}^*$ and the proxy NCM $\widehat{M}$ do not match (for concreteness, refer to example 5 in Appendix. C). Inferences using mutilation on $\widehat{M}$ would work as if they were on $\mathcal{M}^*$ itself, and they would be correct so long as certain stringent properties were satisfied – L_1-consistency, G-constraint, and identifiability. As shown earlier, if these properties are not satisfied, inferences within a proxy model will almost never be valid, likely bearing no relationship with the ground truth (see examples 2, 3, or 4 in Appendix C).

Still, one special class of SCMs in which any interventional distribution is identifiable is called *Markovian*, where all U_i are assumed independent and affect only one endogenous variable V_i.

Corollary 3 (Markovian Identification)**.** *Whenever the $\mathcal{G}$-constrained NCM $\widehat{\mathcal{M}}$ is Markovian, $P(\mathbf{y} \mid do(\mathbf{x}))$ is always identifiable through the process of mutilation in the proxy NCM (via Corol. 2).* ■

This is obviously not the case for general non-Markovian models, which leads to the very problem of identification. In these cases, we need to decide whether the mutilation procedure (Corol. 2) can, in principle, produce the correct answer. We show in Alg. 1 a learning procedure that decides whether a certain effect is identifiable from observational data. Remarkably, the procedure is both necessary and sufficient, which means that all, and only, identifiable effects are classified as such by our procedure. This implies that, theoretically, deep learning could be as powerful as the do-calculus in deciding identifiability. (For a more nuanced discussion of symbolic versus optimization-based approaches for identification, see Appendix C.4. For non-identifiability examples and discussion, see C.3.)

Algorithm 1: Identifying/estimating queries with NCMs.

```
Input  : causal query $Q = P(\mathbf{y} \mid do(\mathbf{x}))$, $L_1$ data $P(\mathbf{v})$, and
         causal diagram $\mathcal{G}$
Output : $P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$ if identifiable, FAIL otherwise.

$\widehat{M} \leftarrow$ NCM($\mathbf{V}$, $\mathcal{G}$)                         // from Def. 7
$\boldsymbol{\theta}^*_{\min} \leftarrow \arg\min_{\boldsymbol{\theta}} P^{\widehat{M}(\boldsymbol{\theta})}(\mathbf{y} \mid do(\mathbf{x}))$ s.t. $L_1(\widehat{M}(\boldsymbol{\theta})) = P(\mathbf{v})$
$\boldsymbol{\theta}^*_{\max} \leftarrow \arg\max_{\boldsymbol{\theta}} P^{\widehat{M}(\boldsymbol{\theta})}(\mathbf{y} \mid do(\mathbf{x}))$ s.t. $L_1(\widehat{M}(\boldsymbol{\theta})) = P(\mathbf{v})$
if $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y} \mid do(\mathbf{x})) \neq P^{\widehat{M}(\boldsymbol{\theta}^*_{\max})}(\mathbf{y} \mid do(\mathbf{x}))$ then
    return FAIL
else
    return $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y} \mid do(\mathbf{x}))$      // choose min or max
      arbitrarily
```

Corollary 4 (Soundness and Completeness)**.** *Let Ω^* be the set of all SCMs, $\mathcal{M}^* \in \Omega^*$ be the true SCM inducing causal diagram $\mathcal{G}$, $Q = P(\mathbf{y} \mid do(\mathbf{x}))$ be a query of interest, and $\widehat{Q}$ be the result from running Alg. 1 with inputs $P^*(\mathbf{v}) = L_1(\mathcal{M}^*) > 0$, $\mathcal{G}$, and Q. Then Q is identifiable from $\mathcal{G}$ and $P^*(\mathbf{v})$ if and only if $\widehat{Q}$ is not `FAIL`. Moreover, if $\widehat{Q}$ is not `FAIL`, then $\widehat{Q} = P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$.* ■

4 The Neural Estimation Problem

While identifiability is fully solved by the asymptotic theory discussed so far (i.e., it is both necessary and sufficient), we now consider the problem of estimating causal effects in practice under imperfect optimization and finite samples and computation. For concreteness, we discuss next the discrete case with binary variables, but our construction extends naturally to categorical and continuous variables (see Appendix B). We propose next a construction of a $\mathcal{G}$-constrained NCM $\widehat{M}(\mathcal{G}; \boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$, which is a possible instantiation of Def. 7:

$$
\begin{cases}
\mathbf{V} & := \mathbf{V}, \ \widehat{\mathbf{U}} := \{U_{\mathbf{C}} : \mathbf{C} \in C^2(\mathcal{G})\} \cup \{G_{V_i} : V_i \in \mathbf{V}\}, \\
\widehat{\mathcal{F}} & := \left\{ f_{V_i} := \arg\max_{j \in \{0,1\}} g_{j,V_i} + \begin{cases} \log \sigma(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i})) & j = 1 \\ \log(1 - \sigma(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i}))) & j = 0 \end{cases} \right\}, \\
P(\widehat{\mathbf{U}}) & := \{U_{\mathbf{C}} \sim \text{Unif}(0,1) : U_{\mathbf{C}} \in \mathbf{U}\} \cup \\
& \quad \ \{G_{j,V_i} \sim \text{Gumbel}(0,1) : V_i \in \mathbf{V}, j \in \{0,1\}\},
\end{cases}
\tag{2}
$$

where $\mathbf{V}$ are the nodes of $\mathcal{G}$; $\sigma : \mathbb{R} \to (0,1)$ is the sigmoid activation function; $C^2(\mathcal{G})$ is the set of C^2-components of $\mathcal{G}$; each G_{j,V_i} is a standard Gumbel random variable [24]; each $\phi_{V_i}(\cdot; \theta_{V_i})$ is a neural net parameterized by $\theta_{V_i} \in \boldsymbol{\theta}$; $\mathbf{pa}_{V_i}$ are the values of the parents of V_i; and $\mathbf{u}^c_{V_i}$ are the values of $\mathbf{U}^c_{V_i} := \{U_\mathbf{C} : U_\mathbf{C} \in \mathbf{U} \text{ s.t. } V_i \in \mathbf{C}\}$. The parameters $\boldsymbol{\theta}$ are not yet specified and must be learned through training to enforce L_1-consistency (Def. 4).

Let $\mathbf{U}^c$ and $\mathbf{G}$ denote the latent C^2-component variables and Gumbel random variables, respectively. To estimate $P^{\widehat{M}}(\mathbf{v})$ and $P^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$ given Eq. 2, we may compute the probability mass of a datapoint $\mathbf{v}$ with intervention $do(\mathbf{X} = \mathbf{x})$ ($\mathbf{X}$ is empty when observational) as:

$$P^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v} \mid do(\mathbf{x})) = \mathop{\mathbb{E}}_{P(\mathbf{u}^c)}\left[\prod_{V_i \in \mathbf{V}\setminus\mathbf{X}} \tilde{\sigma}_{v_i}\right] \approx \frac{1}{m}\sum_{j=1}^{m} \prod_{V_i \in \mathbf{V}\setminus\mathbf{X}} \tilde{\sigma}_{v_i}, \tag{3}$$

where $\tilde{\sigma}_{v_i} := \begin{cases} \sigma(\phi_i(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i})) & v_i = 1 \\ 1 - \sigma(\phi_i(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i})) & v_i = 0 \end{cases}$ and $\{\mathbf{u}^c_j\}_{j=1}^m$ are samples from $P(\mathbf{U}^c)$. Here, we assume $\mathbf{v}$ is consistent with $\mathbf{x}$ (the values of $X \in \mathbf{X}$ in $\mathbf{v}$ match the corresponding ones of $\mathbf{x}$). Otherwise, $P^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v} \mid do(\mathbf{x})) = 0$. For numerical stability of each $\phi_i(\cdot)$, we work in log-space and use the log-sum-exp trick.

Alg. 1 (lines 2-3) requires non-trivial evaluations of expressions like $\arg\max_{\boldsymbol{\theta}} P^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$ while enforcing L_1-consistency. Whenever only finite samples are available $\{\mathbf{v}_k\}_{k=1}^n \sim P^*(\mathbf{V})$, the parameters of an L_1-consistent NCM may be estimated by minimizing data negative log-likelihood:

$$\begin{aligned} \boldsymbol{\theta} &\in \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{P^*(\mathbf{v})}\left[-\log P^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v})\right] \\ &\approx \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{k=1}^{n} -\log \widehat{P}_m^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v}_k). \end{aligned} \tag{4}$$

To simultaneously maximize $P^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$, we subtract a weighted second term $\log \widehat{P}_m^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x}))$, resulting in the objective $\mathcal{L}(\{\mathbf{v}_k\}_{k=1}^n)$ equal to

$$\frac{1}{n}\sum_{k=1}^{n} -\log \widehat{P}_m^{\widehat{M}}(\mathbf{v}_k) - \lambda \log \widehat{P}_m^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x})), \tag{5}$$

where λ is initially set to a high value and decreases during training. To minimize, we instead subtract $\lambda \log(1 - \widehat{P}_m^{\widehat{M}}(\mathbf{y} \mid do(\mathbf{x})))$ from the log-likelihood.

Algorithm 2: Training Model

```
Input : Data $\{\mathbf{v}_k\}_{k=1}^n$, variables $\mathbf{V}, \mathbf{X} \subseteq \mathbf{V}$,
        $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}, \mathbf{Y} \subseteq \mathbf{V}, \mathbf{y} \in \mathcal{D}_{\mathbf{Y}}$, causal diagram $\mathcal{G}$,
        number of Monte Carlo samples $m$, regularization
        constant $\lambda$, learning rate $\eta$
$\widehat{M} \leftarrow$ NCM($\mathbf{V}, \mathcal{G}$)                // from Def. 7
Initialize parameters $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$
for $k \leftarrow 1$ to $n$ do
    // Estimate from Eq. 3
    $\hat{p}_{\min} \leftarrow$ Estimate($\widehat{M}(\boldsymbol{\theta}_{\min}), \mathbf{V}, \mathbf{v}_k, \emptyset, \emptyset, m$)
    $\hat{p}_{\max} \leftarrow$ Estimate($\widehat{M}(\boldsymbol{\theta}_{\max}), \mathbf{V}, \mathbf{v}_k, \emptyset, \emptyset, m$)
    $\hat{q}_{\min} \leftarrow 0$
    $\hat{q}_{\max} \leftarrow 0$
    for $\mathbf{v} \in \mathcal{D}_{\mathbf{V}}$ do
        if Consistent($\mathbf{v}, \mathbf{y}$) then
            $\hat{q}_{\min} \leftarrow \hat{q}_{\min} +$
              Estimate($\widehat{M}(\boldsymbol{\theta}_{\min}), \mathbf{V}, \mathbf{v}, \mathbf{X}, \mathbf{x}, m$)
            $\hat{q}_{\max} \leftarrow \hat{q}_{\max} +$
              Estimate($\widehat{M}(\boldsymbol{\theta}_{\max}), \mathbf{V}, \mathbf{v}, \mathbf{X}, \mathbf{x}, m$)
    // $\mathcal{L}$ from Eq. 5
    $\mathcal{L}_{\min} \leftarrow -\log \hat{p}_{\min} - \lambda \log(1 - \hat{q}_{\min})$
    $\mathcal{L}_{\max} \leftarrow -\log \hat{p}_{\max} - \lambda \log \hat{q}_{\max}$
    $\boldsymbol{\theta}_{\min} \leftarrow \boldsymbol{\theta}_{\min} + \eta\nabla\mathcal{L}_{\min}$
    $\boldsymbol{\theta}_{\max} \leftarrow \boldsymbol{\theta}_{\max} + \eta\nabla\mathcal{L}_{\max}$
```

Alg. 2 is one possible way of optimizing the parameters $\boldsymbol{\theta}$ required in lines 2,3 of Alg. 1. Eq. 5 is amenable to optimization through standard gradient descent tools, e.g., [36, 48, 47]. [10] [11]

One way of understanding Alg. 1 is as a search within the $\Omega(\mathcal{G})$ space for two NCM parameterizations, $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$, that minimizes/maximizes the interventional distribution, respectively. Whenever the optimization ends, we can compare the corresponding $P(\mathbf{y} \mid do(\mathbf{x}))$ and determine whether an effect is identifiable. With perfect optimization and unbounded resources, identifiability entails the equality between these two quantities. In practice, we rely on a hypothesis testing step such as

$$|f(\widehat{M}(\boldsymbol{\theta}_{\max})) - f(\widehat{M}(\boldsymbol{\theta}_{\min}))| < \tau \tag{6}$$

for quantity of interest f and a certain threshold τ. This threshold is somewhat similar to a significance level in statistics and can be used to control certain types of errors. In our case, the threshold τ can be determined empirically. For further discussion, see Appendix B.

[10] Our approach is flexible and may take advantage of these different methods depending on the context. There are a number of alternatives for minimizing the discrepancy between P^* and $P^{\widehat{M}}$, including minimizing divergences, such as maximum mean discrepancy [23] or kernelized Stein discrepancy [46], performing variational inference [9], or generative adversarial optimization [20].

[11] The NCM can be extended to the continuous case by replacing the Gumbel-max trick on $\sigma(\phi_i(\cdot))$ with a model that directly computes a probability density given a data point, e.g., normalizing flow [59] or VAE [37].

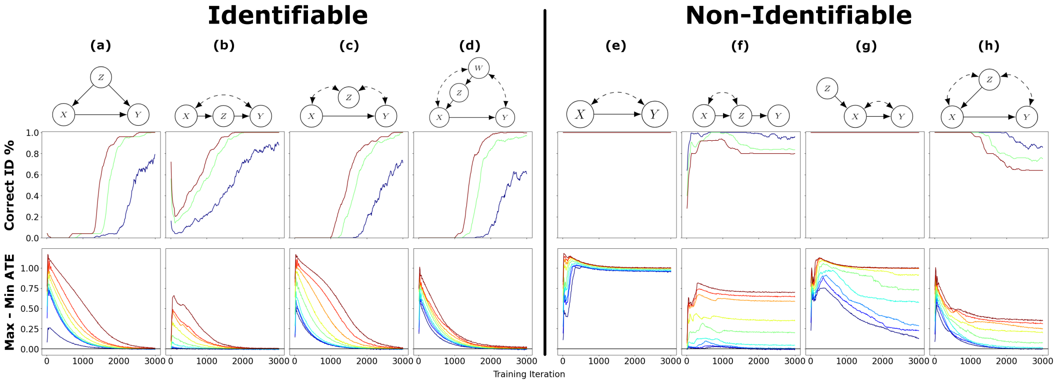


Figure 4: Experimental results on deciding identifiability with NCMs. **Top**: Graphs from left to right: (ID cases) back-door, front-door, M, napkin; (not ID cases) bow, extended bow, IV, bad M. **Middle**: Classification accuracy over 3,000 training epochs from running hypothesis test on Eq. 6 with $\tau = 0.01$ (blue), 0.03 (green), 0.05 (red). **Bottom**: (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentiles for max-min gaps over 3000 training epochs.

5 Experiments

We start by evaluating NCMs (following Eq. 2) in their ability to decide whether an effect is identifiable through Alg. 2. Observational data is generated from 8 different SCMs, and their corresponding causal diagrams are shown in Fig. 4 (top part), and Appendix B provides further details of the parametrizations. Since the NCM does not have access to the true SCM, the causal diagram and generated datasets are passed to the algorithm to decide whether an effect is identifiable. The target effect is $P(Y \mid do(X))$, and the quantity we optimize is the *average treatment effect* (ATE) of X on Y, $ATE_{\mathcal{M}}(X,Y) = \mathbb{E}_{\mathcal{M}}[Y \mid do(X=1)] - \mathbb{E}_{\mathcal{M}}[Y \mid do(X=0)]$. Note that if the outcome Y is binary, as in our examples, $\mathbb{E}[Y \mid do(X=x)] = P(Y=1|do(X=x))$. The effect is identifiable through do-calculus in the settings represented by Fig. 4 in the left part, and not identifiable in right.

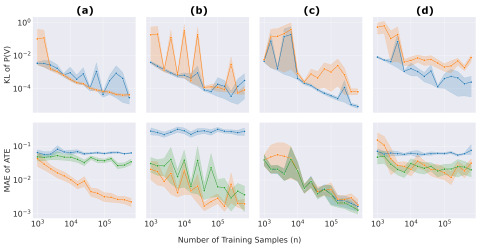


Figure 5: NCM estimation results for ID cases. Columns a, b, c, d correspond to the same graphs as a, b, c, d in Fig. 4. **Top**: KL divergence of $P(\mathbf{V})$ induced by naïve model (blue) and NCM (orange) compared to $P^{\mathcal{M}^*}(\mathbf{V})$. **Bottom**: MAE of ATE of naïve model (blue), NCM (orange), and WERM (green). Plots in log-log scale.

The bottom row of Fig. 4 shows the *max-min gaps*, the l.h.s of Eq. 6 with $f(\mathcal{M}) = \text{ATE}_{\mathcal{M}}(X,Y)$, over 3000 training epochs. The parameter λ is set to 1 at the beginning, and decreases logarithmically over each epoch until it reaches 0.001 at the end of training. The max-min gaps can be used to classify the quantity as "ID" or "non-ID" using the hypothesis testing procedure described in Appendix B. The classification accuracies per training epoch are shown in Fig. 4 (middle row). Note that in identifiable settings, the gaps slowly reduce to 0, while the gaps rapidly grow and stay high throughout training in the unidentifiable ones. The classification accuracy for ID cases then gradually increases as training progresses, while accuracy for non-ID cases remain high the entire time (perfect in the bow and IV cases).

In the identifiable settings, we also evaluate the performance of the NCM at estimating the correct causal effect, as shown in Fig. 5. As a generative model, the NCM is capable of generating samples from both $P(\mathbf{V})$ and identifiable L_2 distributions like $P(Y \mid do(X))$. We compare the NCM to a naïve generative model trained via likelihood maximization fitted on $P(\mathbf{V})$ without using the inductive bias of the NCM. Since the naïve model is not defined to sample from $P(y \mid do(x))$, this shows the implications of arbitrarily choosing $P(y \mid do(x)) = P(y \mid x)$. Both models improve at fitting $P(\mathbf{V})$ with more samples, but the naïve model fails to learn the correct ATE except in case (c), where $P(y \mid do(x)) = P(y \mid x)$. Further, the NCM is competitive with WERM [32], a state-of-the-art estimation method that directly targets estimating the causal effect without generating samples.

6 Conclusions

In this paper, we introduced neural causal models (NCMs) (Def. 3, 18), a special class of SCMs trainable through gradient-based optimization techniques. We showed that despite being as expressive as SCMs (Thm. 1), NCMs are unable to perform cross-layer inferences in general (Corol. 1). Disentangling expressivity and learnability, we formalized a new type of inductive bias based on non-parametric, structural properties of the generating SCM, accompanied with a constructive procedure that allows NCMs to represent constraints over the space of interventional distributions akin to causal diagrams (Thm. 2). We showed that NCMs with this bias retain their full expressivity (Thm. 3) but are now empowered to solve canonical tasks in causal inference, including the problems of identification and estimation (Thm. 4). We grounded these results by providing a training procedure that is both sound and complete (Alg. 1, 2, Cor. 4). Practically speaking, different neural implementations – combination of architectures, training algorithms, loss functions – can leverage the framework results introduced in this work (Appendix D.1). We implemented one of such alternatives as a proof of concept, and experimental results support the feasibility of the proposed approach. After all, we hope the causal-neural framework established in this paper can help develop more principled and robust architectures to empower the next generation of AI systems. We expect these systems to combine the best of both worlds by (1) leveraging causal inference capabilities of processing the structural invariances found in nature to construct more explainable and generalizable decision-making procedures, and (2) leveraging deep learning capabilities to scale inferences to handle challenging, high dimensional settings found in practice.

Acknowledgements

We thank Judea Pearl, Richard Zemel, Yotam Alexander, Juan Correa, Sanghack Lee, and Junzhe Zhang for their valuable feedback. Kevin Xia and Elias Bareinboim were supported in part by funding from the NSF, Amazon, JP Morgan, and The Alfred P. Sloan Foundation. Yoshua Bengio was supported in part by funding from CIFAR, NSERC, Samsung, and Microsoft.

References

[1] Angus, J. E. (1994). The probability integral transform and related results. *SIAM Review*, 36(4):652–654.

[2] Appel, L. J., Moore, T. J., Obarzanek, E., Vollmer, W. M., Svetkey, L. P., Sacks, F. M., Bray, G. A., Vogt, T. M., Cutler, J. A., Windhauser, M. M., and et al. (1997). A clinical trial of the effects of dietary patterns on blood pressure. *New England Journal of Medicine*, 336(16):1117–1124.

[3] Balke, A. and Pearl, J. (1994). Counterfactual Probabilities: Computational Methods, Bounds, and Applications. In de Mantaras, R. L. and D.˜Poole, editors, *Uncertainty in Artificial Intelligence 10*, pages 46–54. Morgan Kaufmann, San Mateo, CA.

[4] Bareinboim, E., Brito, C., and Pearl, J. (2012). Local Characterizations of Causal Bayesian Networks. In Croitoru, M., Rudolph, S., Wilson, N., Howse, J., and Corby, O., editors, *Graph Structures for Knowledge Representation and Reasoning*, pages 1–17, Berlin, Heidelberg. Springer Berlin Heidelberg.

[5] Bareinboim, E., Correa, J. D., Ibeling, D., and Icard, T. (2020). On Pearl's Hierarchy and the Foundations of Causal Inference. Technical Report R-60, Causal AI Lab, Columbia University, Also, In "Probabilistic and Causal Inference: The Works of Judea Pearl" (ACM Turing Series), in press.

[6] Bareinboim, E., Forney, A., and Pearl, J. (2015). Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems*, pages 1342–1350.

[7] Bareinboim, E. and Pearl, J. (2016). Causal inference and the data-fusion problem. In Shiffrin, R. M., editor, *Proceedings of the National Academy of Sciences*, volume 113, pages 7345–7352. National Academy of Sciences.

[8] Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. (2020). A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[9] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[10] Brouillard, P., Lachapelle, S., Lacoste, A., Lacoste-Julien, S., and Drouin, A. (2020). Differentiable causal discovery from interventional data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21865–21877. Curran Associates, Inc.

[11] Casella, G. and Berger, R. (2001). *Statistical Inference*, pages 54–55. Duxbury Resource Center.

[12] Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

[13] Correa, J. and Bareinboim, E. (2020). General transportability of soft interventions: Completeness results. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10902–10912, Vancouver, Canada. Curran Associates, Inc.

[14] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

[15] Du, X., Sun, L., Duivesteijn, W., Nikolaev, A., and Pechenizkiy, M. (2021). Adversarial balancing-based representation learning for causal effect inference with observational data. *Data Mining and Knowledge Discovery*.

[16] Falcon, W. and Cho, K. (2020). A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*.

[17] Forney, A., Pearl, J., and Bareinboim, E. (2017). Counterfactual Data-Fusion for Online Reinforcement Learners. In *Proceedings of the 34th International Conference on Machine Learning*.

[18] Germain, M., Gregor, K., Murray, I., and Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France. PMLR.

[19] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc.

[21] Goudet, O., Kalainathan, D., Caillou, P., Lopez-Paz, D., Guyon, I., and Sebag, M. (2018). Learning Functional Causal Models with Generative Neural Networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, Springer Series on Challenges in Machine Learning. Springer International Publishing.

[22] Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China. PMLR.

[23] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2007). A kernel method for the two-sample-problem. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19, pages 513–520. MIT Press.

[24] Gumbel, E. (1954). *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*. Applied mathematics series. U.S. Government Printing Office.

[25] Guo, R., Cheng, L., Li, J., Hahn, P. R., and Liu, H. (2020). A survey of learning causality with data. *ACM Computing Surveys*, 53(4):1–37.

[26] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257.

[27] Jaber, A., Kocaoglu, M., Shanmugam, K., and Bareinboim, E. (2020). Causal discovery from soft interventions with unknown targets: Characterization and learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561, Vancouver, Canada. Curran Associates, Inc.

[28] Jaber, A., Zhang, J., and Bareinboim, E. (2018). Causal identification under Markov equivalence. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, pages 978–987. AUAI Press.

[29] Jaber, A., Zhang, J., and Bareinboim, E. (2019). Causal identification under Markov equivalence: Completeness results. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2981–2989, Long Beach, CA. PMLR.

[30] Johansson, F. D., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 3020–3029. JMLR.org.

[31] Jung, Y., Tian, J., and Bareinboim, E. (2020a). Estimating causal effects using weighting-based estimators. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY. AAAI Press.

[32] Jung, Y., Tian, J., and Bareinboim, E. (2020b). Learning causal effects via weighted empirical risk minimization. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12697–12709, Vancouver, Canada. Curran Associates, Inc.

[33] Jung, Y., Tian, J., and Bareinboim, E. (2021). Estimating identifiable causal effects through double machine learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, number R-69, Vancouver, Canada. AAAI Press.

[34] Kallus, N. (2020). DeepMatch: Balancing deep covariate representations for causal inference using adversarial training. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5067–5077. PMLR.

[35] Karpathy, A. (2018). pytorch-made. `https://github.com/karpathy/pytorch-made` [Source Code].

[36] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[37] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[38] Kocaoglu, M., Jaber, A., Shanmugam, K., and Bareinboim, E. (2019). Characterization and learning of causal graphs with latent variables from soft interventions. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 14346–14356, Vancouver, Canada. Curran Associates, Inc.

[39] Kocaoglu, M., Shanmugam, K., and Bareinboim, E. (2017). Experimental design for learning causal graphs with latent variables. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 7018–7028. Curran Associates, Inc.

[40] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc.

[41] Lee, S. and Bareinboim, E. (2018). Structural causal bandits: Where to intervene? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 2568–2578, Montreal, Canada. Curran Associates, Inc.

[42] Lee, S. and Bareinboim, E. (2020). Characterizing optimal mixed policies: Where to intervene and what to observe. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8565–8576, Vancouver, Canada. Curran Associates, Inc.

[43] Lee, S., Correa, J. D., and Bareinboim, E. (2019). General Identifiability with Arbitrary Surrogate Experiments. In *Proceedings of the Thirty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence*, Corvallis, OR. AUAI Press, in press.

[44] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861 – 867.

[45] Li, S. and Fu, Y. (2017). Matching on balanced nonlinear representations for treatment effects estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 929–939. Curran Associates, Inc.

[46] Liu, Q., Lee, J., and Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 276–284, New York, New York, USA. PMLR.

[47] Loshchilov, I. and Hutter, F. (2017). SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[48] Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[49] Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6449–6459, Red Hook, NY, USA. Curran Associates Inc.

[50] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6231–6239. Curran Associates, Inc.

[51] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

[52] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

[53] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA.

[54] Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, 82(4):669–688.

[55] Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition.

[56] Pearl, J. and Mackenzie, D. (2018). *The Book of Why*. Basic Books, New York.

[57] Perković, E., Textor, J., Kalisch, M., and H. Maathuis, M. (2018). Complete Graphical Characterization and Construction of Adjustment Sets in Markov Equivalence Classes of Ancestral Graphs. *Journal of Machine Learning Research*, 18.

[58] Peters, J., Janzing, D., and Schlkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press.

[59] Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR.

[60] Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3076–3085, International Convention Centre, Sydney, Australia. PMLR.

[61] Shi, C., Blei, D. M., and Veitch, V. (2019). Adapting neural networks for the estimation of treatment effects. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual*

Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 2503–2513.

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

[63] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

[64] Tian, J. and Pearl, J. (2002). A General Identification Condition for Causal Effects. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI 2002)*, pages 567–573, Menlo Park, CA. AAAI Press/The MIT Press.

[65] Yao, L., Li, S., Li, Y., Huai, M., Gao, J., and Zhang, A. (2018). Representation learning for treatment effect estimation from observational data. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31, pages 2633–2643. Curran Associates, Inc.

[66] Yoon, J., Jordon, J., and van der Schaar, M. (2018). GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*.

[67] Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896.

[68] Zhang, J. and Bareinboim, E. (2021). Non-Parametric Methods for Partial Identification of Causal Effects. Technical Report Technical Report R-72, Columbia University, Department of Computer Science, New York.

A Proofs

In this section, we provide proofs of the statements in the main body of the paper. These results assume that the endogenous variables $\mathbf{V}$ are categorical (i.e. $\mathcal{D}_{\mathbf{V}}$ is discrete) but make no assumptions about the exogenous variables.

A.1 Proofs of Theorem 1 and Corollary 1

In addition to Def. 2, defining layers 1 and 2 of the PCH, we also require a definition for layer 3. While Def. 2 shows how the SCM valuates observational and interventional distributions, the following definition of layer 3 ([5, Def. 7]) shows how the SCM valuates counterfactual distributions, a family of distributions even more expressive than those from lower layers.

Definition 9 (Layer 3 Valuation)**.** An SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ induces a family of joint distributions over counterfactual events $\mathbf{Y}_{\mathbf{x}}, \dots, \mathbf{Z}_{\mathbf{w}}$, for any $\mathbf{Y}, \mathbf{Z}, \dots, \mathbf{X}, \mathbf{W} \subseteq \mathbf{V}$:

$$P^{\mathcal{M}}(\mathbf{y}_{\mathbf{x}}, \dots, \mathbf{z}_{\mathbf{w}}) = \sum_{\substack{\{\mathbf{u} \mid \mathbf{Y}_{\mathbf{x}}(\mathbf{u})=\mathbf{y}, \\ \dots,\ \mathbf{Z}_{\mathbf{w}}(\mathbf{u})=\mathbf{z}\}}} P(\mathbf{u}). \tag{7}$$

■

For the expressiveness proofs of this paper, we leverage some of the notation and results from [68]. These results focus on the idea of a canonical form of SCMs, first explored in a special case by [3]. Let $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ be any SCM. For each $V \in \mathbf{V}$, we denote $\mathcal{H}_V = \{h_V : \mathcal{D}_{\mathbf{pa}_V} \to \mathcal{D}_V\}$ as the set of all possible functions mapping from the domain of the parents $\mathbf{pa}_V$ to the domain of V. We will order the elements of $\mathcal{H}_V$ as $h_V^{(1)}, \dots, h_V^{(m_V)}$, where $m_V = |\mathcal{H}_V|$. Since $\mathcal{H}_V$ fully exhausts all possible functions, we can partition $\mathcal{D}_{\mathbf{U}_V}$ into sets $\mathcal{D}_{\mathbf{U}_V}^{(1)}, \dots, \mathcal{D}_{\mathbf{U}_V}^{(m_V)}$ such that $\mathbf{u}_V \in \mathcal{D}_{\mathbf{U}_V}^{(r_V)}$ if and only if $f_V(\cdot, \mathbf{u}_V) = h_V^{(r_V)}$.

Lemma 1 ([68, Lem. 1])**.** *For an SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$, for each $V \in \mathbf{V}$, function $f_V \in \mathcal{F}$ can be expressed as*

$$f_V(\mathbf{pa}_V, \mathbf{u}_V) = \sum_{r_V=1}^{m_V} h_V^{(r_V)}(\mathbf{pa}_V) \mathbb{1}\left\{\mathbf{u}_V \in \mathcal{D}_{\mathbf{U}_V}^{(r_V)}\right\}$$

■

Definition 10 (Canonical SCM)**.** A canonical SCM is an SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ such that

1. $\mathbf{U} = \{R_V : V \in \mathbf{V}\}$, where $\mathcal{D}_{R_V} = \{1, \dots, m_V\}$ (where $m_V = |\{h_V : \mathcal{D}_{\mathbf{pa}_V} \to \mathcal{D}_V\}|$) for each $V \in \mathbf{V}$.
2. For each $V \in \mathbf{V}$, $f_V \in \mathcal{F}$ is defined as
$$f_V(\mathbf{pa}_V, r_V) = h_V^{(r_V)}(\mathbf{pa}_V).$$

■

Lemma 2. *For any SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$, there exists a canonical SCM $\mathcal{M}_{\mathsf{CM}} = \langle \mathbf{U}_{\mathsf{CM}}, \mathbf{V}, \mathcal{F}_{\mathsf{CM}}, P(\mathbf{U}_{\mathsf{CM}}) \rangle$ such that $\mathcal{M}_{\mathsf{CM}}$ is L_3-consistent with $\mathcal{M}$.* ■

Proof. Since $\mathbf{U}_{\mathsf{CM}}$ and $\mathcal{F}_{\mathsf{CM}}$ are already fixed, we choose $P(\mathbf{U}_{\mathsf{CM}})$ to fix our choice of $\mathcal{M}_{\mathsf{CM}}$. For $\mathbf{r} \in \mathcal{D}_{\mathbf{U}_{\mathsf{CM}}}$, we choose

$$P^{\mathcal{M}_{\mathsf{CM}}}(\mathbf{U}_{\mathsf{CM}} = \mathbf{r}) \tag{8}$$
$$= P^{\mathcal{M}_{\mathsf{CM}}}(R_{V_1} = r_{V_1}, \dots, R_{V_n} = r_{V_n}) \tag{9}$$
$$:= P^{\mathcal{M}}\left(\mathbf{U}_{V_1} \in \mathcal{D}_{\mathbf{U}_{V_1}}^{(r_{V_1})}, \dots, \mathbf{U}_{V_n} \in \mathcal{D}_{\mathbf{U}_{V_n}}^{(r_{V_n})}\right). \tag{10}$$

For $\mathbf{r} \in \mathcal{D}_{\mathbf{U}_{\mathsf{CM}}}$, denote

$$\mathcal{D}_{\mathbf{U}}^{(\mathbf{r})} = \left\{\mathbf{u} : \mathbf{u} \in \mathcal{D}_{\mathbf{U}}, \mathbf{u}_V \in \mathcal{D}_{\mathbf{U}_V}^{(r_V)} \quad \forall V \in \mathbf{V}\right\}$$

We now show that $\mathcal{M}_{\mathsf{CM}}$ and $\mathcal{M}$ valuate in the same way any query of the form $P(\boldsymbol{\varphi})$, where

$$\boldsymbol{\varphi} = \bigwedge_{i \in \{1, \dots, k\}} \mathbf{Y}^i_{\mathbf{x}_i} = \mathbf{y}_i$$

for any $\mathbf{X}_i, \mathbf{Y}_i \subseteq \mathbf{V}$, $\mathbf{Y}_i \neq \emptyset$, $\mathbf{y}_i \in \mathcal{D}_{\mathbf{Y}_i}$, and positive integer k. We say $\mathcal{M}(\mathbf{u}) \models \boldsymbol{\varphi}$ for $\mathbf{u} \in \mathcal{D}_{\mathbf{U}}$ if for all $i \in \{1, \dots, k\}$, $\mathbf{Y}^i_{\mathbf{x}_i}(\mathbf{u}) = \mathbf{y}_i$. We define this notation similarly for $\mathcal{M}_{\mathsf{CM}}$.

Let $\mathbf{u}^1, \mathbf{u}^2 \in \mathcal{D}_{\mathbf{U}}$ be any two instantiations of $\mathbf{U}$. If $\mathbf{u}^1$ and $\mathbf{u}^2$ come from the same partition $\mathcal{D}_{\mathbf{U}}^{(\mathbf{r})}$, then we have for all $V \in \mathbf{V}$,

$$\begin{aligned}
&f_V(\mathbf{pa}_V, \mathbf{u}^1_V) \\
&= \sum_{r'_V=1}^{m_V} h_V^{(r'_V)}(\mathbf{pa}_V) \mathbb{1}\left\{\mathbf{u}^1_V \in \mathcal{D}_{\mathbf{U}_V}^{(r'_V)}\right\} && \text{Lem. 1} \\
&= h_V^{(r_V)}(\mathbf{pa}_V) \\
&= \sum_{r'_V=1}^{m_V} h_V^{(r'_V)}(\mathbf{pa}_V) \mathbb{1}\left\{\mathbf{u}^2_V \in \mathcal{D}_{\mathbf{U}_V}^{(r'_V)}\right\} \\
&= f_V(\mathbf{pa}_V, \mathbf{u}^2_V) && \text{Lem. 1.}
\end{aligned}$$

Hence,

$$\mathcal{M}(\mathbf{u}^1) \models \boldsymbol{\varphi} \Leftrightarrow \mathcal{M}(\mathbf{u}^2) \models \boldsymbol{\varphi}. \tag{11}$$

Let $\mathbf{u} \in \mathcal{D}_{\mathbf{U}}$ and let $\mathbf{r} \in \mathcal{D}_{\mathbf{U}_{\mathsf{CM}}}$. Then if $\mathbf{u} \in \mathcal{D}_{\mathbf{U}}^{(\mathbf{r})}$, we have for all $V \in \mathbf{V}$

$$\begin{aligned}
&f_V(\mathbf{pa}_V, \mathbf{u}_V) \\
&= \sum_{r'_V=1}^{m_V} h_V^{(r'_V)}(\mathbf{pa}_V) \mathbb{1}\left\{\mathbf{u}_V \in \mathcal{D}_{\mathbf{U}_V}^{(r'_V)}\right\} && \text{Lem. 1} \\
&= h_V^{(r_V)}(\mathbf{pa}_V) \\
&= f_V^{\mathsf{CM}}(\mathbf{pa}_V, \mathbf{u}_V) && \text{Def. 10.}
\end{aligned}$$

Hence,

$$\mathcal{M}(\mathbf{u}) \models \boldsymbol{\varphi} \Leftrightarrow \mathcal{M}_{\mathsf{CM}}(\mathbf{r}) \models \boldsymbol{\varphi}. \tag{12}$$

Then, by the previous statements and $\mathcal{M}_{\mathsf{CM}}$'s construction, we have

$$\begin{aligned}
P^{\mathcal{M}}(\boldsymbol{\varphi}) &= \sum_{\{\mathbf{u}: \mathcal{M}(\mathbf{u}) \models \boldsymbol{\varphi}\}} P^{\mathcal{M}}(\mathbf{u}) \\
&= \sum_{\{\mathbf{r}: \mathcal{M}(\mathbf{u}) \models \boldsymbol{\varphi}, \mathbf{u} \in \mathcal{D}_{\mathbf{U}}^{(\mathbf{r})}\}} P^{\mathcal{M}}\left(\mathbf{U} \in \{\mathcal{D}_{\mathbf{U}}^{(\mathbf{r})}\}\right) \\
&\text{by Eq. 11} \\
&= \sum_{\{\mathbf{r}: \mathcal{M}(\mathbf{u}) \models \boldsymbol{\varphi}, \mathbf{u} \in \mathcal{D}_{\mathbf{U}}^{(\mathbf{r})}\}} P^{\mathcal{M}_{\mathsf{CM}}}(\mathbf{r}) \\
&\text{by Eq. 10} \\
&= \sum_{\{\mathbf{r}: \mathcal{M}_{\mathsf{CM}}(\mathbf{r}) \models \boldsymbol{\varphi}\}} P^{\mathcal{M}_{\mathsf{CM}}}(\mathbf{r}) \\
&\text{by Eq. 12} \\
&= P^{\mathcal{M}_{\mathsf{CM}}}(\boldsymbol{\varphi}).
\end{aligned}$$

□

Lemma 2 shows that the canonical SCM can be used as a representative of equivalence classes of SCMs. In the case where $\mathcal{D}_{\mathbf{V}}$ is discrete, the mapping from an SCM to an equivalent canonical model conveniently also remaps $\mathcal{D}_{\mathbf{U}}$ to a discrete space. We next show that any canonical SCM can be constructed in the form of an NCM.

We will focus on feedforward neural networks, specifically multi-layer perceptrons (MLPs) with the binary step activation function, even though other types of neural networks could be compatible with the statement proven here (see Appendix D.1).

Definition 11 (Multi-layer Perceptron)**.** A neural network node is a function defined as

$$\hat{f}(\mathbf{x}; \mathbf{w}, b) = \sigma\left(\sum_i \mathbf{w}_i \mathbf{x}_i + b\right),$$

where $\mathbf{x}$ is a vector of real-valued inputs, $\mathbf{w}$ and b are the real-valued learned weights and bias respectively, and σ is an activation function. For this work, we will often denote σ as the binary step function for our activation function:

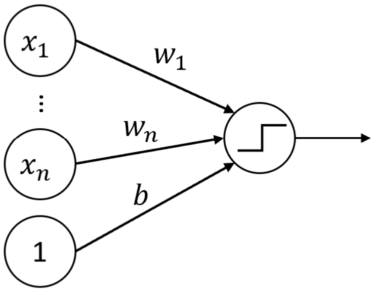


Figure 6: Example diagram of a neural network node from Definition 11. Nodes on the left are inputs, numbers on the edges represent weights, and the weighted sum of the inputs is passed through the binary step activation function.

$$\sigma(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0. \end{cases}$$

This is simply one choice of activation function which always outputs a binary result. (Figure 6 provides an illustration of such a node.)

A neural network layer of width k is comprised of k neural network nodes with the same input vector, together outputting a k-dimensional output:

$$\hat{f}(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \left(\hat{f}_1(\mathbf{x}; \mathbf{w}_1, b_1), \ldots, \hat{f}_k(\mathbf{x}; \mathbf{w}_k, b_k)\right),$$

where $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ and $\mathbf{b} = \{b_1, \ldots, b_k\}$. An MLP is defined as a function comprised of several neural network layers $\hat{f}_1, \ldots, \hat{f}_\ell$, with each layer taking the previous layer's output as its input:

$$\hat{f}_{\text{MLP}}(\mathbf{x}) = \hat{f}_\ell(\ldots \hat{f}_1(\mathbf{x}; \mathbf{W}_1, \mathbf{b}_1) \ldots; \mathbf{W}_\ell, \mathbf{b}_\ell).$$

This means that a neural network is a function that is a composition of the functions of the individual layers, where the input is the input to the first layer, and the output is the output of the last layer. ■

We will show next three basic lemmas (3-5) that will be used later on to help understand the expressiveness of the networks introduced from Def. 11.

Lemma 3. *For any function $f : \mathbf{X} \to Y$ mapping a set of binary variables to a binary variable, there exists an equivalent MLP $\hat{f}$ using binary step activation functions.* ■

Proof. We define the following three neural network components:

- Given binary input x, with $w = -1$ and $b = 0$, neural network function
$$\hat{f}_{\text{NOT}}(x) = \sigma(-x)$$
outputs the negation of x.
- Given binary vector input $\mathbf{x}$, with $\mathbf{w} = \mathbf{1}$ and $b = -1$, neural network function
$$\hat{f}_{\text{OR}}(\mathbf{x}) = \sigma\left(\sum_i x_i - 1\right)$$
outputs the bitwise-OR of $\mathbf{x}$.
- Given binary vector input $\mathbf{x}$, with $\mathbf{w} = \mathbf{1}$ and $b = -|\mathbf{x}|$, neural network function
$$\hat{f}_{\text{AND}}(\mathbf{x}) = \sigma\left(\sum_i x_i - |\mathbf{x}|\right)$$
outputs the bitwise-AND of $\mathbf{x}$.

Since all functions mapping a set of binary variables to a binary variable can be written in disjunctive normal form (DNF), we can combine these three components to build $\hat{f}$. □

Lemma 4. *For any function $f : \mathbf{X} \to Y$ mapping set of variables $\mathbf{X}$ to variable Y, all from countable numerical domains, there exists an equivalent MLP $\hat{f}$ using binary step activations.* ■

Proof. For each value $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$, we first aim to assign a unique binary representation $\text{bin}(\mathbf{x})$, which we can use more flexibly due to Lemma 3. One simple way to accomplish this is to map the values to a one-hot encoding, a binary vector for which each element corresponds to a unique value in $\mathcal{D}_{\mathbf{X}}$.

We will use here a neural function with $w = 1$ and $b = -z$, so we have

$$\hat{f}_{\geq z}(x) = \sigma(x - z)$$

which, on input x, outputs 1 if $x \leq z$ or 0 otherwise. We will also borrow the binary functions from the proof of Lem. 3.

For each $X_i \in \mathbf{X}$ and each $x_i \in \mathcal{D}_{X_i}$, we construct neural network function

$$\hat{f}_{=x_i}(x) = \hat{f}_{\text{AND}}\left(\hat{f}_{\leq x_i}(x), (\forall x'_i < x_i)\hat{f}_{\text{NOT}}\left(\hat{f}_{\leq x'_i}(x)\right)\right)$$

where $x'_i \in \mathcal{D}_{X_i}$, which, on input $x \in \mathcal{D}_{X_i}$, outputs 1 if $x = x_i$ or 0 otherwise.

We can then define for each $\mathbf{z} \in \mathcal{D}_{\mathbf{X}}$

$$\hat{f}_{=\mathbf{z}}(\mathbf{x}) = \hat{f}_{\text{AND}}\left(\forall i \hat{f}_{=z_i}(x_i)\right)$$

which, on input $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$, outputs 1 if $\mathbf{x} = \mathbf{z}$ or 0 otherwise. Here, x_i and z_i denote the ith element of $\mathbf{x}$ and $\mathbf{z}$ respectively.

We can then define an one-hot binary representation of $\mathbf{x}$, $\text{bin}(\mathbf{x})$, to be a vector of the outputs of $\hat{f}_{=\mathbf{z}}(\mathbf{x})$ for all $\mathbf{z} \in \mathcal{D}_{\mathbf{X}}$:

$$\hat{f}_{\text{ENC}}(\mathbf{x}) = \left(\forall(\mathbf{z} \in \mathcal{D}_{\mathbf{X}})\hat{f}_{=\mathbf{z}}(\mathbf{x})\right)$$

This representation is a binary vector of length $|\mathcal{D}_{\mathbf{X}}|$ and is unique for each value of $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$ because $\hat{f}_{=\mathbf{z}}(\mathbf{x}) = 1$ if and only if $\mathbf{x} = \mathbf{z}$, so a different bit is 1 for every choice of $\mathbf{x}$.

We can similarly define a binary representation for each $y \in \mathcal{D}_Y$, $\text{bin}(y)$, as a binary vector of length $|\mathcal{D}_Y|$, where each bit corresponds to a value in $\mathcal{D}_Y$. If $y_i \in \mathcal{D}_Y$ is the value that corresponds with the ith bit of $\text{bin}(y)$, then $\text{bin}(y)_i = 1$ if and only if $y = y_i$. Now we consider the translation from $\text{bin}(y)$ back into y using neural networks. We can create the neural network function on input $\text{bin}(y)$ with $\mathbf{w} = (y_i : y_i \in \mathcal{D}_Y)$ and $b = 0$,

$$\hat{f}_{\text{DEC}}(\text{bin}(y)) = \mathbf{w}^{\intercal}\,\text{bin}(y),$$

omitting the binary step activation function σ. This function simply computes the dot product of $\text{bin}(y)$ with a vector of all of the possible values of Y, which results in y since $\text{bin}(y)$ is 0 in every location except for the bit corresponding to y.

Combining all of these constructed neural network functions, we can construct a final MLP $\hat{f}$ for mapping $\mathbf{X}$ to Y:

1. On input $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$, convert $\mathbf{x}$ to $\text{bin}(\mathbf{x})$ using $\hat{f}_{\text{ENC}}(\mathbf{x})$.
2. By Lemma 3, find some MLP mapping $\text{bin}(\mathbf{x})$ to $\text{bin}(y)$.
3. Finally, use $\hat{f}_{\text{DEC}}$ to convert $\text{bin}(y)$ to y.

The final MLP $\hat{f}$ is the composition of all of the neural networks used to realize these three steps. □

Although neural networks as defined in Def. 11 are undefined for non-numerical inputs and outputs, any kind of categorical data can be considered if first converted into a numerical representation.

The above two lemmas show that MLPs can be used to express any function, but we will need another result to incorporate the exogenous sources of randomness. Specifically, we show that MLPs can map $\text{Unif}(0, 1)$ noise to any other distribution of variables.

Lemma 5 (Neural Inverse Probability Integral Transform (Discrete)). *For any probability mass function $P(\mathbf{X})$, there exists an MLP $\hat{f}$ which maps* $\text{Unif}(0,1)$ *to* $P(\mathbf{X})$. ■

Proof. Let $\mathbf{x}_1, \mathbf{x}_2, \dots$ be the elements of the support of $P(\mathbf{X})$, ordered arbitrarily. We also define some arbitrary $\mathbf{x}_0$ such that $P(\mathbf{x}_0) = 0$. For each $i \in \{0, 1, 2, \dots\}$, construct neural network function, with $w = 1$ and $b = -\sum_{j=0}^{i} P(\mathbf{x}_j)$

$$\hat{f}_{\mathbf{x}_i}(u) = \sigma\left(u - \sum_{j=0}^{i} P(\mathbf{x}_j)\right)$$

which, on input u, returns 1 if and only if $u \geq \sum_{j=0}^{i} P(\mathbf{x}_j)$. Note that $\hat{f}_{\mathbf{x}_0}(u)$ is always 1. We then construct a neural network function $\hat{f}_{\text{OUT}}$ which, on inputs $(\hat{f}_{\mathbf{x}_0}, \hat{f}_{\mathbf{x}_1}, \hat{f}_{\mathbf{x}_2}, \dots)$, outputs one of $\mathbf{x}_1, \mathbf{x}_2, \dots$. Specifically, it operates as follows:

1. For each $i \in \{1, 2, \dots\}$, if $\hat{f}_{\mathbf{x}_i} = 0$ and $\hat{f}_{\mathbf{x}_{i-1}} = 1$, then output $\mathbf{x}_i$.
2. If none hold, output any arbitrary $\mathbf{x}_i$ (this will never happen).

By Lemma 4, we can construct such a function since all $\hat{f}_{\mathbf{x}_i}$ are binary. Then, let $\hat{g}(u) = \hat{f}_{\text{OUT}}(\hat{f}_{\mathbf{x}_0}(u), \hat{f}_{\mathbf{x}_1}(u), \hat{f}_{\mathbf{x}_2}(u), \dots)$. Observe that for u sampled from $\text{Unif}(0,1)$,

$$\begin{aligned}
P(\hat{g}(u) = \mathbf{x}_i) &= P\left(\hat{f}_{\text{OUT}}\left(\hat{f}_{\mathbf{x}_0}(u), \hat{f}_{\mathbf{x}_1}, \hat{f}_{\mathbf{x}_2}, \dots\right) = \mathbf{x}_i\right) \\
&= P\left(\hat{f}_{\mathbf{x}_i}(u) = 0 \wedge \hat{f}_{\mathbf{x}_{i-1}}(u) = 1\right) \\
&= P\left(u < \sum_{j=0}^{i} P(\mathbf{x}_j) \wedge u \geq \sum_{j=0}^{i-1} P(\mathbf{x}_j)\right) \\
&= P\left(\sum_{j=0}^{i-1} P(\mathbf{x}_j) \leq u < \sum_{j=0}^{i} P(\mathbf{x}_j)\right) \\
&= \sum_{j=0}^{i} P(\mathbf{x}_j) - \sum_{j=0}^{i-1} P(\mathbf{x}_j) \\
&= P(\mathbf{x}_i)
\end{aligned}$$

for each $i \in \{1, 2, \dots\}$. Therefore, we see that $\hat{g}$ successfully maps the $\text{Unif}(0,1)$ distribution to $P(\mathbf{X})$. □

We can now combine these neural network results with the canonical SCM results to complete the expressiveness proof for NCMs.

Theorem 1 (NCM Expressiveness). *For any SCM $\mathcal{M}^* = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$, there exists an NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ s.t. $\widehat{M}$ is L_3-consistent w.r.t. $\mathcal{M}^*$.* ■

Proof. Lemma 2 guarantees that there exists a canonical SCM $\mathcal{M}_{\textsf{CM}} = \langle \mathbf{U}_{\textsf{CM}}, \mathbf{V}, \mathcal{F}_{\textsf{CM}}, P(\mathbf{U}_{\textsf{CM}}) \rangle$ that is L_3-consistent with $\mathcal{M}^*$. Hence, to construct $\widehat{M}$, it suffices to show how to construct $\mathcal{M}_{\textsf{CM}}$ using the architecture of an NCM.

Following the structure of Def. 3, we choose $\widehat{\mathbf{U}} = \{\widehat{U}_{\mathbf{V}}\}$. For each $V_i \in \mathbf{V}$, we construct $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ using the following components:

1. By Lemma 5, construct $\hat{f}^R_{V_i} : \mathcal{D}_{\widehat{U}_{\mathbf{V}}} \to \mathcal{D}_{\mathbf{U}_{\textsf{CM}}}$ such that

$$\hat{f}^R_{V_i}(\widehat{u}_{\mathbf{V}}) = \mathbf{u}_{\textsf{CM}}, \tag{13}$$

where

$$P^{\widehat{M}}\left(\hat{f}^R_{V_i}(\widehat{U}_{\mathbf{V}}) = \mathbf{u}_{\textsf{CM}}\right) = P^{\mathcal{M}_{\textsf{CM}}}(\mathbf{u}_{\textsf{CM}}). \tag{14}$$

2. By Lemma 4, construct $\hat{f}^H_{V_i} : \mathcal{D}_{\mathbf{Pa}_{V_i}} \times \mathcal{D}_{\mathbf{U}_{\text{CM}}} \rightarrow \mathcal{D}_{V_i}$ such that

$$\begin{aligned} \hat{f}^H_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}_{\text{CM}}) &= f^{\text{CM}}_{V_i}(\mathbf{pa}_{V_i}, r_{V_i}) && (15) \\ &= h^{(r_V)}_{V_i}(\mathbf{pa}_{V_i}) && (16) \end{aligned}$$

where r_{V_i} is the value in $\mathbf{u}_{\text{CM}}$ corresponding to V_i.

Combining these two components leads to MLP

$$\hat{f}_{V_i}(\mathbf{pa}_{V_i}, \widehat{u}_{\mathbf{V}}) = \hat{f}^H_{V_i}\left(\mathbf{pa}_{V_i}, \hat{f}^R_{V_i}(\widehat{u}_{\mathbf{V}})\right) \tag{17}$$

Although this does not exactly fit the structure in Def. 11 because $\mathbf{pa}_V$ is not included as an input into $\hat{f}^R_{V_i}$, this can be altered by simply having $\hat{f}^R_{V_i}$ accepting $\mathbf{pa}_V$ as an input and outputting it alongside $\mathbf{u}_{\text{CM}}$ without changing it.

By Eqs. 14 and 16, the NCM $\widehat{M}$ is constructed to match $\mathcal{M}_{\text{CM}}$ on all outputs. Hence, for any counterfactual query $\boldsymbol{\varphi}$, we have

$$\mathcal{M}_{\text{CM}} \models \boldsymbol{\varphi} \Leftrightarrow \widehat{M} \models \boldsymbol{\varphi}$$

and therefore

$$\mathcal{M}^* \models \boldsymbol{\varphi} \Leftrightarrow \widehat{M} \models \boldsymbol{\varphi}.$$

□

While Thm. 1 demonstrates the expressive power of an SCM parameterized by neural networks, we now consider its limitations. Notably, we show in the sequel that NCMs suffer from the same consequences implied by the CHT.

Fact 1 (Causal Hierarchy Theorem (CHT) [5, Thm. 1])**.** *Let Ω^* be the set of all SCMs. We say that Layer j of the causal hierarchy for SCMs* collapses *to Layer i ($i < j$) relative to $\mathcal{M}^* \in \Omega^*$ if $L_i(\mathcal{M}^*) = L_i(\mathcal{M})$ implies that $L_j(\mathcal{M}^*) = L_j(\mathcal{M})$ for all $\mathcal{M} \in \Omega^*$. Then, with respect to the Lebesgue measure over (a suitable encoding of L_3-equivalence classes of) SCMs, the subset in which Layer j of NCMs collapses to Layer i is measure zero.* ■

We prove a similar result for NCMs as a corollary of Fact 1 and Thm. 1.

Corollary 1 (Neural Causal Hierarchy Theorem (N-CHT))**.** *Let Ω^* and Ω be the sets of all SCMs and NCMs, respectively. We say that Layer j of the causal hierarchy for NCMs* collapses *to Layer i ($i < j$) relative to $\mathcal{M}^* \in \Omega^*$ if $L_i(\mathcal{M}^*) = L_i(\widehat{M})$ implies that $L_j(\mathcal{M}^*) = L_j(\widehat{M})$ for all $\widehat{M} \in \Omega$. Then, with respect to the Lebesgue measure over (a suitable encoding of L_3-equivalence classes of) SCMs, the subset in which Layer j of NCMs collapses to Layer i has measure zero.* ■

Proof. Since all NCMs are SCMs, an SCM-collapse with respect to $\mathcal{M}^*$ also implies an NCM-collapse with respect to $\mathcal{M}^*$.

If layer j does not SCM-collapse to layer i with respect to $\mathcal{M}^*$, then there exists an SCM $\mathcal{M}$ such that $L_i(\mathcal{M}^*) = L_i(\mathcal{M})$ but $L_j(\mathcal{M}^*) \neq L_j(\mathcal{M})$. By Thm. 1, this implies that there exists an NCM $\widehat{\mathcal{M}}$ such that $L_i(\mathcal{M}^*) = L_i(\widehat{\mathcal{M}})$ but $L_j(\mathcal{M}^*) \neq L_j(\widehat{\mathcal{M}})$, which means that layer j also does not NCM-collapse to layer i.

These two statements imply that the set of SCMs that undergo some form of SCM-collapse is equivalent to the set of SCMs that undergo some form of NCM-collapse. Therefore, the result from Fact 1 must also hold for NCMs. □

A.2 Proof of Theorem 2

The results proven in this section involve the incorporation of structural constraints, as introduced through the graphical treatment provided in [54], and made it explicit and generalized for models with latent variables in [5, Sec.1. 4]. For convenience, we list the basic definitions below, but refer the readers to the references for more detailed explanations and further examples.

Definition 12 (Causal Diagram [5, Def. 13])**.** Consider an SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$. We construct a graph $\mathcal{G}$ using $\mathcal{M}$ as follows:

(1) add a vertex for every variable in $\mathbf{V}$,

(2) add a directed edge ($V_j \rightarrow V_i$) for every $V_i, V_j \in \mathbf{V}$ if V_j appears as an argument of $f_{V_i} \in \mathcal{F}$,

(3) add a bidirected edge ($V_j \dashleftarrow\!\!\dashrightarrow V_i$) for every $V_i, V_j \in \mathbf{V}$ if the corresponding $\mathbf{U}_{V_i}, \mathbf{U}_{V_j} \subseteq \mathbf{U}$ are not independent or if f_{V_i} and f_{V_j} share some $U \in \mathbf{U}$ as an argument.

We refer to $\mathcal{G}$ as the causal diagram induced by $\mathcal{M}$ (or "causal diagram of $\mathcal{M}$" for short). ■

Definition 13 (Confounded Component [5, Def. 14])**.** Let G be a causal diagram. Let $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$ be a partition over the set of variables $\mathbf{V}$, where $\mathbf{C}_i$ is said to be a confounded component (C-component for short) of G if for every $V_a, V_b \in \mathbf{C}_i$, there exists a path made entirely of bidirected edges between V_a and V_b in G, and $\mathbf{C}_i$ is maximal. We denote $\mathbf{C}(V_a)$ as the C-component containing V_a. ■

Definition 14 (Semi-Markov Relative [5, Def. 15])**.** A distribution P is said to be *semi-Markov relative* to a graph G if for any topological order $<$ of G through its directed edges, P factorizes as

$$P(\mathbf{v}) = \prod_{V_i \in \mathbf{V}} P(v_i \mid \mathbf{pa}^{+}_{V_i}), \tag{18}$$

where $\mathbf{Pa}^{+}_{V_i} = \mathbf{Pa}(\{V \in \mathbf{C}(V_i) : V \leq V_i\})$, with $\leq$ referring to the topological ordering. ■

Definition 15 (Causal Bayesian Network (CBN) [5, Def. 16])**.** Given observed variables $\mathbf{V}$, let $\mathbf{P}^*$ be the collection of all interventional distributions $P(\mathbf{V} \mid do(\mathbf{x}))$, $\mathbf{X} \subseteq \mathbf{V}$, $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$. A causal diagram $\mathcal{G}$ is a Causal Bayesian Network for $\mathbf{P}^*$ if for every intervention $do(\mathbf{X} = \mathbf{x})$ and every topological ordering $<$ of $\mathcal{G}_{\overline{X}}$ through its directed edges,

(i) $P(\mathbf{V} \mid do(\mathbf{X} = \mathbf{x}))$ is semi-Markov relative to $\mathcal{G}_{\overline{\mathbf{X}}}$.

(ii) For every $V_i \in \mathbf{V} \setminus \mathbf{X}$, $\mathbf{W} \subseteq \mathbf{V} \setminus (\mathbf{Pa}^{\mathbf{X}+}_{i} \cup \mathbf{X} \cup \{V_i\})$:

$$P(v_i \mid do(\mathbf{x}), \mathbf{pa}^{\mathbf{x}+}_{i}, do(\mathbf{w})) = P(v_i \mid do(\mathbf{x}), \mathbf{pa}^{\mathbf{x}+}_{i})$$

,

(iii) For every $V_i \in \mathbf{V} \setminus \mathbf{X}$, let $\mathbf{Pa}^{\mathbf{X}+}_{i}$ be partitioned into two sets of confounded and unconfounded parents, $\mathbf{Pa}^{c}_{i}$ and $\mathbf{Pa}^{u}_{i}$ in $\mathcal{G}_{\overline{\mathbf{X}}}$. Then

$$\begin{aligned} &P(v_i \mid do(\mathbf{x}), \mathbf{pa}^{c}_{i}, do(\mathbf{pa}^{u}_{i})) \\ &= P(v_i \mid do(\mathbf{x}), \mathbf{pa}^{c}_{i}, \mathbf{pa}^{u}_{i}) \end{aligned}$$

Here, $\mathbf{Pa}^{\mathbf{x}+}_{V_i} = \mathbf{Pa}(\{V \in \mathbf{C}_{\overline{\mathbf{X}}}(V_i) : V \leq V_i\})$, with $\mathbf{C}_{\overline{\mathbf{X}}}$ referring to the corresponding C-component in $\mathcal{G}_{\overline{\mathbf{X}}}$ and $\leq$ referring to the topological ordering. ■

In fact, for any SCM $\mathcal{M}$, its induced causal diagram and interventional distributions form a CBN. This means that the diagram encodes the qualitative constraints induced over the space of interventional distributions, despite the specific values that these distributions attain and the $\mathcal{F}$ and $P(\mathbf{U})$ of $\mathcal{M}$.

Fact 2 (SCM-CBN L_2 connection [5, Thm. 4])**.** *The causal diagram $\mathcal{G}$ induced by SCM $\mathcal{M}$ is a CBN for $L_2(\mathcal{M})$.* ■

We can now show that, indeed, all of the CBN constraints implied by a causal diagram $\mathcal{G}$ are encoded in a $\mathcal{G}$-constrained NCM constructed via Def. 7.

Lemma 6. *Let $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ be a $\mathcal{G}$-constrained NCM. Let $\widehat{\mathcal{G}}$ be the causal diagram induced by $\widehat{M}$. Then $\widehat{\mathcal{G}} = \mathcal{G}$.* ■

Proof. Considering Def. 12 in the context of $\widehat{\mathcal{G}}$'s construction, note that by step 1 all of the vertices match, simply having one for each variable in $\mathbf{V}$.

Step 2 adds a directed edge from V_i to V_j if $\hat{f}_{V_i}$ has V_j as an argument. By step 2 of Def. 7, $\mathbf{pa}_{V_j}$ will contain V_i if and only if there was a directed edge from V_i to V_j in $\mathcal{G}$. This implies that $\hat{f}_{V_j}$ will contain V_i as an argument if and only if there was a directed edge from V_i to V_j in $\mathcal{G}$, so the directed edges must also match in $\widehat{\mathcal{G}}$.

Finally, step 3 of Def. 12 states that a bidirected edge between V_i and V_j is added to $\widehat{\mathcal{G}}$ when $\hat{f}_{V_i}$ and $\hat{f}_{V_j}$ share some $\widehat{U} \in \widehat{\mathbf{U}}$ as an argument or have arguments from $\widehat{\mathbf{U}}$ that are not independent. Def. 3 ensures that all variables in $\widehat{\mathbf{U}}$ are independent, so a shared $\widehat{U} \in \widehat{\mathbf{U}}$ between functions in $\widehat{\mathcal{F}}$ is the only way a bidirected edge would be generated in $\widehat{\mathcal{G}}$. Step 1 of Def. 7 constructs $\widehat{U}$ such that it contains some $\widehat{U}_{\mathbf{C}}$ if and only if $\mathbf{C}$ is a C^2-component in $\mathcal{G}$. If $V_i, V_j \in \mathbf{V}$ are connected by a bidirected edge in $\mathcal{G}$, then there must exist some C^2-component $\mathbf{C}$ in $\mathcal{G}$ such that $V_i, V_j \in \mathbf{C}^*$, so there must exist $\widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}}$. Hence, since $\widehat{U}_{\mathbf{C}}$ is shared by both $\hat{f}_{V_i}$ and $\hat{f}_{V_j}$, the corresponding bidirected edge in $\mathcal{G}$ must also match in $\widehat{\mathcal{G}}$.

Therefore, since all vertices and edges match between $\mathcal{G}$ and $\widehat{\mathcal{G}}$, we can conclude that $\mathcal{G} = \widehat{\mathcal{G}}$. □

Theorem 2 (NCM $\mathcal{G}$-Consistency)**.** *Any $\mathcal{G}$-constrained NCM $\widehat{M}(\boldsymbol{\theta})$ is $\mathcal{G}$-consistent.* ■

Proof. This follows directly from Lemma 6 and Fact 2. □

A.3 Proof of Theorem 3

Our proof approach for Theorem 3 will be similar to that of Theorem 1. However, one notable difference is that Theorem 3 is no longer focused on layer 3 of the PCH, and we simplify this proof and follow some results from [68], as discussed next.

Consider once again $\mathcal{H}_V = \{h_V : \mathcal{D}_{\mathbf{pa}_V} \rightarrow \mathcal{D}_V\}$ as the set of all possible functions mapping from the domain of the parents $\mathbf{pa}_V$ to the domain of V, and let the elements of $\mathcal{H}_V$ be ordered as $h_V^{(1)}, \dots, h_V^{(m_V)}$, where $m_V = |\mathcal{H}_V|$. We utilize a new type of canonical model that is constructed to fit a specific causal diagram.

Definition 16 ($\mathcal{G}$-Canonical SCM [68, Def. 6])**.** Given a causal diagram $\mathcal{G}$, a *$\mathcal{G}$-canonical SCM* is an SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ such that

1. $\mathbf{U} = \{\mathbf{R}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G})\}$, where $\mathbb{C}(\mathcal{G})$ is the set of all C^2-components of $\mathcal{G}$. For each $\mathbf{C}$, we have $\mathbf{R}_{\mathbf{C}} = \{R_{V,\mathbf{C}} : V \in \mathbf{C}\}$.
2. For each $R_{V,\mathbf{C}}$, we have $\mathcal{D}_{R_{V,\mathbf{C}}} = \{0,1\}^{\mathbb{N}}$, so each $r_{V,\mathbf{C}}$ is a binary sequence $\left(r_{V,\mathbf{C}}^{(i)}\right)_{i \in \mathbb{N}}$.
3. For each $V \in \mathbf{V}$, $\mathbf{R}_V = \{\mathbf{R}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C} \text{ s.t. } V \in \mathbf{C}\}$.
4. For each $V \in \mathbf{V}$, we define $f_V \in \mathcal{F}$ as follows:
$$f_V(\mathbf{pa}_V, \mathbf{r}_V) = \sum_{i \in \mathbb{N}} h_V^{(j_V^{(i)})}(\mathbf{pa}_V) \prod_{\mathbf{R}_{\mathbf{C}} \in \mathbf{R}_V} r_{V,\mathbf{C}}^{(i)}$$
where $j_V^{(i)} \in \{1, \dots, m_V\}$ and $\sum_{i \in \mathbb{N}} \prod_{\mathbf{R}_{\mathbf{C}} \in \mathbf{R}_V} r_{V,k}^{(i)} \leq 1$.

■

These models are called "canonical causal models" in [68], but for the purposes of this proof, we will call them $\mathcal{G}$-canonical SCMs to emphasize that their construction is dependent on a causal diagram $\mathcal{G}$. This model class can be used to represent every equivalence class of SCMs out of the ones that induce $\mathcal{G}$, at least on the L_2-level, as evident from the next proposition.

Fact 3 ([68, Lem. 4])**.** *For any SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-canonical SCM $\mathcal{M}_{GCM} = \langle \mathbf{U}_{GCM}, \mathbf{V}, \mathcal{F}_{GCM}, P(\mathbf{U}_{GCM}) \rangle$ such that $\mathcal{M}_{GCM}$ is L_2-consistent with $\mathcal{M}$.* ■

With this result, we can proceed to prove Theorem 3 by building an equivalent $\mathcal{G}$-constrained NCM for every $\mathcal{G}$-canonical SCM.

Theorem 3 (L_2-$\mathcal{G}$ Representation)**.** *For any SCM $\mathcal{M}^*$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-constrained NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ that is L_2-consistent w.r.t. $\mathcal{M}^*$.* ■

Proof. Fact 3 states that there exists a $\mathcal{G}$-canonical SCM $\mathcal{M}_{\mathsf{GCM}} = \langle \mathbf{U}_{\mathsf{GCM}}, \mathbf{V}, \mathcal{F}_{\mathsf{GCM}}, P(\mathbf{U}_{\mathsf{GCM}}) \rangle$ that is L_2-consistent with $\mathcal{M}^*$. Hence, to construct $\widehat{M}$, we can simply show how to construct $\mathcal{M}_{\mathsf{GCM}}$ using the architecture of an NCM.

Following Def. 7, we choose $\widehat{\mathbf{U}} = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G})\}$, where $\mathbb{C}(\mathcal{G})$ is the set of all C^2-components of $\mathcal{G}$. Denote $\widehat{\mathbf{U}}_V = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G}) \text{ s.t. } V \in \mathbf{C}\}$. For each $V_i \in \mathbf{V}$, we construct $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ using the following components:

1. By Lemma 5, construct $\hat{f}^R_{V_i} : \mathcal{D}_{\widehat{\mathbf{U}}_{V_i}} \to \mathcal{D}_{\mathbf{R}_{V_i}}$ such that
$$\hat{f}^R_{V_i}(\widehat{\mathbf{u}}_{V_i}) = \mathbf{r}_{V_i}, \tag{19}$$
where
$$P^{\widehat{M}}\left(\hat{f}^R_{V_i}(\widehat{\mathbf{U}}_{V_i}) = \mathbf{r}_{V_i}\right) = P^{\mathcal{M}_{\mathsf{CM}}}(\mathbf{r}_{V_i}). \tag{20}$$
Here, $\mathbf{R}_V$ is defined as from Def. 16.

2. By Lemma 4, construct $\hat{f}^H_{V_i} : \mathcal{D}_{\mathbf{Pa}_{V_i}} \times \mathcal{D}_{\mathbf{R}_{V_i}} \to \mathcal{D}_{V_i}$ such that
$$\hat{f}^H_{V_i}(\mathbf{pa}_{V_i}, \mathbf{r}_{V_i}) \tag{21}$$
$$= f^{\mathsf{GCM}}_{V_i}(\mathbf{pa}_{V_i}, \mathbf{r}_{V_i}) \tag{22}$$
$$= \sum_{k \in \mathbb{N}} h_{V_i}^{(j_{V_i}^{(k)})}(\mathbf{pa}_{V_i}) \prod_{\mathbf{R}_{\mathbf{C}} \in \mathbf{R}_{V_i}} r^{(k)}_{V_i, \mathbf{C}}. \tag{23}$$

Combining these two components leads to the MLP,
$$\hat{f}_{V_i}(\mathbf{pa}_{V_i}, \widehat{\mathbf{u}}_{V_i}) = \hat{f}^H_{V_i}\left(\mathbf{pa}_{V_i}, \hat{f}^R_{V_i}(\widehat{\mathbf{u}}_{V_i})\right). \tag{24}$$

By Eqs. 20 and 23, the NCM $\widehat{M}$ is constructed to match $\mathcal{M}_{\mathsf{CM}}$ on all outputs. Hence, $\widehat{M}$ must be L_2-consistent with $\mathcal{M}^*$. □

A.4 Proofs of Theorem 4 and Corollaries 2 and 4

Theorem 4 (Graphical-Neural Equivalence (Dual ID))**.** *Let Ω^* be the set of all SCMs and Ω the set of NCMs. Consider the true SCM $\mathcal{M}^*$ and the corresponding causal diagram $\mathcal{G}$. Let $Q = P(\mathbf{y} \mid do(\mathbf{x}))$ be the query of interest and $P(\mathbf{v})$ the observational distribution. Then, Q is neural identifiable from $\Omega(\mathcal{G})$ and $P(\mathbf{v})$ if and only if it is identifiable from $\mathcal{G}$ and $P(\mathbf{v})$.* ■

Proof. Since $\Omega \subset \Omega^*$, it is trivially the case that identifiability in $\mathcal{G}$ and Ω^* implies identifiability in $\mathcal{G}$ and Ω. If $P^{\mathcal{M}_1}(\mathbf{y} \mid do(\mathbf{x})) = P^{\mathcal{M}_2}(\mathbf{y} \mid do(\mathbf{x}))$ for every pair of models $\mathcal{M}_1, \mathcal{M}_2 \in \Omega^*$ that both induce $\mathcal{G}$ with $P^{\mathcal{M}_1}(\mathbf{v}) = P^{\mathcal{M}_2}(\mathbf{v}) > 0$, then it must also hold if $\mathcal{M}_2 \in \Omega$.

If Q is not identifiable from $\mathcal{G}$ and Ω^*, then there must exist $\mathcal{M}_1, \mathcal{M}_2 \in \Omega^*$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ both induce $\mathcal{G}$, $L_1(\mathcal{M}_1) = L_1(\mathcal{M}_2) > 0$, but $P^{\mathcal{M}_1}(\mathbf{Y} \mid do(\mathbf{X})) \neq P^{\mathcal{M}_2}(\mathbf{Y} \mid do(\mathbf{X}))$. Theorem 3 states that there must exist NCM $\widehat{\mathcal{M}_2} \in \Omega$ such that $\widehat{\mathcal{M}_2}$ induces $\mathcal{G}$ and $L_2(\widehat{\mathcal{M}_2}) = L_2(\mathcal{M}_2)$. This implies that $L_1(\widehat{\mathcal{M}_2}) = L_1(\mathcal{M}_2) = L_1(\mathcal{M}_1) > 0$ and $P^{\widehat{\mathcal{M}_2}}(\mathbf{Y} \mid do(\mathbf{X})) = P^{\mathcal{M}_2}(\mathbf{Y} \mid do(\mathbf{X})) \neq P^{\mathcal{M}_1}(\mathbf{Y} \mid do(\mathbf{X}))$. This means that if Q is not identifiable from $\mathcal{G}$ and Ω^*, then there exists $\mathcal{M}_1 \in \Omega^*$ and $\widehat{\mathcal{M}_2} \in \Omega$ such that $\mathcal{M}_1$ and $\widehat{\mathcal{M}_2}$ both induce $\mathcal{G}$, and $P^{\mathcal{M}^*}(\mathbf{v}) = P^{\widehat{M}}(\mathbf{v}) > 0$, but $P^{\widehat{\mathcal{M}_2}}(\mathbf{Y} \mid do(\mathbf{X})) \neq P^{\mathcal{M}_1}(\mathbf{Y} \mid do(\mathbf{X}))$. In other words, if Q is not identifiable from $\mathcal{G}$ and Ω^*, then it is also not identifiable from $\mathcal{G}$ and Ω. □

Corollary 2 (Neural Mutilation (Operational ID))**.** *Consider the true SCM $\mathcal{M}^* \in \Omega^*$, causal diagram $\mathcal{G}$, the observational distribution $P(\mathbf{v})$, and a target query Q equal to $P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$. Let $\widehat{\mathcal{M}} \in \Omega(\mathcal{G})$ be a $\mathcal{G}$-constrained NCM that is L_1-consistent with $\mathcal{M}^*$. If the effect is identifiable from $\mathcal{G}$ and $P(\mathbf{v})$, then Q is computable through a mutilation process on a proxy NCM $\widehat{\mathcal{M}}$, i.e., for each $X \in \mathbf{X}$, replacing the equation f_x with a constant x $\;(Q = \textsc{proc-mutilation}(\widehat{M}; \mathbf{X}, \mathbf{Y}))$.* ■

Proof. We can compute $P^{\widehat{M}}(\mathbf{Y} = \mathbf{y} \mid do(\mathbf{X} = \mathbf{x}))$ from $\widehat{M}$ using Def. 2. By Def. 8, since both $\mathcal{M}^*$ and $\widehat{M}$ induce $\mathcal{G}$ and $L_1(\widehat{M}_1) = L_1(\mathcal{M}^*) > 0$, we have $P^{\mathcal{M}^*}(\mathbf{Y} \mid do(\mathbf{X})) = P^{\widehat{M}}(\mathbf{Y} \mid do(\mathbf{X}))$. □

Corollary 4 (Soundness and Completeness)**.** *Let Ω^* be the set of all SCMs, $\mathcal{M}^* \in \Omega^*$ be the true SCM inducing causal diagram $\mathcal{G}$, $Q = P(\mathbf{y} \mid do(\mathbf{x}))$ be a query of interest, and $\widehat{Q}$ be the result from running Alg. 1 with inputs $P^*(\mathbf{v}) = L_1(\mathcal{M}^*) > 0$, $\mathcal{G}$, and Q. Then Q is identifiable from $\mathcal{G}$ and $P^*(\mathbf{v})$ if and only if $\widehat{Q}$ is not* `FAIL`*. Moreover, if $\widehat{Q}$ is not* `FAIL`*, then $\widehat{Q} = P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$.* ■

Proof. Theorem 4 states that Q must be identifiable from $\mathcal{G}$ and Ω^* if and only if for all pairs of $\mathcal{G}$-consistent NCMs, $\widehat{M}_1, \widehat{M}_2 \in \Omega$ with $L_1(\widehat{M}_1) = L_1(\widehat{M}_2) > 0$, $P^{\widehat{M}_1}(\mathbf{Y} \mid do(\mathbf{X})) = P^{\widehat{M}_2}(\mathbf{Y} \mid do(\mathbf{X}))$. This holds if and only if $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{Y} \mid do(\mathbf{X})) = P^{\widehat{M}(\boldsymbol{\theta}^*_{\max})}(\mathbf{Y} \mid do(\mathbf{X}))$. If they are not equal, then $\widehat{M}(\boldsymbol{\theta}^*_{\min})$ and $\widehat{M}(\boldsymbol{\theta}^*_{\max})$ are a counterexample of two NCMs that do not match for Q. Otherwise, if they are equal, then all other NCMs must also induce the same answer for Q. A result for Q that is less than $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{Y} \mid do(\mathbf{X}))$ or greater than $P^{\widehat{M}(\boldsymbol{\theta}^*_{\max})}(\mathbf{Y} \mid do(\mathbf{X}))$ would contradict the optimality of $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$.

If Q is identifiable, then Corollary 2 guarantees that any NCM that induces $P^*(\mathbf{V})$ and $\mathcal{G}$ will induce the correct $P^{\mathcal{M}^*}(\mathbf{Y} \mid do(\mathbf{X}))$. □

A.5 Proof of Corollary 3

In our discussion of the identification problem in Sec. 3, we stated Corol. 3 as the solution to a special class of models known as Markovian. In SCMs, Markovianity implies that all variables in $\mathbf{U}$ are independent and not shared between functions. Correspondingly, this means that no variable in $\widehat{\mathbf{U}}$ of an NCM can be associated with more than one function. In the causal diagram, this implies that there are no bidirected edges.

We emphasize that Markovianity is a strong constraint in many settings, and the following corollary from [5] illustrates that identification in the Markovian setting is quite trivial.

Fact 4 ([5, Corol. 2])**.** *In Markovian models (i.e., models without unobserved confounding), for any treatment $\mathbf{X}$ and outcome $\mathbf{Y}$, the interventional distribution $P(\mathbf{Y} \mid do(\mathbf{x}))$ is always identifiable and given by the expression*

$$P(\mathbf{Y} \mid do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{Y} \mid \mathbf{x}, \mathbf{z}) P(\mathbf{z}), \tag{25}$$

where $\mathbf{Z}$ is the set of all variables not affected by the action $\mathbf{X}$ (non-descendants of $\mathbf{X}$). ■

In other words, every query in a Markovian setting can be identified via Eq. 25, also known as the backdoor adjustment formula. Naturally, this result extends to identifiability via mutilation in NCMs due to the connection between neural identification and graph identification.

Corollary 3 (Markovian Identification)**.** *Whenever the $\mathcal{G}$-constrained NCM $\widehat{M}$ is Markovian, $P(\mathbf{y} \mid do(\mathbf{x}))$ is always identifiable through the process of mutilation in the proxy NCM (via Corol. 2).* ■

Proof. Fact 4 shows that any query $P(\mathbf{y} \mid do(\mathbf{x}))$ is identifiable from $\mathcal{G}$ and $P(\mathbf{v})$ in Markovian settings. Hence, by Thm. 4, it must also be identifiable from $\Omega(\mathcal{G})$ and $P(\mathbf{v})$. Moreover, Corol. 2 states that the effect can be computed via the mutilation procedure on a proxy $\mathcal{G}$-constrained NCM $\widehat{M}$ with $L_1(\widehat{M}) = P(\mathbf{v})$. □

Without the Markovianity assumption, the identification problem becomes significantly more challenging. As we show in Example 2 from Appendix C.1, a query can be non-ID even in a two variable case whenever unobserved confounding cannot be ruled out.

A.6 Derivation of Results in Section 4

Recall the description of our choice of NCM architecture from Eq. 2:

$$\begin{cases} \mathbf{V} & := \mathbf{V},\ \widehat{\mathbf{U}} := \{U_{\mathbf{C}} : \mathbf{C} \in C^2(\mathcal{G})\} \cup \{G_{V_i} : V_i \in \mathbf{V}\}, \\ \widehat{\mathcal{F}} & := \left\{ f_{V_i} := \arg\max_{j\in\{0,1\}} g_{j,V_i} + \begin{cases} \log\sigma(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i})) & j=1 \\ \log(1-\sigma(\phi_{V_i}(\mathbf{pa}_{V_i},\mathbf{u}^c_{V_i};\theta_{V_i}))) & j=0\end{cases} \right\}, \\ P(\widehat{\mathbf{U}}) & := \{U_{\mathbf{C}} \sim \text{Unif}(0,1) : U_{\mathbf{C}} \in \mathbf{U}\} \cup \\ & \quad \{G_{j,V_i} \sim \text{Gumbel}(0,1) : V_i \in \mathbf{V}, j \in \{0,1\}\}, \end{cases} \tag{2}$$

Using this architecture, we show the derivation of Eq. 3, starting with sampling from $P(\mathbf{v})$. Let $\mathbf{U}^c$ and $\mathbf{G}$ denote the latent C^2-component variables and Gumbel random variables [24], respectively. The formulation in Eq. 2 allows us to compute the probability mass of a datapoint $\mathbf{v}$ as:

$$\begin{aligned} & P^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v}) \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u})}[\mathbb{1}[\mathcal{F}(\mathbf{u}) = \mathbf{v}]] \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u}^c)P(\mathbf{g})}\left[\prod_{V_i \in \mathbf{V}} \mathbb{1}\left[f_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}_{V_i}) = v_i\right]\right] \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u}^c)}\left[\prod_{V_i \in \mathbf{V}} \tilde{\sigma}_{v_i}(\phi_i(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i}))\right], \end{aligned} \tag{26}$$

where $\tilde{\sigma}_{v_i}(x) := \begin{cases} \sigma(x) & v_i = 1 \\ 1-\sigma(x) & v_i = 0 \end{cases}$. We can then derive a Monte Carlo estimator given samples $\{\mathbf{u}^c_j\}_{j=1}^m \sim P(\mathbf{U}^c)$:

$$\hat{P}_m^{\widehat{M}(\mathcal{G};\boldsymbol{\theta})}(\mathbf{v}) = \frac{1}{m}\sum_{j=1}^{m}\prod_{V_i \in \mathbf{V}} \tilde{\sigma}_{v_i}(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{j,V_i}; \theta_{V_i})). \tag{27}$$

One may similarly estimate the interventional query, $P(\mathbf{y}|do(\mathbf{x}))$, where $\mathbf{y}, \mathbf{x}$ are the values of the variable sets $\mathbf{Y}, \mathbf{X} \subseteq \mathbf{V}$. We first compute an estimable expression for $P^{\widehat{M}}(\mathbf{y}|do(\mathbf{x}))$:

$$\begin{aligned} & P^{\widehat{M}(\mathcal{G},\boldsymbol{\theta})}(\mathbf{y}|do(\mathbf{x})) \\ & = \sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} P^{\widehat{M}}(\mathbf{v}|do(\mathbf{x})) \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u})}\left[\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} \mathbb{1}\left[\mathcal{F}_{\mathbf{x}}(\mathbf{u}) = \mathbf{v}\right]\right] \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u}^c)P(\mathbf{g})}\left[\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} \prod_{V_i \in \mathbf{V}\setminus\mathbf{X}} \mathbb{1}\left[f_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}_{V_i}) = v_i\right]\right] \\ & = \mathop{\mathbb{E}}_{P(\mathbf{u}^c)}\left[\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})} \prod_{V_i \in \mathbf{V}\setminus\mathbf{X}} \tilde{\sigma}_{v_i}(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i}))\right]. \end{aligned} \tag{28}$$

The interventional distribution may be similarly estimated. We derive a Monte Carlo estimator for the above expression using a submodel of the NCM under $do(\mathbf{X} = \mathbf{x})$ given samples $\{\mathbf{u}^c_j\}_{j=1}^m \sim P(\mathbf{U}^c)$:

$$\begin{aligned} & P_m^{\widehat{M}(\mathcal{G},\boldsymbol{\theta})}(\mathbf{y}|do(\mathbf{x})) \\ & = \frac{1}{m}\sum_{j=1}^{m}\sum_{\mathbf{V}\setminus(\mathbf{Y}\cup\mathbf{X})}\prod_{V_i \in \mathbf{V}\setminus\mathbf{X}} \tilde{\sigma}_{v_i}(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{j,V_i}; \theta_{V_i})). \end{aligned} \tag{29}$$

The implementation of Eq. 3 can be summarized in Alg. 3, which defines the “Estimate” function used in Alg. 2.

Algorithm 3: Estimate $P^{\widehat{M}}(\mathbf{v} \mid do(\mathbf{x}))$ (Eq. 3)

```
Input  : NCM $\widehat{M}$, variables $\mathbf{V}$ in topological order, $\mathbf{v} \in \mathcal{D}_{\mathbf{V}}$, intervention set $\mathbf{X} \subset \mathbf{V}$, $\mathbf{x} \in \mathcal{D}_{\mathbf{X}}$,
         number of Monte Carlo samples $m$
Output : estimate of $P^{\widehat{M}}(\mathbf{v} \mid do(\mathbf{x}))$
if not Consistent(v, x) then return $0$
$\widehat{\mathbf{u}}^c_{1:m} \leftarrow$ Sample($P(\widehat{\mathbf{U}}^c)$)
$\hat{p} \leftarrow 0$
for $j \leftarrow 1$ to $m$ do
    $\hat{p}_j \leftarrow 1$
    for $i \leftarrow 1$ to $|\mathbf{V}|$ do
        if $V_i \notin \mathbf{X}$ then
            $\hat{p}_j \leftarrow \hat{p}_j \sigma'_{v_i}(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{j,V_i}; \theta_{V_i}))$        // From Eq. 3
    $\hat{p} \leftarrow \hat{p} + \hat{p}_j$
return $\frac{\hat{p}}{m}$
```

B Experimental Details

This section provides the detailed information about our experiments. Our models are primarily written in PyTorch [52], and training is facilitated using PyTorch Lightning [16].

B.1 Data Generation Process

The NCM architecture follows the description in Eq. 2. For each function, we use a MADE module [18] following the implementation in [35] (MIT license). Each MADE module uses 4 hidden layers of size 32 with ReLU activations. For each complete confounded component's unobserved confounder, we use uniform noise with dimension equal to the sum of the complete confounded component's variables' dimensions.

To generate data for our experiments, we use a version of the canonical SCM similar to Def. 10. Given data from $P(\mathbf{V})$ and a graph $\mathcal{G}$, we construct a canonical SCM $\mathcal{M}_{\mathsf{CM}} = \langle \mathbf{U}_{\mathsf{CM}}, \mathbf{V}, \mathcal{F}_{\mathsf{CM}}, P(\mathbf{U}_{\mathsf{CM}}) \rangle$ such that for each $V \in \mathbf{V}$, $\mathbf{pa}_V$ contains the set of variables with a directed edge into V in $\mathcal{G}$. For any bidirected edge between V_1 and V_2, we introduce a dependency between the noise generated from R_{V_1} and R_{V_1}. Otherwise, variables in $\mathbf{U}_{\mathsf{CM}}$ are independent.

The canonical SCM is implemented in PyTorch, with trainable parameters that determine $P(\mathbf{U}_{\mathsf{CM}})$. In an experiment trial, we create true model $\mathcal{M}^*$, which is an instantiation of a canonical SCM with random parameters. Unlike cases where $\mathcal{M}^*$ comes from a set family of SCMs, which may produce biased results, the expressiveness of the canonical SCM allows us to robustly sample data generating models that can take any behavior consistent with the constraints of $\mathcal{G}$.

In the estimation experiments, to make things more challenging, we take one additional step after constructing $\mathcal{M}^*$ in cases where $P(Y \mid do(X))$ is not necessarily equal to $P(Y \mid X)$. To ensure that this inequality holds, we widen the difference between ATE and total variation (TV), computed in this case as $P(Y = 1 \mid X = 1) - P(Y = 1 \mid X = 0)$. We accomplish this by performing gradient descent on the parameters of the canonical SCM until this difference is at least 0.05.

From the perspective of simulating the ground truth, sampling from the canonical SCM can be accomplished using the same procedure from Def. 2.

B.2 Identification Running Procedure

For the identification experiments, we test on the 8 graphs shown in the top of Fig. 4. For each graph, we run 20 trials, each repeated 4 times on the same dataset for the sake of hypothesis testing.

For each trial, we create a canonical SCM from the graph $\mathcal{G}$ for the data generating model $\mathcal{M}^*$. We sample 10,000 data points from $P(\mathbf{V})$, where $\mathbf{V}$ is determined by the nodes in $\mathcal{G}$, each being a binary variable.

With $\mathcal{G}$ as input, we instantiate two NCMs, $\widehat{M}_{\min}$ and $\widehat{M}_{\max}$, and train them on the given data using a PyTorch Lightning trainer. Both models are trained for 3000 epochs using the minimization and maximization version of the objective in Eq. 5, respectively. λ is initialized to 1 at the beginning of training and decreases logarithmically throughout training until it reaches 0.001 at the end.

We use the AdamW optimizer [48] with the Cosine Annealing Warm Restarts learning rate scheduler [47]. When using Eq. 3 to compute probabilities, we choose $m = 20000$.

We log the ATE of both models every 10 iterations and use it to compute the max-min gaps from the l.h.s. of Eq. 6. The gaps from each trial are averaged across the 4 runs. The percentiles of these gaps over the 20 trials are plotted across the 3000 epochs in the bottom row of Fig. 4. We use these gaps in the hypothesis testing procedure described later in Sec. B.4 with $\tau = 0.01, 0.03, 0.05$ and plot the accuracy over the 20 trials in the middle row of Fig. 4. For smoother plots, we use the running average over 50 iterations.

To ensure reproducibility, each run's random seed is generated through a SHA512 encoding of a key. The key is determined by the parameters of the trial such as the graph, the number of samples, and the trial index.

The NCMs are trained on NVIDIA Tesla V100 GPUs provided by Amazon Web Services. In total, we used about 150 GPU hours for the identification experiments.

B.3 Estimation Running Procedure

For the estimation experiments, we test on the 4 identifiable graphs at the top of Fig. 4. For each graph, we test 15 different amounts of sample data, ranging from 10^3 to 10^6 on a logarithmic scale. Each setting is run for 25 trials.

For each trial, we create a canonical SCM from the graph $\mathcal{G}$ for the data generating model $\mathcal{M}^*$. We sample the data from $P(\mathbf{V})$, where $\mathbf{V}$ is determined by the nodes in $\mathcal{G}$, each being a binary variable.

We instantiate an NCM, $\widehat{M}$, given $\mathcal{G}$, and we train it on the given data using a PyTorch Lightning trainer. The model is trained using the objective in Eq. 4. We incorporate early stopping, where the model ends training if loss does not decrease by a minimum of 10^{-6} within 100 epochs. After training, the parameters of the best model encountered during training are reloaded.

We use the AdamW optimizer [48] with the Cosine Annealing Warm Restarts learning rate scheduler [47]. When using Eq. 3 to compute probabilities, we choose $m = 1.6 \times 10^6$.

For each trial, we save the training data and perform the same estimation experiment with the naïve likelihood maximization model and WERM [32]. The naïve model is also optimized with the objective in Eq. 4 to fit $P(\mathbf{V})$. Since code for WERM is not available, we implement our own version in Python trying to match the code provided in the paper. Notably, we use XGBoost [12] regressors trained on the data to learn the WERM weights. These regressors are chosen with 20 estimators and a max depth of 10. They are trained with the binary logistic objective for binary outputs, otherwise they are trained using the squared error objective. The regularization parameter λ_{WERM} (not to be confused with the λ in Eq. 5) is learned through hyperparameter tuning, checking every value from 0 to 10 in increments of 0.2. The regularization parameter α_{WERM} is set to $\lambda_{\text{WERM}}/2$.

We record the KL divergence of the best models of the NCM and naïve models after training, computed as

$$D_{\text{KL}}\left(P^{\mathcal{M}^*}(\mathbf{V})||P^{\widehat{M}}(\mathbf{V})\right) = \sum_{\mathbf{v} \in \mathcal{D}_{\mathbf{V}}} P^{\mathcal{M}^*}(\mathbf{v}) \log\left(\frac{P^{\mathcal{M}^*}(\mathbf{v})}{P^{\widehat{M}}(\mathbf{v})}\right).$$

The mean and 95%-confidence intervals of the KL divergence across the 25 trials are plotted over each setting of sample size in the top row of Fig. 5.

We also record the ATE computed from all three methods after training. In the case of the naïve model, we use TV as the ATE calculation. For the NCM and naïve model, we use the sampling procedure described in Eq. 3 with $m = 10^6$ to compute these results. The mean and 95%-confidence

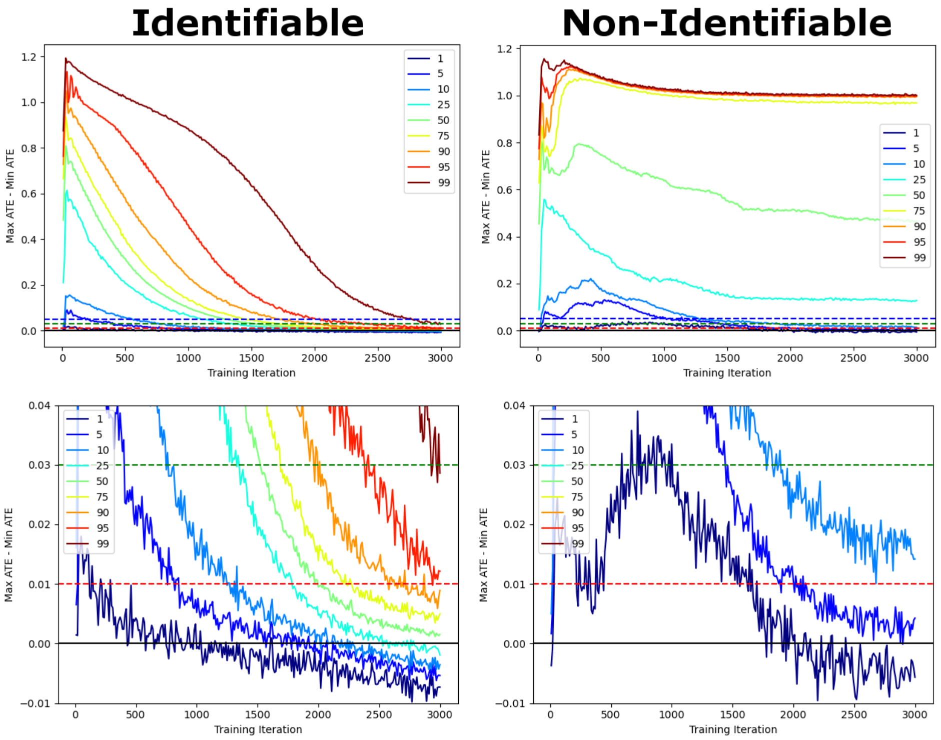


Figure 7: Example plots aggregating max-min gaps across multiple settings. Left is ID, right is nonID. Bottom plots are zoomed versions of top plots. Dashed lines represent choices of τ at 0.01 (red), 0.03 (green) and 0.05 (blue).

intervals of the ATE computations across the 25 trials are plotted over each setting of sample size in the bottom row of Fig. 5.

Randomization is performed similarly to the identification experiments.

The NCMs and naïve models are trained on NVIDIA Tesla V100 GPUs provided by Amazon Web Services. In total, we used about 300 GPU hours for the estimation experiments.

B.4 Identification Hypothesis Testing

We incorporate a basic hypothesis testing procedure to test the inequality in Eq. 6. In each trial of the identification experiments, we rerun the procedure on the same dataset r times ($r = 4$ in our experiments). Let $\Delta\widehat{Q}$ denote the random variable representing the max-min gap from a run in a specific trial, and let $\{\Delta\hat{q}_i\}_{i=1}^{r}$ denote the empirical set of max-min gaps from all r runs. The randomness in $\Delta\widehat{Q}$ arises from the randomness of parameter initialization in the NCM. $\Delta\widehat{Q}$ is not necessarily normally distributed, but we assume that the mean of $\{\Delta\hat{q}_i\}_{i=1}^{r}$ will be normally distributed given a large enough r due to the central limit theorem.

For a given τ, we test the hypothesis that $\mathbb{E}[\Delta\widehat{Q}] < \tau$ with the null hypothesis that $\mathbb{E}[\Delta\widehat{Q}] \geq \tau$. We estimate

$$\mathbb{E}[\Delta\widehat{Q}] \approx \overline{\Delta\hat{q}} := \frac{1}{r}\sum_{i=1}^{r} \Delta\hat{q}_i,$$

the mean of the empirical set of gaps. Then we compute the standard error

$$\mathrm{SE}(\Delta\hat{q}) := \frac{1}{r}\sqrt{\sum_{i=1}^{r}\left(\Delta\hat{q}_i - \overline{\Delta\hat{q}}\right)}.$$

Finally, we check

$$\overline{\Delta\hat{q}} + 1.65\,\mathrm{SE}(\Delta\hat{q}) < \tau.$$

If this quantity holds, we reject the null hypothesis with 95% confidence, and we return the result that the query is identifiable. If not, we fail to reject the null hypothesis and we return the result that the query is not identifiable.

We showed the results of three different values of τ $(0.01, 0.03, 0.05)$ in Fig. 4 (main paper). Here, we try to understand how τ can be determined more systematically by running a preliminary round of identification experiments and observing the percentiles of the resulting gaps; see Fig. 7. Note that most gaps in the ID case fall below the 0.03 line by the end of training process, while most gaps in the nonID case stay above it. This can be used to motivate a particular choice of τ.

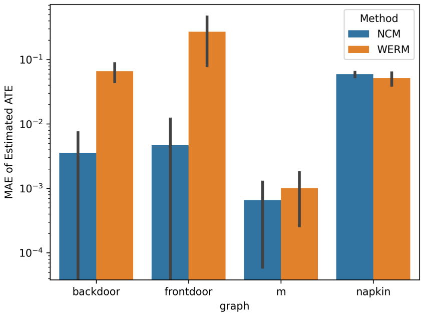


Figure 8: NCM estimation results for ID graphs when setting the number of observed samples to 1000000 and the number of dimensions for each multi-dimensional binary covariate (any variable which is not X or Y) to 20. The graphs displayed in the plot correspond to the same graphs as a, b, c, d in Fig. 4. Plot in log scale.

B.5 Additional Results

To demonstrate the potential of NCMs to scale to larger settings, we provide additional estimation results on higher dimensional data in Fig. 8. We use the same canonical SCM architecture but probabilistically map the value of each binary variable which is not X or Y to a 20-dimensional binary vector, such that there exists a deterministic mapping from each covariate's 20-dimensional binary vectors to the original binary value (that is, the original binary value is recoverable from the 20-dimensional mapped vector). This preserves ATE and TV while increasing the difficulty of the NCM's optimization problem by testing the NCM's ability to learn in high-dimensional spaces. In this more challenging high-dimensional setting, the NCM maintains superior or equal performance when compared to WERM.

B.6 Other Possible Architectures

In Sec. 4, we provided one possible architecture of the NCM (Eq. 2), which produced the empirical results discussed in Sec. 5. These results show that, in principle, training a proxy model can be used to solve the identification and estimation problems in practice.

Still, other architectures may be more suitable and perform more efficiently than the one shown in Eq. 2 in larger, more involved domains. In such settings, it may be challenging to match the distributions well enough to reach the correct identification result, or to lower the error on estimating the query of interest. While Corol. 4 guarantees the correct result with perfect optimization, this is rarely achieved in practice. It's certainly not fully understood how optimization errors on the learning of $P(\mathbf{V})$ will affect the performance of identification/estimation of Layer 2 quantities.

Naturally, the framework developed here to support proxy SCMs is not limited to the architecture in Eq. 2. For instance, as discussed in Footnote 10, there are alternative methods of learning $P(\mathbf{V})$ such as by minimizing divergences, performing variational inference, generative adversarial optimization, to cite a few prominent choices. Some of these methods may be more scalable and robust to optimization errors. See Appendix D.1 for a discussion on how the theory generalizes for other architectures.

Further, we note that if $\mathbf{V}$ comes from continuous domains, there are many ways the architecture of the NCM could be augmented accordingly. As discussed in Footnote 11, one could replace the Gumbel-max trick in the architecture in Eq. 2 with a model that directly computes a probability density given a data point. Some of the alternative architectures mentioned above may work as well.

We believe this leads to an exciting research agenda that is to understand the tradeoffs involved in the choice of the architectures and how to properly optimize NCMs for specific applications.

C Examples and Further Discussion

In this section, we complement the discussion provided in the main paper through examples. Specifically, in Sec. C.1, we discuss what expressiveness means and provide examples to show why it's insufficient to perform cross-layer inferences. In Sec. C.2, we provide an intuitive discussion on why causal diagrams are able to perform cross-layer inferences and basic examples illustrating the structural constraints encoded in such models, which can be seen as an inductive bias as discussed in the body of the paper. In Sec. C.3, we exemplify how this inductive bias can be added to NCMs such that they are capable of solving ID instances. In Sec. C.4, we discuss possible reasons why one would prefer to perform ID through neural computation instead the machinery of the do-calculus.

C.1 Expressiveness Examples

In this section, we try to clarify through examples the notion of expressiveness, as shown in Thm. 1, and discuss the reasons it alone is not sufficient for performing cross-layer inferences.

Example 1. Consider a study on the effects of diet (D) on blood pressure (B), inspired by studies such as [2]. D and B are binary variables such that $D = 1$ represents a high-vegetable diet, and $B = 1$ represents high blood pressure. Suppose the true relationship between B and D is modeled by the SCM $\mathcal{M}^* = \{\mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U})\}$, where

$$\mathcal{M}^* := \begin{cases} \mathbf{U} & := \{U_D, U_{DB}, U_{B1}, U_{B2}\}, \text{ all binary} \\ \mathbf{V} & := \{D, B\} \\ \mathcal{F} & := \begin{cases} f_D(U_D, U_{DB}) & = \neg U_D \wedge \neg U_{DB} \\ f_B(D, U_{B1}, U_{B2}) & = ((\neg D \oplus U_{B1}) \vee U_{DB}) \oplus U_{B2} \end{cases} \\ P(\mathbf{U}) & := \begin{cases} P(U_D = 1) = P(U_{DB} = 1) = P(U_{B1} = 1) = P(U_{B2} = 1) = \frac{1}{4} \\ \text{all } U \in \mathbf{U} \text{ are independent} \end{cases} \end{cases} \tag{30}$$

The set of exogenous variables $\mathbf{U}$ affects the set of endogenous variables $\mathbf{V}$. For example, U_{DB} is an unobserved confounding variable that affects both diet and blood pressure (ethnicity, for example), while the remaining variables in $\mathbf{U}$ represent other factors unaccounted for in the study. $\mathcal{F}$ describes the relationship between the variables in $\mathbf{V}$ and $\mathbf{U}$, and $P(\mathbf{U})$ describes the distribution of $\mathbf{U}$.

U_D	U_{DB}	U_{B1}	U_{B2}	D	B	$P(\mathbf{U})$
0	0	0	0	1	$\neg D$	$p_0 = 81/256$
0	0	0	1	1	D	$p_1 = 27/256$
0	0	1	0	1	D	$p_2 = 27/256$
0	0	1	1	1	$\neg D$	$p_3 = 9/256$
0	1	0	0	0	1	$p_4 = 27/256$
0	1	0	1	0	0	$p_5 = 9/256$
0	1	1	0	0	1	$p_6 = 9/256$
0	1	1	1	0	0	$p_7 = 3/256$
1	0	0	0	0	$\neg D$	$p_8 = 27/256$
1	0	0	1	0	D	$p_9 = 9/256$
1	0	1	0	0	D	$p_{10} = 9/256$
1	0	1	1	0	$\neg D$	$p_{11} = 3/256$
1	1	0	0	0	1	$p_{12} = 9/256$
1	1	0	1	0	0	$p_{13} = 3/256$
1	1	1	0	0	1	$p_{14} = 3/256$
1	1	1	1	0	0	$p_{15} = 1/256$

Table 1: Truth table showing induced values of $\mathcal{M}^*$ for Example 1. Probabilites in $P(\mathbf{U})$ are labeled from p_0 to p_{15} for convenience.

This SCM $\mathcal{M}^*$ induces three collections of distributions corresponding to the layers of PCH with respect to $\mathbf{V}$. These can be summarized in Table 1. For example, a layer 1 quantity such as $P(B = 1 \mid D = 1)$, the probability of someone having high blood pressure given a high-vegetable diet, can be computed as

$$\begin{aligned} P(B=1 \mid D=1) &= \frac{P(D=1, B=1)}{P(D=1)} && (31) \\ &= \frac{p_1+p_2}{p_0+p_1+p_2+p_3} = \frac{54}{144} = 0.375. && (32) \end{aligned}$$

A layer 2 quantity such as $P(B=1 \mid do(D=1))$, the probability from $\mathcal{M}^*$ of someone having high blood pressure when intervened on a high-vegetable diet, can be computed as

$$\begin{aligned} P(B=1 \mid do(D=1)) &= p_1+p_2+p_4+p_6+p_9+p_{10}+p_{12}+p_{14} && (33) \\ &= \frac{120}{256} = 0.46875. && (34) \end{aligned}$$

Layer 3 quantities can also be computed, for example, $P(B_{D=1}=1 \mid D=0, B=1)$. Given that an individual has high blood pressure and a low-vegetable diet, this quantity represents the probability that they would have high-blood pressure had they instead eaten a high-vegetable diet. This can be computed as

$$\begin{aligned} &P(B_{D=1}=1 \mid D=0, B=1) \\ &= \frac{P(B_{D=1}=1, D=0, B=1)}{P(D=0, B=1)} = \frac{p_4+p_6+p_{12}+p_{14}}{p_4+p_6+p_8+p_{11}+p_{12}+p_{14}} = \frac{48}{78} \approx 0.6154. \qquad (35) \end{aligned}$$

Following the construction in the proof for Thm. 1, we will show the (somewhat tedious) construction of the corresponding NCM that induces the same distributions. First, we build a simple canonical SCM from Def. 10 to match the functionality in $\mathcal{M}^*$. We build $\mathcal{M}_{\mathsf{CM}} = \langle \mathbf{U}_{\mathsf{CM}}, \mathbf{V}, \mathcal{F}_{\mathsf{CM}}, P(\mathbf{U}_{\mathsf{CM}})$ such that

$$\mathcal{M}_{\mathsf{CM}} := \begin{cases} \mathbf{U}_{\mathsf{CM}} & := \{R_D, R_B\}, \mathcal{D}_{R_D} = \{0,1\}, \mathcal{D}_{R_B} = \{0,1,2,3\} \\ \mathbf{V} & := \{D, B\} \\ \mathcal{F}_{\mathsf{CM}} & := \begin{cases} f_D^{\mathsf{CM}}(R_D) & = R_D \\ f_B^{\mathsf{CM}}(D, R_B) & = \begin{cases} 0 & \text{if } R_B = 0 \\ D & \text{if } R_B = 1 \\ \neg D & \text{if } R_B = 2 \\ 1 & \text{if } R_B = 3 \end{cases} \end{cases} \end{cases} \qquad (36)$$

Looking at Table 1, we mimic the same distributions of $\mathcal{M}^*$ in $\mathcal{M}_{\mathsf{CM}}$ by choosing

$$P(\mathbf{U}_{\mathsf{CM}}) = P(R_D, R_B) = \begin{cases} p_5+p_7+p_{13}+p_{15} = 16/256 & R_D=0, R_B=0 \\ p_9+p_{10} = 18/256 & R_D=0, R_B=1 \\ p_8+p_{11} = 30/256 & R_D=0, R_B=2 \\ p_4+p_6+p_{12}+p_{14} = 48/256 & R_D=0, R_B=3 \\ p_1+p_2 = 54/256 & R_D=1, R_B=1 \\ p_0+p_3 = 90/256 & R_D=1, R_B=2 \\ 0 & \text{otherwise.} \end{cases} \qquad (37)$$

We state the following useful neural network functions defined in the proof of Thm. 1:

- $\hat{f}_{\text{AND}}(x_1, \dots, x_n) = \sigma\left(\sum_{i=1}^n x_i - n\right)$
- $\hat{f}_{\text{OR}}(x_1, \dots, x_n) = \sigma\left(\sum_{i=1}^n x_i - 1\right)$
- $\hat{f}_{\text{NOT}}(x) = \sigma(-x)$
- $\hat{f}_{\geq z}(x) = \sigma(x - z)$

For this appendix, σ will be the binary step activation function. $\hat{f}_{\text{AND}}$, $\hat{f}_{\text{OR}}$, and $\hat{f}_{\text{NOT}}$ compute the bitwise AND, OR, and NOT of the inputs (if binary) respectively. $\hat{f}_{\geq z}$ will output 1 if its input is greater than or equal to z and will output 0 otherwise.

We begin the construction of our NCM $\widehat{M} = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}})$ as follows:

$$\widehat{M} := \begin{cases} \widehat{\mathbf{U}} & := \{\widehat{U}\}, \mathcal{D}_{\widehat{U}} = [0,1] \\ \mathbf{V} & := \{D, B\} \\ \widehat{\mathcal{F}} & := \begin{cases} \hat{f}_D(\widehat{U}) & = \ ? \\ \hat{f}_B(D, \widehat{U}) & = \ ? \end{cases} \\ P(\widehat{\mathbf{U}}) & := P(\widehat{U}) \sim \text{Unif}(0,1) \end{cases}$$

We must then construct the neural networks in $\widehat{\mathcal{F}}$. With the goal of reproducing the behavior of $\mathcal{M}_{\textsf{CM}}$, we first start by converting a uniform distribution into $P(R_D, R_B)$. We can accomplish this by dividing the $[0,1]$ interval into chunks for each individual value of $(R_D = r_D, R_B = r_B)$ with the size of $P(R_D = r_D, R_B = r_B)$. For instance, considering values of $P(R_D, R_B)$ in the order in Eq. 37, the interval $[0, 16/256)$ would correspond to $(R_D = 0, R_B = 0)$, the interval $[16/256, 34/256)$ would correspond to $(R_D = 0, R_B = 1)$, and so on. In this case, $\widehat{U}$ would be broken into six intervals corresponding to the 6 cases of Eq. 37: $[0, 16/256)$, $[16/256, 34/256)$, $[34/256, 64/256)$, $[64/256, 112/256)$, $[112/256, 166/256)$, and $[166/256, 1)$. For the sake of convenience, we will label these I_1 to I_6 respectively

We can check if U is contained within an interval like $[16/256, 34/256)$ with the following neural network function:

$$\hat{f}_{I_2} = \hat{f}_{[16/256,34/256)}(u) = \hat{f}_{\text{AND}}\left(\hat{f}_{\geq 16/256}(u), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 34/256}(u)\right)\right) \tag{38}$$

Observe that f_{I_1} to f_{I_6} provide a one-hot encoding of the location of $\widehat{U}$. Mapping this encoding to R_D and R_B is simply multiplying the corresponding values of r_D and r_B as weights to the encoding:

$$\hat{f}_{R_D}(u) = 0 \cdot \hat{f}_{I_1}(u) + 0 \cdot \hat{f}_{I_2}(u) + 0 \cdot \hat{f}_{I_3}(u) + 0 \cdot \hat{f}_{I_4}(u) + 1 \cdot \hat{f}_{I_5}(u) + 1 \cdot \hat{f}_{I_6}(u) \tag{39}$$

$$\hat{f}_{R_B}(u) = 0 \cdot \hat{f}_{I_1}(u) + 1 \cdot \hat{f}_{I_2}(u) + 2 \cdot \hat{f}_{I_3}(u) + 3 \cdot \hat{f}_{I_4}(u) + 1 \cdot \hat{f}_{I_5}(u) + 2 \cdot \hat{f}_{I_6}(u) \tag{40}$$

Note that $\hat{f}_{R_D}$ and $\hat{f}_{R_B}$ are indeed neural networks, and this final layer does not contain an activation function.

Since $D = R_D$ in $\mathcal{M}_{\textsf{CM}}$, we can already construct

$$\hat{f}_D(u) = \hat{f}_{R_D}(u). \tag{41}$$

The function $\hat{f}_B$ is somewhat more involved since it must take D as an input. Since R_B is not a binary variable, we first convert it to binary form to simplify the mapping. We can accomplish this by using the same strategy of obtaining the one-hot encoding of R_B. For instance, to check if $R_B = 1$, we can build the neural network function

$$\hat{f}_{=1}(r_B) = \hat{f}_{\text{AND}}\left(\hat{f}_{\geq 1}(r_B), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 2}(r_B)\right)\right). \tag{42}$$

The encoding is formed with $\hat{f}_{=0}$, $\hat{f}_{=1}$, $\hat{f}_{=2}$, and $\hat{f}_{=3}$.

Using this, we can construct $\hat{f}_B(u)$ following the desired properties in Eq. 36:

$$\hat{f}_B(d,u) = \hat{f}_{\text{OR}}\left(\hat{f}_{\text{AND}}\left(\hat{f}_{=1}(r_B), d\right), \hat{f}_{\text{AND}}\left(\hat{f}_{=2}(r_B), \hat{f}_{\text{NOT}}(d)\right), \hat{f}_{=3}(r_B)\right) \tag{43}$$

where $r_B = \hat{f}_{R_B}(u)$. With $\hat{f}_D$ and $\hat{f}_B$ defined, our construction of $\widehat{M}$ is complete.

We can now verify that the distributions induced by $\widehat{M}$, as represented in Table 2, matches the distributions induced by $\mathcal{M}^*$. Consider the same three queries from Eqs. 31, 33, and 35:

$$P^{\widehat{M}}(B = 1 \mid D = 1) = \frac{P^{\widehat{M}}(D = 1, B = 1)}{P^{\widehat{M}}(D = 1)} = \frac{q_4}{q_4 + q_5} = \frac{54}{144} = 0.375$$

$$P^{\widehat{M}}(B = 1 \mid do(D = 1)) = q_1 + q_3 + q_4 = \frac{120}{256} = 0.46875$$

$\widehat{U}$	D	B	$P(\mathbf{U})$
$[0, 16/256)$	0	0	$q_0 = 16/256$
$[16/256, 34/256)$	0	D	$q_1 = 18/256$
$[34/256, 64/256)$	0	$\neg D$	$q_2 = 30/256$
$[64/256, 112/256)$	0	1	$q_3 = 48/256$
$[112/256, 166/256)$	1	D	$q_4 = 54/256$
$[166/256, 1)$	1	$\neg D$	$q_5 = 90/256$

Table 2: Truth table showing induced values of $\widehat{M}$ for Example 1. Probabilites in $P(\mathbf{U})$ are labeled from q_0 to q_5 for convenience.

$$\begin{aligned} & P^{\widehat{M}}(B_{D=1} = 1 \mid D = 0, B = 1) \\ &= \frac{P^{\widehat{M}}(B_{D=1} = 1, D = 0, B = 1)}{P^{\widehat{M}}(D = 0, B = 1)} = \frac{q_3}{q_2 + q_3} = \frac{48}{78} \approx 0.6154 \end{aligned}$$

This demonstrates that the NCM is indeed expressive enough to model this setting on all three layers of PCH. While a more basic model might be fitted to answer L_1 queries (such as $P(B = 1 \mid D = 1)$), a well-parameterized NCM is expressive enough to represent distributions on higher layers. ■

Example 1 shows a scenario where the expressiveness of the NCM class allowed us to fit an NCM instance that could completely match a complex SCM and answer any question of interest. However, we also highlight that expressiveness alone does not necessarily mean that the NCM can solve any causal problem. Experimental data, data from layer 2, can be difficult to obtain in practice, so it would be convenient if we could fit a model $\widehat{M}$ only on observational data from layer 1, then deduce the same causal results using $\widehat{M}$ as if it were the true SCM, $\mathcal{M}^*$. Unfortunately, Corol. 1 states that this deduction typically cannot be made no matter the expressiveness of the model. The next example will illustrate this point more concretely.

D	B	$P(D, B)$
0	0	$34/256$
0	1	$78/256$
1	0	$90/256$
1	1	$54/256$

Table 3: Observational distribution $P(\mathbf{V})$ induced by $\mathcal{M}^*$ from Example 1.

Example 2. Consider the same study of the effects of diet on blood pressure in Example 1 in which the two variables are described by the SCM, $\mathcal{M}^*$, in Eq. 30.

While in the previous example we were able to construct NCM $\widehat{M}$ which matched $\mathcal{M}^*$'s behavior on all three layers of the PCH, this was contingent on having access to the true SCM. This is unlikely to happen in practice. Instead, we typically only have partial information from the SCM. In many cases, we may only have observational data from layer 1. The data in Table 3 exemplifies the aspect of $\mathcal{M}^*$ we may have access to.

Naturally, we should construct an NCM such that matches on the given observed dataset. The NCM $\widehat{M}$ from Example 1 will indeed induce the same observational data $P(\mathbf{V})$, but there are several other ways we could construct a model that fits such criterion. For illustration purposes, consider four NCMs $\widehat{M}_1$, $\widehat{M}_2$, $\widehat{M}_3$, and $\widehat{M}_4$ such that $\widehat{M}_i = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}_i, P(\widehat{\mathbf{U}}) \rangle$ is defined as follows:

$$\begin{cases} \widehat{\mathbf{U}} & := \{\widehat{U}_D, \widehat{U}_B\}, \mathcal{D}_{\widehat{U}_D} = \mathcal{D}_{\widehat{U}_B} = [0, 1] \\ \mathbf{V} & := \{D, B\} \\ \widehat{\mathcal{F}}_i & := \{\hat{f}^i_D, \hat{f}^i_B\} \\ P(\widehat{\mathbf{U}}) & := \widehat{U}_D, \widehat{U}_B \sim \text{Unif}(0, 1), \widehat{U}_D \perp\!\!\!\perp \widehat{U}_B \end{cases}$$

We use two variables in $\widehat{\mathbf{U}}$ with the intent of simplifying some of the expressions, but it is feasible to construct NCMs with the same behavior using one uniform random variable, similar to Example 1.

Concretely, we construct $\widehat{\mathcal{F}}_1$ such that $\widehat{M}_1$ has no unobserved confounding,

$$\widehat{\mathcal{F}}_1 := \begin{cases} \hat{f}_D^1(u_d) & = \begin{cases} 1 & u_d \geq \frac{112}{256} \\ 0 & \text{otherwise} \end{cases} \\ \hat{f}_B^1(d, u_b) & = \begin{cases} 1 & (d = 0) \wedge (u_b \geq \frac{34}{112}) \\ 1 & (d = 1) \wedge (u_b \geq \frac{90}{144}) \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{44}$$

The corresponding neural network for $\widehat{\mathcal{F}}_1$ can be seen as

$$\hat{f}_D^1(u_d) = \hat{f}_{\geq 112/256}(u_d) \tag{45}$$

$$\hat{f}_B^1(d, u_b) = \hat{f}_{\text{OR}}\left(\hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(d), \hat{f}_{\geq 34/112}(u_b)\right), \hat{f}_{\text{AND}}\left(d, \hat{f}_{\geq 90/144}(u_b)\right)\right). \tag{46}$$

$\widehat{\mathcal{F}}_2$ is constructed such that the causal direction of D and B in $\widehat{M}_2$ is reversed, i.e.,

$$\widehat{\mathcal{F}}_2 := \begin{cases} \hat{f}_D^2(b, u_d) & = \begin{cases} 1 & (b = 0) \wedge (u_d \geq \frac{34}{124}) \\ 1 & (b = 1) \wedge (u_d \geq \frac{78}{132}) \\ 0 & \text{otherwise} \end{cases} \\ \hat{f}_B^2(u_b) & = \begin{cases} 1 & u_b \geq \frac{124}{256} \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{47}$$

The corresponding neural network for $\widehat{\mathcal{F}}_2$ can be written as

$$\hat{f}_D^2(b, u_d) = \hat{f}_{\text{OR}}\left(\hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(b), \hat{f}_{\geq 34/124}(u_d)\right), \hat{f}_{\text{AND}}\left(b, \hat{f}_{\geq 78/132}(u_d)\right)\right) \tag{48}$$

$$\hat{f}_B^2(u_b) = \hat{f}_{\geq 124/256}(u_b). \tag{49}$$

$\widehat{\mathcal{F}}_3$ is constructed such that $P^{\widehat{M}_3}(B = 1 \mid do(D = 1))$ is maximized, namely,

$$\widehat{\mathcal{F}}_3 := \begin{cases} \hat{f}_D^3(u_d) & = \begin{cases} 1 & u_d \geq \frac{112}{256} \\ 0 & \text{otherwise} \end{cases} \\ \hat{f}_B^3(d, u_b, u_d) & = \begin{cases} 1 & (d = 1) \wedge (u_b < \frac{34}{112}) \wedge (u_d < \frac{112}{256}) \\ 1 & (u_b \geq \frac{34}{112}) \wedge (u_d < \frac{112}{256}) \\ 1 & (d = 1) \wedge (u_b < \frac{54}{144}) \wedge (u_d \geq \frac{112}{256}) \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{50}$$

The corresponding neural network for $\widehat{\mathcal{F}}_3$ can be written as

$$\hat{f}_D^3(u_d) = \hat{f}_{\geq 112/256}(u_d) \tag{51}$$

$$\hat{f}_B^3(d, u_b, u_d) = \hat{f}_{\text{OR}} \begin{cases} \hat{f}_{\text{AND}}\left(d, \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 34/112}(u_b)\right), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 112/256}(u_d)\right)\right) \\ \hat{f}_{\text{AND}}\left(\hat{f}_{\geq 34/112}(u_b), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 112/256}(u_d)\right)\right) \\ \hat{f}_{\text{AND}}\left(d, \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 54/144}(u_b)\right), \hat{f}_{\geq 112/256}(u_d)\right). \end{cases} \tag{52}$$

Finally, $\widehat{\mathcal{F}}_4$ is constructed such that $P^{\widehat{M}_4}(B = 1 \mid do(D = 1))$ is minimized,

$$\widehat{\mathcal{F}}_4 := \begin{cases} \hat{f}_D^4(u_d) & = \begin{cases} 1 & u_d \geq \frac{112}{256} \\ 0 & \text{otherwise} \end{cases} \\ \hat{f}_B^4(d, u_b, u_d) & = \begin{cases} 1 & (d = 0) \wedge (u_b \geq \frac{34}{112}) \wedge (u_d < \frac{112}{256}) \\ 1 & (d = 1) \wedge (u_b < \frac{54}{144}) \wedge (u_d \geq \frac{112}{256}) \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{53}$$

The corresponding neural network for $\widehat{\mathcal{F}}_4$ can be written as

$$\hat{f}^4_D(u_d) = \hat{f}_{\geq 112/256}(u_d) \tag{54}$$

$$\hat{f}^4_B(d, u_b, u_d) = \hat{f}_{\text{OR}} \begin{cases} \hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(d), \hat{f}_{\geq 34/112}(u_b), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 112/256}(u_d)\right)\right) \\ \hat{f}_{\text{AND}}\left(d, \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 54/144}(u_b)\right), \hat{f}_{\geq 112/256}(u_d)\right). \end{cases} \tag{55}$$

Even though the functions of these NCMs are constructed in very different ways, one can verify that they indeed induce the same layer 1 as shown in Table 3.

Despite this match, they all disagree on a simple layer 2 quantity such as $P(B = 1 \mid do(D = 1))$; to witness note that

$$P^{\widehat{M}_1}(B = 1 \mid do(D = 1)) = P(B = 1 \mid D = 1) = \frac{54}{144} = 0.375 \tag{56}$$

$$P^{\widehat{M}_2}(B = 1 \mid do(D = 1)) = P(B = 1) = \frac{132}{256} \approx 0.5156 \tag{57}$$

$$P^{\widehat{M}_3}(B = 1 \mid do(D = 1)) = \frac{166}{256} \approx 0.6484 \tag{58}$$

$$P^{\widehat{M}_4}(B = 1 \mid do(D = 1)) = \frac{54}{256} \approx 0.2109 \tag{59}$$

Without further information, it's not possible to distinguish which model (if any) is the correct one. Naïevely, choosing an arbitrary model is likely to result in misleading results, even if the model can reproduce the given L_1 data with high fidelity.

In practice, depending on the chosen model, the conclusion of the study could be dramatically different. For instance, someone who fits models $\widehat{M}_1$ or $\widehat{M}_4$ may conclude that a high-vegetable diet is beneficial for lowering blood pressure, while someone who fits model $\widehat{M}_2$ may conclude that it has no effect at all. Someone who fits model $\widehat{M}_3$ may even conclude that eating more vegetables is harmful for regulating blood pressure.

■

C.2 Structural Constraints Embedded in Causal Bayesian Networks

In this section, we provide an example to illustrate the inductive bias embedded in causal diagrams, building on the more comprehensive treatment provided in [5, Sec. 1.4].

Example 3. Consider the following two SCMs:

$$\mathcal{M}_1 := \begin{cases} \mathbf{U} & := \{U_X, U_Y\}, \text{ all binary} \\ \mathbf{V} & := \{X, Y\}, \text{ all binary} \\ \mathcal{F}_1 & := \begin{cases} f^1_X(U_X) & = U_X \\ f^1_Y(X, U_Y) & = X \oplus U_Y \end{cases} \\ P(\mathbf{U}) & := P(U_X = 1) = \frac{1}{2}, P(U_Y = 1) = \frac{1}{4}, U_X \perp\!\!\!\perp U_Y \end{cases}$$

$$\mathcal{M}_2 := \begin{cases} \mathbf{U} & := \{U_X, U_Y\}, \text{ all binary} \\ \mathbf{V} & := \{X, Y\}, \text{ all binary} \\ \mathcal{F}_2 & := \begin{cases} f^2_X(Y, U_X) & = Y \oplus U_X \\ f^2_Y(U_Y) & = U_Y \end{cases} \\ P(\mathbf{U}) & := P(U_X = 1) = \frac{1}{4}, P(U_Y = 1) = \frac{1}{2}, U_X \perp\!\!\!\perp U_Y \end{cases}$$

Note that both $\mathcal{M}_1$ and $\mathcal{M}_2$ induce the same L_1 distributions.

For example, in both models, $P(Y = 1) = \frac{1}{2}$, and $P(Y = 1 \mid X = 1) = \frac{3}{4}$. However, even without making any computation, it seems intuitive from the structure of the functions and noise that a causal query like $P(Y = 1 \mid do(X = 1))$ would have different answers in both models. First, note that f^2_Y in $\mathcal{M}_2$ does not have X as an argument, and there are no other variables in $\mathbf{V}$, so intervening on X would have no causal effect on Y. In other words, the constraint $P^{\mathcal{M}_2}(Y = 1 \mid do(X = 1)) = P(Y = 1)$ is implied. More subtly in the case of $\mathcal{M}_1$, we note that X directly affects Y and there is no unobserved confounding between X and Y. This means that the observed association between X and Y must be due to the causal effect (i.e. $P^{\mathcal{M}_1}(Y = 1 \mid do(X = 1)) = P(Y = 1 \mid X = 1)$). The relationship between X and Y in these two cases can be seen graphically, as shown in Fig. 9.

(a) Graph for $\mathcal{M}_1$.

(b) Graph for $\mathcal{M}_2$.

Figure 9: Causal diagrams for SCMs in Example 3. A directed edge from A to B indicates that A is an argument of the function for B.

■

As illustrated in the previous example, and discussed more formally in [5], these equality constraints across distributions arise from the qualitative structural properties of the SCM such as which variable is an argument of which function, and which variables share exogenous influence. In fact, these constraints are invariant to the details of the functions and the distribution of the exogenous variables.

Moreover, there are exponentially many constraints implied by an SCM in the collection of L_1 and L_2 distributions, which can be parsimoniously represented in the form of a *causal bayesian network* (Def. 15)[12] In fact, the graphical component of this object provides an intuitive interpretation for these constraints. It can be shown that the closure of such constraints entails the do-calculus [5, Thm. 5], which means that all identifiable effects can be computed from them (due to the completeness shown in [43]).

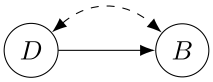


Figure 10: Causal diagram $\mathcal{G}$ of $\mathcal{M}^*$ from Example 1

Revisiting Examples 1 and 2 and applying the definition of causal diagram (Def. 12), we can see from the structure of $\mathcal{M}^*$ that the causal diagram $\mathcal{G}$ in Fig. 10 is induced. If we had this information, we could immediately eliminate $\widehat{M}_2$ from Example 2 as a possible model. The reason is that we can read off from diagram $\mathcal{G}$ that B cannot be an argument of f_D.

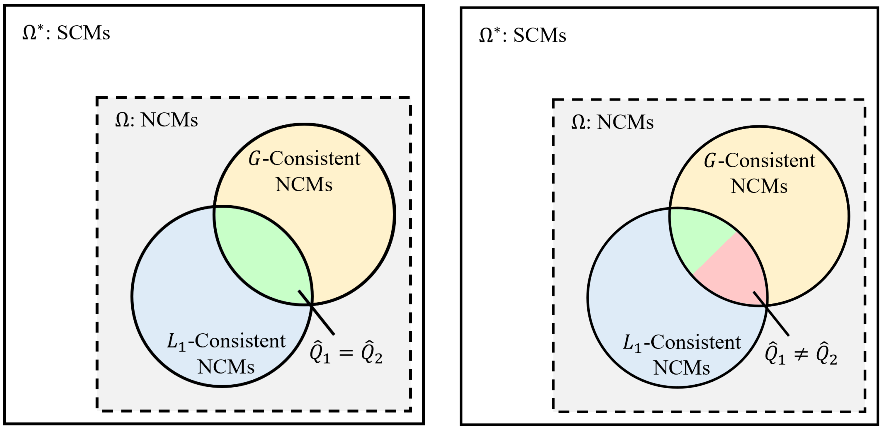
With an SCM-like structure, the NCM naturally induces its own structural constraints. Def. 7 in the main paper shows how to construct an NCM to fit the same constraints entailed by a given causal diagram from another SCM. In some way, the original model induces mark in the collection of observational and experimental distributions. In the case of the NCM, these constraints are used as input, inductive bias to constrain its functions. This is powerful because any inferences performed on a $\mathcal{G}$-constrained NCM will automatically satisfy the constraints of $\mathcal{G}$.

C.3 Solving Identification through NCMs

The notion of identification requires that a certain effect is identifiable by all (unobserved) SCMs compatible with the corresponding structural constraints. This was extended to NCMs through Def. 8; see also Fig. 11. In fact, Alg. 1 helps to solve the neural identification problem such that if there are two NCMs compatible with the constraints but with different predictions for the causal effect, it will return "`FAIL`". Otherwise, it will returns the estimation of the query in every ID case. Note that identification of an effect does not require that the NCM and the true SCM have the same functions, just the effects need to match. The following two examples illustrate both positive and negative instances of such operation.

Example 4. Consider once again the study of diet on blood pressure introduced in Example 1. Suppose we are particularly interested in the L_2 expression of $Q = P(B = 1 \mid do(D = 1))$, the causal effect of diet on blood pressure. The true SCM $\mathcal{M}^*$ is not available, but L_1 data ($P(\mathbf{v})$)

[12] This is a generalization of the notion of markov compatibility used in L_1 models, such as bayesian networks [53]. In our case, there are multiple distributions of different nature, observational and experimental, and the constraints are among them (i.e., not conditional independences).


Ω*: SCMs
Ω: NCMs
$\mathcal{G}$-Consistent NCMs
L_1-Consistent NCMs
$\widehat{Q}_1 = \widehat{Q}_2$
Ω*: SCMs
Ω: NCMs
$\mathcal{G}$-Consistent NCMs
L_1-Consistent NCMs
$\widehat{Q}_1 \neq \widehat{Q}_2$

(a) In the identifiable case, all NCMs that are $\mathcal{G}$-consistent and L_1-consistent with $\mathcal{M}^*$ will also match in Q.

(b) In the non-identifiable case, there could exist two NCMs, $\widehat{M}_1$ and $\widehat{M}_2$, that are both $\mathcal{G}$-consistent and L_1-consistent but still disagree in Q.

Figure 11: A visual representation of the ID problem. Here, Q is a query of interest, and $\widehat{Q}_i$ is the answer for that query induced by NCM $\widehat{M}_i$. The goal is to check if all NCMs that are $\mathcal{G}$-consistent and L_1-consistent are also consistent in Q.

presented in Table 3 is (as shown Example 2). Furthermore, we are informed by a doctor that the two variables follow the relation specified by the causal diagram $\mathcal{G}$ in Fig. 10. Is Q identifiable from $P(\mathbf{v})$ and $\Omega(\mathcal{G})$, where $\Omega(\mathcal{G})$ is the set of all $\mathcal{G}$-constrained NCMs?

Unfortunately, the answer is no for this particular case. To see why, example 2 provides 4 NCMs whose L_1 distributions match $P(\mathbf{v})$ but disagree on Q. Even though $\widehat{M}_2$ can be eliminated as a possible proxy since it is not $\mathcal{G}$-consistent, the other three models are compatible with $\mathcal{G}$, so we still cannot pinpoint the correct answer for Q.

Running Alg. 1 on this setting would result in the algorithm generating two sets of parameters, $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$, that minimize and maximize Q. The behaviors of an NCM with these parameters may reflect the behaviors of $\widehat{M}_3$ and $\widehat{M}_4$ from Example 2, and they do not agree on Q. ■

Example 5. Suppose now we return to the data collection process and we receive new information about a third variable, sodium intake (S) ($S = 1$ represents a high sodium diet, 0 otherwise).

With the introduction of S in the endogenous set $\mathbf{V}$, $\mathcal{M}^*$ can be written as follows:

$$\mathcal{M}^* := \begin{cases} \mathbf{U} & := \{U_D, U_{DB}, U_S, U_B\}, \text{ all binary} \\ \mathbf{V} & := \{D, S, B\} \\ \mathcal{F} & := \begin{cases} f_D(U_D, U_{DB}) & = \neg U_D \wedge \neg U_{DB} \\ f_S(D, U_S) & = \neg D \oplus U_S \\ f_B(S, U_B) & = (S \vee U_{DB}) \oplus U_B \end{cases} \\ P(\mathbf{U}) & := \begin{cases} P(U_D = 1) = P(U_{DB} = 1) = P(U_S = 1) = P(U_B = 1) = \frac{1}{4} \\ \text{all } U \in \mathbf{U} \text{ are independent} \end{cases} \end{cases} \tag{60}$$

As reflected in the updated $\mathcal{M}^*$, diet only affects blood pressure through sodium content, which tends to be higher in low-vegetable diets. Note that the L_1, L_2, and L_3 distributions relating D and B are unchanged with this modification of $\mathcal{M}^*$. In this case, the query of interest is evaluated as

$$\begin{aligned} P(B = 1 \mid do(D = 1)) &= P\left((S_{D=1} \vee U_{DB}) \oplus U_B = 1\right) \\ &= P\left(((\neg 1 \oplus U_S) \vee U_{DB}) \oplus U_B = 1\right) \\ &= P\left((U_S \vee U_{DB}) \oplus U_B = 1\right) \\ &= \frac{120}{256} = 0.46875, \end{aligned} \tag{61}$$

which matches the result in Eq. 33.

D	S	B	$P(D,S,B)$
0	0	0	$13/256$
0	0	1	$15/256$
0	1	0	$21/256$
0	1	1	$63/256$
1	0	0	$81/256$
1	0	1	$27/256$
1	1	0	$9/256$
1	1	1	$27/256$

Table 4: Updated observational distribution $P(\mathbf{V})$ induced by $\mathcal{M}^*$ for Example 5.

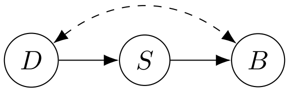


Figure 12: Causal diagram $\mathcal{G}$ of $\mathcal{M}^*$ with additional S variable for Example 5

This is just the computation of the distribution from Nature's perspective. As the previous examples, we do not have access to the true model ($\mathcal{M}^*$), just the observational distribution $P(\mathbf{v})$, as shown in Table 4. We do have access to the updated causal diagram as shown in Fig. 12.

The addition of the new variable and the refinement of the model change the identifiability status of the query $Q = P(B = 1 \mid do(D = 1))$. No matter how Alg. 1 chooses the parameters for $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$, $\widehat{M}$ will always induce the same result for Q as long as it induces the correct $P(\mathbf{v})$. Specifically, through standard do-calculus derivation we would obtain:

$$\begin{aligned}
&P(B = 1 \mid do(D = 1)) \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid do(D = 1)) P(B = 1 \mid do(D = 1), s) && \text{Marginalization} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid do(D = 1)) P(B = 1 \mid do(D = 1), do(s)) && \text{Rule 2} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid do(D = 1)) P(B = 1 \mid do(s)) && \text{Rule 3} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid D = 1) P(B = 1 \mid do(s)) && \text{Rule 2} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid D = 1) \sum_{d \in \mathcal{D}_D} P(B = 1 \mid d, do(s)) P(d \mid do(s)) && \text{Marginalization} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid D = 1) \sum_{d \in \mathcal{D}_D} P(B = 1 \mid d, do(s)) P(d) && \text{Rule 3} \\
&= \sum_{s \in \mathcal{D}_S} P(s \mid D = 1) \sum_{d \in \mathcal{D}_D} P(B = 1 \mid d, s) P(d) && \text{Rule 2}
\end{aligned}$$

Finally, we can replace the corresponding probabilities with the actual values, i.e.,

$$\begin{aligned}
P(B=1 \mid do(D=1)) &= P(S=0 \mid D=1)P(B=1 \mid D=0, S=0)P(D=0) \\
&\quad + P(S=0 \mid D=1)P(B=1 \mid D=1, S=0)P(D=1) \\
&\quad + P(S=1 \mid D=1)P(B=1 \mid D=0, S=1)P(D=0) \\
&\quad + P(S=1 \mid D=1)P(B=1 \mid D=1, S=1)P(D=1) \\
&= \left(\frac{108}{144}\right)\left(\frac{15}{28}\right)\left(\frac{112}{256}\right) + \left(\frac{108}{144}\right)\left(\frac{27}{108}\right)\left(\frac{144}{256}\right) \\
&\quad + \left(\frac{36}{144}\right)\left(\frac{63}{84}\right)\left(\frac{112}{256}\right) + \left(\frac{36}{144}\right)\left(\frac{27}{36}\right)\left(\frac{144}{256}\right) \\
&= \frac{15}{32} = 0.46875 \qquad (62)
\end{aligned}$$

On the other hand, *any* $\mathcal{G}$-constrained NCM that induces $P(\mathbf{v})$ will produce the same result, matching Eq. 62. For instance, consider the following NCM construction:

$$\widehat{M} := \begin{cases}
\widehat{\mathbf{U}} & := \{\widehat{U}_{DB}, \widehat{U}_S\}, \mathcal{D}_{\widehat{U}_{DB}} = \mathcal{D}_{\widehat{U}_S} = [0,1] \\
\mathbf{V} & := \{D, S, B\} \\
\widehat{\mathcal{F}} & := \begin{cases}
\hat{f}_D(u_{DB}) & = \begin{cases} 1 & u_{DB} \geq \frac{112}{256} \\ 0 & \text{otherwise} \end{cases} \\
\hat{f}_S(d, u_S) & = \begin{cases} 1 & (d=0) \wedge (u_S \geq \frac{1}{4}) \\ 1 & (d=1) \wedge (u_S \geq \frac{3}{4}) \\ 0 & \text{otherwise} \end{cases} \\
\hat{f}_B(b, u_{DB}) & = \begin{cases} 1 & (s=0) \wedge \left((\frac{52}{256} \leq u_{DB} < \frac{112}{256}) \vee (u_{DB} \geq \frac{220}{256})\right) \\ 1 & (s=1) \wedge \left((\frac{28}{256} \leq u_{DB} < \frac{112}{256}) \vee (u_{DB} \geq \frac{148}{256})\right) \\ 0 & \text{otherwise} \end{cases}
\end{cases} \\
P(\widehat{\mathbf{U}}) & := \widehat{U}_{DB}, \widehat{U}_S \sim \text{Unif}(0,1), \widehat{U}_{DB} \perp\!\!\!\perp \widehat{U}_S
\end{cases} \qquad (63)$$

The corresponding neural network for $\widehat{\mathcal{F}}$ can be written as

$$\hat{f}_D(u_{DB}) = \hat{f}_{\geq 112/256}(u_{DB}) \qquad (64)$$

$$\hat{f}_S(d, u_S) = \hat{f}_{\text{OR}}\left(\hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(d), \hat{f}_{\geq 1/4}(u_S)\right), \hat{f}_{\text{AND}}\left(d, \hat{f}_{\geq 3/4}(u_S)\right)\right) \qquad (65)$$

$$\hat{f}_B(s, u_{DB}) = \hat{f}_{\text{OR}} \begin{cases}
\hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(s), \hat{f}_{\geq 52/256}(u_{DB}), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 112/256}(u_{DB})\right)\right) \\
\hat{f}_{\text{AND}}\left(\hat{f}_{\text{NOT}}(s), \hat{f}_{\geq 220/256}(u_{DB})\right) \\
\hat{f}_{\text{AND}}\left(s, \hat{f}_{\geq 28/256}(u_{DB}), \hat{f}_{\text{NOT}}\left(\hat{f}_{\geq 112/256}(u_{DB})\right)\right) \\
\hat{f}_{\text{AND}}\left(s, \hat{f}_{\geq 148/256}(u_{DB})\right).
\end{cases} \qquad (66)$$

Note that $\widehat{M}$ follows the format of a $\mathcal{G}$-constrained NCM defined in Def. 7. Additionally, one can verify that $\widehat{M}$ induces the same L_1 quantities shown in Table 4. Applying the mutilation procedure on $\widehat{M}$ to compute the query,

$$
\begin{aligned}
&P^{\widehat{M}}(B = 1 \mid do(D = 1)) \\
&= P(S_{D=1} = 0)P\left(\left(\frac{52}{256} \leq U_{DB} < \frac{112}{256}\right) \vee \left(U_{DB} \geq \frac{220}{256}\right)\right) \\
&+ P(S_{D=1} = 1)P\left(\left(\frac{28}{256} \leq U_{DB} < \frac{112}{256}\right) \vee \left(U_{DB} \geq \frac{148}{256}\right)\right) \\
&= \left(\frac{96}{256}\right) P(S_{D=1} = 0) + \left(\frac{192}{256}\right) P(S_{D=1} = 1) \\
&= \left(\frac{96}{256}\right) P\left(U_S < \frac{3}{4}\right) + \left(\frac{192}{256}\right) P\left(U_S \geq \frac{3}{4}\right) \\
&= \frac{72}{256} + \frac{48}{256} = \frac{120}{256} = 0.46875.
\end{aligned}
$$

This result indeed matches Eq. 61. We further note $\widehat{M}$ is constructed to fit $P(\mathbf{v})$ and not necessarily to match $\mathcal{M}^*$. This is evident when comparing the inputs and outputs of the functions in $\widehat{\mathcal{F}}$ to those from $\mathcal{M}^*$. Still, due to the structural constraints encoded in the NCM, it so happens that $\widehat{M}$ also matches in our L_2 query of interest even though it is only constructed to match only on layer 1. It is remarkable that incorporating these family of constraints into the NCM structure allows one to successfully perform cross-layer inferences.

■

Thm. 4 is powerful because it says that performing the identification task in the class of NCMs will yield the correct result, even if the true model can be any SCM. We note that this is not the case for every class of models, even if a model from that class can be both L_1-consistent and $\mathcal{G}$-consistent with the true model. In particular, the expressivity of the NCM allows for this result, and using a less expressive model may produce incorrect results. We illustrate this with the following examples.

Example 6. Suppose we attempt to identify $P(B = 1 \mid do(D = 1))$ from the problem in Example 4 using a less expressive model class such as the set of all Markovian models. Recall that the given observational $P(\mathbf{V})$ is shown in Table 3, and the given causal diagram $\mathcal{G}$ is shown in Fig. 10. An example model from the class of Markovian models that achieve L_1 and $\mathcal{G}$ consistency with $\mathcal{M}^*$ is $\widehat{M}_1$ from Example 2.

Note that due to the lack of unobserved confounding in Markovian models, the induced value for $P(B = 1 \mid do(D = 1))$ for any L_1 and $\mathcal{G}$-consistent Markovian model including $\widehat{M}_1$ is $P(B = 1 \mid do(D = 1)) = P(B = 1 \mid D = 1) = \frac{54}{144} = 0.375$. Since all models from this class (Markovian) agree on the same value for this quantity, the conclusion reached is that it must be identifiable. However, this is clearly not the case, as we know the true value, $P^{\mathcal{M}^*}(B = 1 \mid do(D = 1)) = 0.46875$. In this case, the reason we reach the wrong result is that the true model is not Markovian, and the set of all Markovian models is not expressive enough to account for all possible SCMs. ■

More interestingly, another example using a different class of models follows.

Example 7. Consider a special class of SCMs which, given variables $\mathbf{V}$ and the graph $\mathcal{G}$, take the following structure.

$$
\widehat{M} = \begin{cases}
\widehat{\mathbf{U}} & := \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in C^2(\mathcal{G})\}, \mathcal{D}_{\widehat{U}} = [0, 1] \text{ for all } \widehat{U} \in \widehat{\mathbf{U}}, \\
\widehat{\mathbf{V}} & := \mathbf{V}, \\
\widehat{\mathcal{F}} & := \left\{ \hat{f}_V(\mathbf{pa}_V, \mathbf{u}_V) = \left(\bigoplus_{x \in \mathbf{pa}_V} x \right) \oplus \left(\bigoplus_{u_i \in \mathbf{u}_V} \mathbb{1}(a_{V,i} \leq u_i < b_{V,i}) \right) : V \in \mathbf{V} \right\}, \\
& \quad a_{V,i}, b_{V,i} \in [0, 1], a_{V,i} \leq b_{V,i}, \\
& \quad \mathbf{pa}_V \text{ is the set of parents of } V \text{ in } \mathcal{G}, \\
& \quad \mathbf{u}_V = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in C^2(\mathcal{G}) \text{ such that } V \in \mathbf{C}\}, \\
P(\widehat{\mathbf{U}}) & := \widehat{U} \sim \text{Unif}(0, 1) \text{ for all } \widehat{U} \in \widehat{\mathbf{U}}.
\end{cases}
\tag{67}
$$

Here, $C^2(\mathcal{G})$ denotes the set of all C^2-components of $\mathcal{G}$. The $\oplus$ operator denotes bitwise XOR, with the larger version representing the bitwise XOR of all elements of a set (0 if empty).

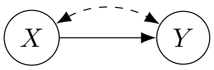


Figure 13: Causal diagram $\mathcal{G}$ of $\mathcal{M}^*$ from Example 7

X	Y	$P(X, Y)$
0	0	p_0
0	1	p_1
1	0	p_2
1	1	p_3

Table 5: General observational distribution $P(\mathbf{V})$ induced by $\mathcal{M}^*$ from Example 7. We assume positivity (i.e. $p_i > 0$ for all i).

We may opt to use a model like this for its simplicity, since having fewer parameters allows for easier optimization. Suppose we try to use this model class to decide if $P(Y \mid do(X))$ is identifiable in the case where the true model, $\mathcal{M}^*$, induces the graph $\mathcal{G}$ in Fig. 13. We are given $\mathcal{G}$ along with $P(\mathbf{v})$, which can be represented as shown in Table 5. If we construct a model $\widehat{M}$ from Eq. 67, it would have the following form.

$$\widehat{M} = \begin{cases} \widehat{\mathbf{U}} & := \{\widehat{U}_{XY}\}, \mathcal{D}_{\widehat{U}_{XY}} = [0, 1], \\ \widehat{\mathbf{V}} & := \{X, Y\} \\ \widehat{\mathcal{F}} & := \begin{cases} \hat{f}_X(\widehat{u}_{XY}) & = \mathbb{1}(a_X \leq \widehat{u}_{XY} < b_X), \\ \hat{f}_Y(x, \widehat{u}_{XY}) & = x \oplus \mathbb{1}(a_Y \leq \widehat{u}_{XY} < b_Y) \end{cases} \\ P(\widehat{\mathbf{U}}) & := U_{XY} \sim \text{Unif}(0, 1). \end{cases}$$

The parameters we choose to fit $P(\mathbf{V})$ are the values of a_X, b_X, a_Y, b_Y. However, note that there in order for $\widehat{M}$ to attain L_1-consistency with $\mathcal{M}^*$, we must have:

1. $b_X - a_X = p_2 + p_3$.
2. $(a_X - a_Y = p_1$ and $b_Y - a_X = p_2)$ or $(b_Y - b_X = p_1$ and $b_X - a_Y = p_2)$

2 implies $b_Y - a_Y = p_1 + p_2$, so in all cases, $P^{\widehat{M}}(Y = 1 \mid do(X = 0)) = p_1 + p_2$ and $P^{\widehat{M}}(Y = 1 \mid do(X = 1)) = p_0 + p_3$. In other words, since $P^{\widehat{M}}(Y \mid do(X))$ matches for all models from this class that are L_1 and $\mathcal{G}$-consistent, the conclusion reached is that it must be identifiable.

However, suppose $\mathcal{M}^*$ takes the following form:

$$\mathcal{M}^* = \begin{cases} \mathbf{U} & := \{U_X, U_Y\}, \mathcal{D}_{U_X} = \mathcal{D}_{U_Y} = \{0, 1\}, \\ \mathbf{V} & := \{X, Y\} \\ \mathcal{F} & := \begin{cases} f_X(u_X) & = u_X, \\ f_Y(x, u_Y) & = u_Y, \end{cases} \\ P(\mathbf{U}) & := \begin{cases} P(U_X = 0, U_Y = 0) & = p_0 \\ P(U_X = 0, U_Y = 1) & = p_1 \\ P(U_X = 1, U_Y = 0) & = p_2 \\ P(U_X = 1, U_Y = 1) & = p_3 \end{cases} \end{cases}$$

One can quickly verify that $\mathcal{M}^*$ does indeed induce the $P(\mathbf{V})$ in Table 5 and $\mathcal{G}$ from Fig. 13. Note that in this $\mathcal{M}^*$, $P(Y = 1 \mid do(X = 0)) = P(Y = 1 \mid do(X = 1)) = p_1 + p_3$. In fact, this means there is a complete mismatch from any possible choice of $\widehat{M}$. Like shown in Example 4, $P(Y \mid do(X))$ is actually a non-identifiable query when $\mathcal{G}$ takes the form in Fig. 13. Simply put, $\mathcal{M}^*$ could be an SCM for which the model class in Eq. 67 is not expressive enough to capture.

■

C.4 Symbolic versus Optimization-based Approaches for Identification

In this section, we discuss the relationship between current approaches to causal identification/estimation versus the new, neural/optimization-based approach proposed in this work.

The problem of effect identification has been extensively studied in the literature, and [54] introduced the *do-calculus* (akin to differential or integral calculus), a set of symbolic rules that can be applied to any expression and evaluate whether a certain invariance across interventional distributions hold given local assumptions encoded in a causal diagram. Multiple applications of the rules can be combined to search for a reduction from a target effect to the distributions in which data is available. There exist algorithms capable of utilizing the structural constraints encoded in the causal diagram, based on what is known as *c-component factorization* [64], to find a symbolic expression of the L_2 query in terms of the L_1 distribution efficiently (for more general tasks, see, e.g., [43, 13]). We call this approach *symbolic* since it aims to find a closed-form expression of the target effect in terms of the input distributions.

Alternatively, one could take an optimization-based route to tackle the identification problem, such as in Alg. 1 discussed in Sec. 3. This approach entails a search through the space of possible structural models while trying to maximize/minimize the value of the target query subject to the constraints found in the inputted data. We call this an *optimization-based approach*. These two approaches are indeed linked as evident from Thm. 2, which establishes a duality saying that the identification status of a target query is shared across symbolic and optimization-based approaches.

We discuss next some of the possible trade-offs and synergies between these two families of methods. We start with the symbolic approach and re-stating the definition of identification [55], with minor modifications to highlight its model-theoretic perspective regarding the space of SCMs:

Definition 17 (Causal Effect Identification)**.** Let Ω^* be the space containing all SCMs defined over endogenous variables $\mathbf{V}$. We say that a causal effect $P(\mathbf{y} \mid do(\mathbf{x}))$ is identifiable from the observational distribution $P(\mathbf{v})$ and the causal diagram $\mathcal{G}$ if $P^{\mathcal{M}_1}(\mathbf{y} \mid do(\mathbf{x})) = P^{\mathcal{M}_2}(\mathbf{y} \mid do(\mathbf{x}))$ for every pair of models $\mathcal{M}_1, \mathcal{M}_2 \in \Omega^*$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ both induce $\mathcal{G}$ and $P^{\mathcal{M}_1}(\mathbf{v}) = P^{\mathcal{M}_2}(\mathbf{v})$ ■

Fig. 14 provides an illustration of the many parts involved in this definition. In words, an interventional distribution $Q = P(\mathbf{Y} \mid do(\mathbf{x}))$ is said to be identifiable if for all SCMs in Ω^* (top-left) that share the same causal diagram $\mathcal{G}$ (top-right), and induce the same probability distribution $P(\mathbf{V})$ (bottom-left), they generate the same distribution to the target query Q (bottom-right). In fact, identifiability can be understood as if the details of the specific form of the true SCM $\mathcal{M}^*$ – its functions and probability distribution over the exogenous variables – are irrelevant, and the constraints encoded in the causal diagram are sufficient to perform the intended cross-layer inference from the source to the target distributions.

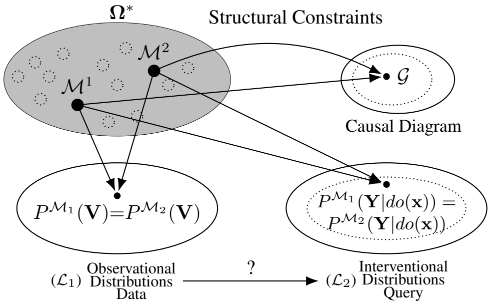

Ω^*
Structural Constraints
$\mathcal{M}^2$
$\mathcal{M}^1$
$\mathcal{G}$
Causal Diagram
$P^{\mathcal{M}_1}(\mathbf{V})=P^{\mathcal{M}_2}(\mathbf{V})$
$P^{\mathcal{M}_1}(\mathbf{Y}|do(\mathbf{x})) =$
$P^{\mathcal{M}_2}(\mathbf{Y}|do(\mathbf{x}))$
$(\mathcal{L}_1)$ Observational Distributions Data
?
$(\mathcal{L}_2)$ Interventional Distributions Query

Figure 14: $P(\mathbf{Y} \mid do(\mathbf{x}))$ is identifiable from $P(\mathbf{V})$ and $\mathcal{G}$ if for all SCM $\mathcal{M}^1, \mathcal{M}^2$ (top left) such that $\mathcal{M}^1, \mathcal{M}^2$ match in $P(\mathbf{V})$ (bottom left) and $\mathcal{G}$ (top right), then they also match in the target distribution $P(\mathbf{Y} \mid do(\mathbf{x}))$ (bottom right).

One important observation that follows is that, operationally, symbolic methods do not work directly in the Ω^* space, but on top of the constraints implied by the true SCM on the causal diagram. There are a number of reasons for this, but we note that if all that is available about $\mathcal{M}^*$ are the constraints encoded in $\mathcal{G}$, it is somewhat expected and reasonable to operate directly on those, instead of considering the elusive, underlying structural causal model. Further, we already know through the CHT that we will never be able to recover $\mathcal{M}^*$ anyways, so one could see it as unreasonable to consider $\mathcal{M}^*$ as the target of the analysis. Still, even if we wanted to refer directly to $\mathcal{M}^*$, practically speaking, searching through the space Ω^* is a daunting task since each candidate SCM $\mathcal{M} = \langle \mathcal{F}, P(\mathbf{U}) \rangle \in \Omega^*$ is a complex, infinite dimensional object. Note that the very definition of identification (Def. 17) does not even mention the true SCM $\mathcal{M}^*$, since it is completely out of reach in most practical situations. The task here is indeed about whether one can get by without having to know much about the underlying collection of mechanisms and still answer

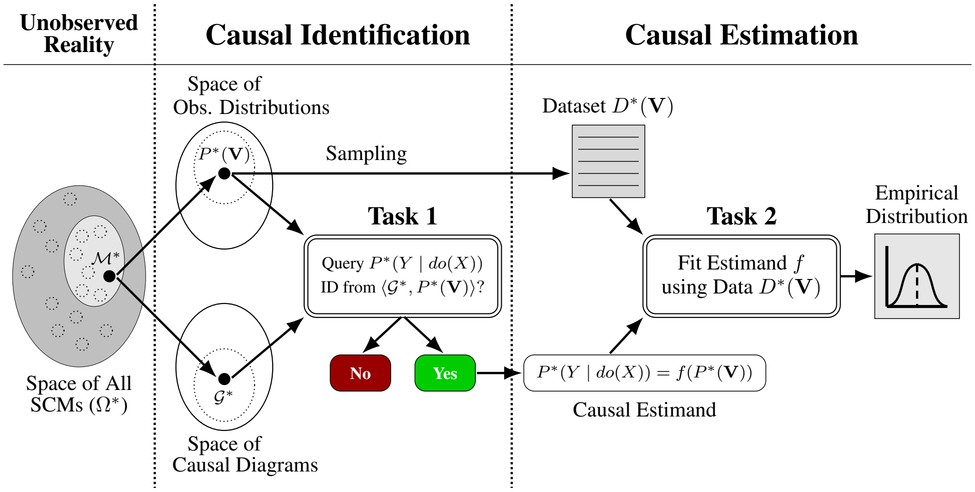

Unobserved Reality
Causal Identification
Causal Estimation
Space of Obs. Distributions
$P^*(\mathbf{V})$
Sampling
Dataset $D^*(\mathbf{V})$
Task 1
Task 2
Empirical Distribution
$\mathcal{M}^*$
Query $P^*(Y \mid do(X))$ ID from $\langle \mathcal{G}^*, P^*(\mathbf{V}) \rangle$?
Fit Estimand f using Data $D^*(\mathbf{V})$
No
Yes
$P^*(Y \mid do(X)) = f(P^*(\mathbf{V}))$
Causal Estimand
$\mathcal{G}^*$
Space of All SCMs (Ω^*)
Space of Causal Diagrams

Figure 15: Causal Pipeline with unobserved SCM $\mathcal{M}^*$ in the left, generating both $\mathcal{G}$ and $P(\mathbf{v})$, which is taken as input for the identification task (1), which generates input to the estimation task (2).

the query of interest. (In causal terminology, an identification instance is called *non-parametric* whenever no constraints are imposed over the functional class $\mathcal{F}$ or exogenous distribution $P(\mathbf{U})$.

The full causal pipeline encompassing both tasks of effect identification and estimation is illustrated in Fig. 15. Note that the detachment with respect to the true Nature, alluded to above, is quite prominent in the figure. The very left part contains $\mathcal{M}^*$, which is abstracted away in the form of the causal diagram $\mathcal{G}$ and through its marks imprinted into the observational distribution $P(\mathbf{V})$. The symbolic methods takes the pair $\langle \mathcal{G}, P(\mathbf{V}) \rangle$ as input (instead of the unobservable $\langle \mathcal{F}^*, P^*(\mathbf{U}) \rangle$), and attempt to determine whether the target distribution Q can be expressed in terms of the available input distributions. Formally, the question asked is about whether there exists a mapping f such that

$$Q = f_{\mathcal{G}}(P(\mathbf{V})). \tag{68}$$

The decision problem regarding the existence of $f(.)$ is certainly within the domain of causal reasoning since it takes an arbitrary causal diagram as input, which encodes causal assumptions about the underlying SCM, and reason through causal axioms on whether this is sufficient to perform the intended cross-layer inference. Causal reasoning is nothing more than the manipulation and subsequent derivation of new causal facts from known causal invariances. Whenever this step is successfully realized, one can then ignore the causal invariances (or assumptions) entirely, and simply try to evaluate the r.h.s. of Eq. 68 to compute Q.

We note that in most practical settings, only a dataset with samples collected from $P(\mathbf{V})$ is available, say $D(\mathbf{V})$, as opposed to the distribution itself. This entails the second task that asks for a computationally and statistically attractive way of evaluating the r.h.s. of Eq. 68 from the finite samples contained in the dataset $D(\mathbf{V})$. Even though the l.h.s. of Eq. 68 is a causal distribution, the evaluation is complete oblivious to the semantics of such quantity and is entirely about fitting f using the data $D(\mathbf{V})$ in the best possible way.

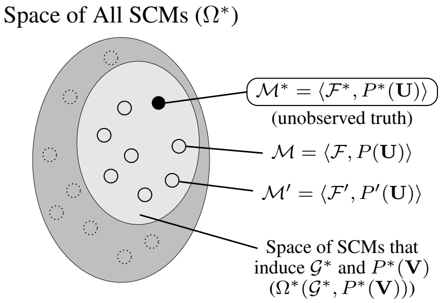

Space of All SCMs (Ω^*)
$\mathcal{M}^* = \langle \mathcal{F}^*, P^*(\mathbf{U}) \rangle$
(unobserved truth)
$\mathcal{M} = \langle \mathcal{F}, P(\mathbf{U}) \rangle$
$\mathcal{M}' = \langle \mathcal{F}', P'(\mathbf{U}) \rangle$
Space of SCMs that induce $\mathcal{G}^*$ and $P^*(\mathbf{V})$ ($\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$)

Figure 16: In the case where Q is identifiable, any two SCMs in $\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$ – the space of SCMs matching $\mathcal{M}^*$ in $P^*(\mathbf{V})$ and $\mathcal{G}^*$ – will also match $\mathcal{M}^*$ in Q.

We now turn our attention to the optimization-based approach as outlined in Alg. 1 (Sec. 3) and note that it will have a very different interpretation of the identifiability definition. In fact, instead of avoiding the SCM altogether and focusing on the causal diagrams' constraints, as done by the symbolic approach, it will put the SCM at the front and center of the analysis. Fig. 16 illustrates this point by showing the space of all SCMs called Ω^* (in dark gray). The true, unknown SCM $\mathcal{M}^*$ is shown as a black dot and generates the pair $\langle \mathcal{G}^*, P^*(\mathbf{V})$, the latter is taken as the input of the identification analysis. The approach will then focus on the set of SCMs that have the same interventional constraints as $\mathcal{G}^*$ (Def. 5) and that also have the capability of generating the same observational distribution $P^*(\mathbf{V})$ (Def. 4). This subspace is called $\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$ and marked in light gray. Since $\mathcal{M}^*$ is almost never inferrable, the optimization-approach will search for two SCMs $\mathcal{M}, \mathcal{M}' \in \Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$ such that the former will try to maximize the target query (Q_{max}), while the latter minimizes it (Q_{min}). Two candidate SCMs for this job are shown in the figure. Whenever the search ends and two of such SCMs are discovered, they will predict exactly the same interventional distribution (i.e., $Q_{min} = Q_{max}$) if Q is identifiable. This is because by the definition of identifiability (Def. 17), all SCMs in the light gray area will necessarily exhibited the same interventional behavior.

On the other hand, the situation is qualitatively different when considering non-identifiable effects. To understand how, the first observation comes from the contrapositive of Def. 17, which says that non-identifiability implies there exists (at least) two SCMs $\mathcal{M}, \mathcal{M}'$ within the $\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$ subspace that share the constraints as in $\mathcal{G}^*$ and generate the same observational distribution ($P(\mathbf{V}) = P'(\mathbf{V})$), but induce different interventional distributions($P(Y|do(X)) \neq P'(Y|do(X))$). The optimization-based approach will try to exploit precisely this fact, searching for non-identifiability witnesses, possibly different than the true $\mathcal{M}^*$. Those are illustrated as red dots and hollow circles in Fig. 17. Still, this is all that is needed to determine whether an effect is not identifiable. In practice, each distribution $\mathcal{Q}$, $\mathcal{Q}'$ will only be approximations and it may be hard to detect non-identifiability depending on how close they are from each other. This is indeed what leads to the probabilistic nature of such identifiability statements when using optimization-based methods (as discussed in Appendix B), as opposed to the deterministic ones entailed by the do-calculus and symbolic family.

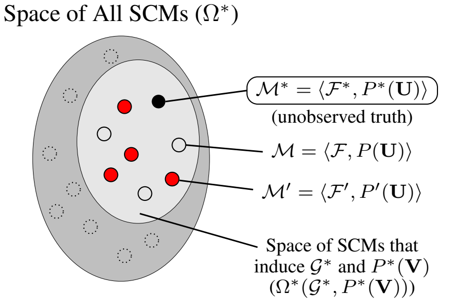

Space of All SCMs (Ω^*)
$\mathcal{M}^* = \langle \mathcal{F}^*, P^*(\mathbf{U}) \rangle$
(unobserved truth)
$\mathcal{M} = \langle \mathcal{F}, P(\mathbf{U}) \rangle$
$\mathcal{M}' = \langle \mathcal{F}', P'(\mathbf{U}) \rangle$
Space of SCMs that induce $\mathcal{G}^*$ and $P^*(\mathbf{V})$ ($\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$)

Figure 17: In the case where Q is non-identifiable, there exist two SCMs in $\Omega^*(\mathcal{G}^*, P^*(\mathbf{V}))$ (the space of SCMs matching $\mathcal{M}^*$ in $P^*(\mathbf{V})$ and $\mathcal{G}^*$) that do not match $\mathcal{M}^*$ in Q.

By and large, both approaches take the causal assumptions and try to infer something about the unknown, target quantity $P^*(Y|do(X = x))$ that is defined by the unobserved $\mathcal{M}^*$. In the case of the symbolic approach, the structural assumptions encoded in the causal diagram $\mathcal{G}$ are used, while an optimization-based approach will use constraints encoded through the scope of the functions and independence among the exogenous variables $\mathbf{U}$. Evaluating $P^*(Y \mid do(X = x))$ is possible in identifiable cases, which means that our predictions will match what the true $\mathcal{M}^*$ would say, while the assumptions would be too weak in others cases, and non-identifiability will take place, which means we are unable to make any statement about $\mathcal{M}^*$ without strengthening the assumptions.

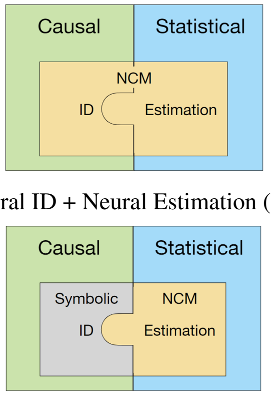
After having understood the different takes of both approaches to the problem of identification, we consider again the entire pipeline shown in Fig. 15. We note that the two tasks, identification and estimation, are both necessary in any causal analysis, but they are usually studied separately in the symbolic literature. Symbolic methods returns the identifiability status of a query and include the mapping f whenever the effect is identifiable. The target quantity can then be evaluated by standard statistical methods (plug-in) or more refined learning procedures, e.g., multi-propensity score/inverse probability weighting [31], empirical risk minimization [32], and double machine learning [33].

On the other hand, the optimization-based framework proposed here is capable of identifying and estimating the target quantities in an integrated manner. As shown through simulations (Sec. 5), given the sampling nature of the optimization procedure, the performance of the method relies not only on the causal assumptions, but also on accurate optimization, which may require a large amount of samples and computation for training.

In practice, one may consider a hybrid approach where a symbolic algorithm for the identification step is ran first, since it is deterministic and always returns the right answer, and then the estimation step is performed through an NCM. With this use case in mind, we define Alg. 4 in Fig. 18 (left) as this hybrid alternative to the more pure Alg. 1. In some sense, this approach combines the best

Algorithm 4: Identifying queries with a symbolic ID procedure and then estimating with NCMs.

```
Input  : causal query $Q = P(\mathbf{y} \mid do(\mathbf{x}))$, $L_1$ data
         $P(\mathbf{v})$, and causal diagram $\mathcal{G}$
Output : $P^{\mathcal{M}^*}(\mathbf{y} \mid do(\mathbf{x}))$ if identifiable, FAIL
         otherwise
if symbolicID($Q$) then
    $\widehat{M} \leftarrow$ NCM($\mathbf{V}$, $\mathcal{G}$)              // from Def. 7
    $\boldsymbol{\theta}^* \leftarrow \arg\min_{\boldsymbol{\theta}} D(P^{\widehat{M}(\boldsymbol{\theta})}(\mathbf{v}), P(\mathbf{v}))$     // for
      some divergence $D$
    return $P^{\widehat{M}(\boldsymbol{\theta}^*)}(\mathbf{y} \mid do(\mathbf{x}))$
else
    return FAIL
```



(a) Neural ID + Neural Estimation (Alg. 1)

(b) Symbolic ID + Neural Estimation (Alg. 4)

Figure 18: (Left panel) Algorithm for solving identification problem with symbolic solvers and estimation with NCMs. (Right) Schematic illustrating differences between Alg. 1 (a) and Alg. 4 (b).

of both worlds – it relies on an ideal ID algorithm, which is correct with probability one, and the powerful computational properties of neural networks, which may perform well and scale to complex settings, for estimation. We illustrate this using the two figures in Fig. 18 (right).

After all, the NCM approach provides a cohesive framework that unifies both the causal and the statistical components involved in evaluating interventional distributions. In the case of the hybrid approach, we note that the causal reasoning is shifted to the symbolic identification methods. It is remarkable but somewhat unsurprising that neural methods, which are the state-of-the-art for function approximations, are capable of performing the statistical step required for causal estimation. More interestingly, we demonstrate in this work that neural nets (and proxy SCMs, in general) are also capable of performing causal reasoning, such as in the inferences required to solve the ID problem. Many other works have studied using neural networks as an estimation tool, as listed in the introduction, but to the best of our knowledge, our work is the first one that utilizes neural networks to provide a complete solution within the neural framework to the identification problem, and causal reasoning more broadly.

In addition to showing that neural nets are also capable of performing causal reasoning, there are other implications for using a proxy SCM in place of the true SCM, as opposed to abstracting the true SCM entirely. Firstly, the generative capabilities of proxy SCMs, like the NCM, can be useful if the user desires a source of infinite data, which is a quite common use case found in the literature. Secondly, it provides quick estimation results for multiple queries via the mutilation procedure without the need for retraining. Whenever we apply a symbolic approach we need to derive (and then train) a specialized estimand, which can be time consuming. Thirdly, working in the space of SCMs tends to provide straightforward interpretation of the causal problems using its semantics, which implies that it has the potential to be more easily extensible to other related problems. One may be interested in generalizations of the ID problem such as identifying other types of queries or identifying and fusing from different sources and experimental conditions [4, 6, 43], such as in reinforcement learning. In general, extending the line of reasoning of Alg. 1 to other identification cases is much simpler than deriving a new symbolic solution under different constraints.

D Generalizations

D.1 NCMs with other Functions and Noise Distributions (Proofs)

For the sake of concreteness and simplicity of exposition, the NCM from Def. 3 specifically uses $\mathrm{Unif}(0,1)$ noise for the variables in $\mathbf{U}$ and MLPs as functions. This leads to easy implementation and concrete examples for discussion (as in Appendix C).

We note that these are not meant to be limitations for the NCM framework. The NCM object can be extended to use other function or noise types, provided they exhibit certain properties. We will henceforth refer to specific versions of the NCM with their particular implementation of the noise and functions (e.g., Def. 3 will be called NCM-FF-Unif).

Consider a more general version of the NCM below:

Definition 18 (NCM (General Case)). A Neural Causal Model (for short, NCM) $\widehat{M}(\boldsymbol{\theta})$ over variables $\mathbf{V}$ with parameters $\boldsymbol{\theta} = \{\theta_{V_i} : V_i \in \mathbf{V}\}$ is an SCM $\langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}})\rangle$ such that

- $\widehat{\mathbf{U}} \subseteq \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \subseteq \mathbf{V}\}$, where each $\widehat{U}$ is associated with some subset of variables $\mathbf{C} \subseteq \mathbf{V}$;
- $\widehat{\mathcal{F}} = \{\hat{f}_{V_i} : V_i \in \mathbf{V}\}$, where each $\hat{f}_{V_i}$, parameterized by $\theta_{V_i} \in \boldsymbol{\theta}$, maps values of $\mathbf{U}_{V_i} \cup \mathbf{Pa}_{V_i}$ to values of V_i for some $\mathbf{Pa}_{V_i} \subseteq \mathbf{V}$ and $\mathbf{U}_{V_i} = \{\widehat{U}_{\mathbf{C}} : \widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}}, V_i \in \mathbf{C}\}$;

Furthermore, a NCM $\widehat{M}(\boldsymbol{\theta})$ should have the following properties:
(P1) For all $\widehat{U} \in \widehat{\mathbf{U}}$, $\widehat{U}$ has a well-defined probability density function $P(\widehat{U})$ (i.e. its cumulative density function is absolutely continuous).
(P2) For all functions $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ and all functions $g : \mathbb{R} \to \mathbb{S}$, where $\mathbb{S} \subset \mathbb{R}$ is a countable set, there exists parameterization $\theta^*_{V_i}$ such that $\hat{f}_{V_i}(\cdot; \theta^*_{V_i}) = g$ (universal representation). ■

We can use these properties to prove a more general version of Thm. 1. We first state the following result to aid the proof.

Fact 5 (Probability Integral Transform [11, Thm. 2.1.10]). *For any random variable X with a well-defined probability density function $P(X)$, there exists a function $f : \mathcal{D}_X \to [0,1]$ such that $f(X)$ is a* $\mathrm{Unif}(0,1)$ *random variable.* ■

The generalized proof follows.

Theorem 5 (NCM Expressiveness). *For any SCM $\mathcal{M}^* = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U})\rangle$, there exists a NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}})\rangle$ s.t. $\widehat{M}$ is L_3-consistent w.r.t. $\mathcal{M}^*$.* ■

Proof. Lemma 2 guarantees that there exists a canonical SCM $\mathcal{M}_{\mathsf{CM}} = \langle \mathbf{U}_{\mathsf{CM}}, \mathbf{V}, \mathcal{F}_{\mathsf{CM}}, P(\mathbf{U}_{\mathsf{CM}})\rangle$ that is L_3-consistent with $\mathcal{M}^*$. Hence, to construct $\widehat{M}$, it suffices to show how to construct $\mathcal{M}_{\mathsf{CM}}$ using the architecture of a NCM.

Following Def. 18, we choose $\widehat{\mathbf{U}} = \{\widehat{U}_{\mathbf{V}}\}$. By property P1 and Fact 5, there exists a function g^U such that $g^U(\widehat{U}_{\mathbf{V}})$ is a $\mathrm{Unif}(0,1)$ random variable.

Using the construction in Lem. 5, there must also exist a function g^R such that

$$P(g^R(g^U(\widehat{U_{\mathbf{V}}}))) = P(\mathbf{U}_{\mathsf{CM}}) \tag{69}$$

To choose the functions of $\widehat{\mathcal{F}}$, consider each $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$. Combining the above results, there must exist

$$g_{V_i} = f^{\mathsf{CM}}_{V_i}\left(\mathbf{pa}_{V_i}, g^R\left(g^U(\widehat{U}_{\mathbf{V}})\right)\right) \tag{70}$$

By property P2, we can choose parameterization $\theta^*_{V_i}$ such that $\hat{f}_{V_i}(\cdot; \theta^*_{V_i}) = g_{V_i}$.

By Eqs. 69 and 70, the NCM $\widehat{M}$ is constructed to match $\mathcal{M}_{\mathsf{CM}}$ on all outputs. Hence, for any counterfactual query $\boldsymbol{\varphi}$, we have

$$\mathcal{M}_{\mathsf{CM}} \models \boldsymbol{\varphi} \Leftrightarrow \widehat{M} \models \boldsymbol{\varphi}$$

and therefore

$$\mathcal{M}^* \models \boldsymbol{\varphi} \Leftrightarrow \widehat{M} \models \boldsymbol{\varphi}.$$

□

The general $\mathcal{G}$-constrained version of the NCM follows the same procedure as Def. 7.

Definition 19 ($\mathcal{G}$-Constrained NCM (General Case))**.** Let $\mathcal{G}$ be the causal diagram induced by SCM $\mathcal{M}^*$. Construct NCM $\widehat{M}$ as follows. **(1)** Choose $\widehat{\mathbf{U}}$ s.t. $\widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}}$ if and only if $\mathbf{C}$ is a C^2-component in $\mathcal{G}$. Each $\widehat{U}_{\mathbf{C}}$ has its own distribution $P(\widehat{U}_{\mathbf{C}})$ independent of other variables in $\widehat{\mathbf{U}}$, but it is shared as input to the functions of all variables in $\mathbf{C}$. **(2)** For each variable $V_i \in \mathbf{V}$, choose $\mathbf{Pa}_{V_i} \subseteq \mathbf{V}$ s.t. for every $V_j \in \mathbf{V}$, $V_j \in \mathbf{Pa}_{V_i}$ if and only if there is a directed edge from V_j to V_i in $\mathcal{G}$. Any NCM in this family is said to be $\mathcal{G}$-constrained. ■

We prove a general version of Thm. 3 similarly.

Theorem 6 (NCM L_2-$\mathcal{G}$ Representation)**.** *For any SCM $\mathcal{M}^*$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-constrained NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, \widehat{P}(\widehat{\mathbf{U}}) \rangle$ that is L_2-consistent w.r.t. $\mathcal{M}^*$.* ■

Proof. Fact 3 states that there exists a $\mathcal{G}$-canonical SCM $\mathcal{M}_{\mathsf{GCM}} = \langle \mathbf{U}_{\mathsf{GCM}}, \mathbf{V}, \mathcal{F}_{\mathsf{GCM}}, P(\mathbf{U}_{\mathsf{GCM}}) \rangle$ that is L_2-consistent with $\mathcal{M}^*$. Hence, to construct $\widehat{M}$, we can simply show how to construct $\mathcal{M}_{\mathsf{GCM}}$ using the architecture of a NCM.

Following Def. 19, we choose $\widehat{\mathbf{U}} = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G})\}$, where $\mathbb{C}(\mathcal{G})$ is the set of all C^2-components of $\mathcal{G}$. Denote $\widehat{\mathbf{U}}_V = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G}) \text{ s.t. } V \in \mathbf{C}\}$.

By property P1 and Fact 5, there exists a function g_V^U such that $g_V^U(\widehat{\mathbf{U}}_V)$ is a $\mathrm{Unif}(0,1)$ random variable for each $V \in \mathbf{V}$.

Using the construction in Lem. 5, there must also exist a function g_V^R such that

$$P(g_V^R(g_V^U(\widehat{\mathbf{U}}_V))) = P(\mathbf{R}_V) \tag{71}$$

for each $V \in \mathbf{V}$.

To choose the functions of $\widehat{\mathcal{F}}$, consider each $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$. Combining the above results, there must exist

$$g_{V_i} = f_{V_i}^{\mathsf{GCM}}\left(\mathbf{pa}_{V_i}, g_{V_i}^R\left(g_{V_i}^U(\widehat{\mathbf{U}}_V)\right)\right) \tag{72}$$

By property P2, we can choose parameterization $\theta^*_{V_i}$ such that $\hat{f}_{V_i}(\cdot; \theta^*_{V_i}) = g_{V_i}$.

By Eqs. 71 and 72, the NCM $\widehat{M}$ is constructed to match $\mathcal{M}_{\mathsf{CM}}$ on all outputs. Hence, $\widehat{M}$ must be L_2-consistent with $\mathcal{M}^*$. □

The remaining results, including Corol. 1, Thm. 2, Thm. 4, Corol. 2, Corol. 3, and Corol. 4 can be proven for NCMs with minimal changes to the proofs for NCMs.

D.2 Pearl's Causal Hierarchy and Other Classes of Models

As developed in the paper, we note that the expressiveness power of NCMs is one desirable features of this class as models, which makes them comparable to the SCM class when considering the PCH's capabilities. Broadly speaking, neural networks are well understood to be maximally expressive due to the Universal Approximation Theorem [14]. We note that Theorem 1 follows the same spirit when considering causal inferences, i.e., NCMs is a data structure built from neural networks that are maximally expressive on *every* distribution in *every* layer of the PCH generated by the underlying SCM $\mathcal{M}^*$, and which has the potential of acting as a proxy under some conditions. This capability is not shared across all neural architectures, and to understand how this discussion extends to these other classes, we define expressivity more generally (Fig. 19):

Definition 20 (L_i-Expressiveness)**.** Let Ω^* be the set of SCMs and Ω be a model class of interest. We say that Ω is L_i expressive if for all $\mathcal{M}^* \in \Omega^*$, there exists $\mathcal{M} \in \Omega$ s.t. $L_i(\mathcal{M}^*) = L_i(\mathcal{M})$. ■

It should be emphasized that expressiveness on layers higher than Layer 1 is a highly nontrivial property (e.g., Example 1 in Appendix C.1 shows the complications involved in constructing an NCM to replicate another SCM on all 3 layers). Even a model class that is L_2-expressive can represent far more settings than model classes that are only L_1-expressive, since there are many more ways in which SCMs can differ on L_2 than on L_1. Theorem 1 states that NCMs are L_3-expressive, which leads to the question: what model classes are not L_3-expressive?

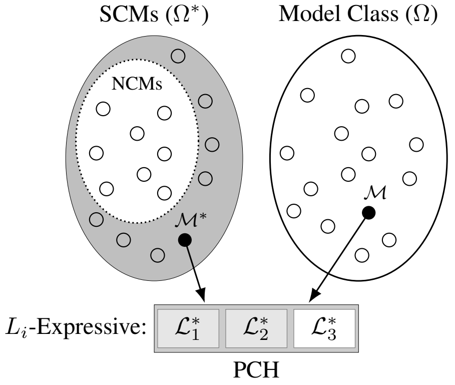


Figure 19: The SCM class Ω^* is shown in the l.h.s. and another class (non-SCM) Ω class is shown in the r.h.s.. If we consider $M^* \in \Omega^*$ that generates a specific PCH, Ω is L_i representative if there is a model $M \in \Omega^*$ that exhibit the same behavior under L_i.

Many neural model classes are L_1-expressive, which can be seen as the property of modeling an unique probability distribution over a set of observed variables. Additionally, this model should be generative, i.e., one should be able to draw samples from the model's fitted distribution. Model classes that have this property include variational models with normalizing flows [59], which are generative models and are proven to be able to fit any data distribution. It is also believed that popular generative models like generative adversarial networks [20] have this property as well.

Interesting enough, being generative does not imply being causal or counterfactual. For instance, these model classes are not well defined for valuating any causal distributions. A model from one of these classes that is fitted on $P(\mathbf{V})$ can only provide samples from $P(\mathbf{V})$. On the other hand, distributions like $P(\mathbf{V}_{\mathbf{x}})$, the distribution of $\mathbf{V}$ under the intervention $do(\mathbf{X} = \mathbf{x})$ for some $\mathbf{X} \subseteq \mathbf{V}$, are entirely different distributions from Layer 2 of the PCH. A model fitted on $P(\mathbf{V})$ cannot output samples from $P(\mathbf{V}_{\mathbf{x}})$, and it certainly cannot output samples from *every* Layer 2 or Layer 3 distributions. These models, therefore, are not even L_2-expressive, let alone L_3-expressive.

It is somewhat natural, and perhaps obvious, that model classes that are not defined on higher layers cannot be expressive on higher layers. Still, we can also consider this property on model classes proposed by other deep learning works which include a causal component. Works such as [66] and [61] have had great success estimating higher layer quantities like average treatment effects under backdoor/conditional ignorability conditions [55, Sec. 3.3.1]. Perhaps these models can always succeed at modeling Layer 2 expressions as long as the ground truth can be modeled by an SCM that also meets those same backdoor conditions. However, since there are SCMs that do not fit the backdoor setting, we cannot say that these models are L_2 or L_3-expressive. We illustrate this point by providing an example (Example 5) in Appendix C.3, where the NCM estimates a causal quantity in a setting that does not meet the backdoor condition.

In some way, these works and most of the literature are concerned with estimating causal effects that are identifiable from observational data $P(V)$, usually in the form of the backdoor formula, which means that the "causal reasoning" already happened outside the network itself. In other words, a causal distribution $Q = P(\mathbf{Y}|do(\mathbf{X} = \mathbf{x}))$ being identifiable means that there exists a function f such that $Q = f(P(V))$. The causal "inference" means determining from causal assumptions (for example the causal diagram G) whether Q is identifiable, or whether f exists. Whenever this is the case, and f has a closed-expression, the inference is done, and the task that remains is to estimate f efficiently from finite samples obtained from $P(V)$.

Even though many existing works do not have the L_2 or L_3-expressiveness properties, this is not to say that the NCM is the only model class that is expressive enough to represent all the three layers of the PCH. Consider the following example of a model class artificially constructed to be L_2-expressive:

Definition 21 (L_2-Expression Wrapper)**.** Let M^{L_1} be an L_1-expressive model class such as a normalizing flow model. Denote $M_P^{L_1}$ as a model from this class fitted on distribution P. For a collection of L_2 distributions $\mathbf{P}^*$, define L_2-expression wrapper $M^{L_2} := \{M_P^{L_1} : P \in \mathbf{P}^*\}$. ■

It is fairly straightforward to show that the L_2-expression wrapper is L_2-expressive. For any SCM $\mathcal{M}^*$, we can simply construct L_2-expression wrapper M^{L_2} with $\mathbf{P}^* = L_2(\mathcal{M}^*)$. Then, for any L_2-query Q, M^{L_2} answers Q by providing the distribution of $M_Q^{L_1}$. In other words, we could simply train a generative model for *every* L_2 distribution of an SCM, and collectively, these models induce all L_2 distributions of the SCM.

Although expressive, this model is highly impractical for most practical uses. First, there are typically too many distributions in Layer 2 to effectively learn all of them. A new distribution would be created for every possible intervention of every subset of variables. In fact, this amount grows exponentially with the number of variables, and in continuous cases, it is infinite.

Second, unlike the NCM data structure (Def. 7), there is no clear way to incorporate the constraints of a CBN in the expression wrapper. Without these constraints, it is obvious that the expression wrapper suffers from the same limitations from the CHT as NCMs (Corol. 1). Suppose we are given $P(\mathbf{V})$ from true SCM $\mathcal{M}^*$, but we are interested in the distribution $P(\mathbf{V_x})$. We could guarantee an expression wrapper M^{L_2} such that $M^{L_1}_{P(\mathbf{V})} = P^{\mathcal{M}^*}(\mathbf{V})$, but $M^{L_1}_{P(\mathbf{V_x})}$ could be any distribution over $\mathbf{V}$ that is consistent with $\mathbf{x}$, and we would not be able to guarantee that it matches $P^{\mathcal{M}^*}(\mathbf{V_x})$.

Attempting to incorporate the constraints of a CBN in an expression wrapper would scale poorly and be difficult to realize in practice. Consider the following example.

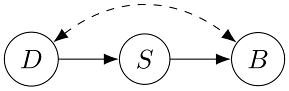


Figure 20: Causal diagram $\mathcal{G}$ of $\mathcal{M}^*$ from Example 5.

Example 8. Consider the same setting from Example 5, where we would like to evaluate a query like $Q = P(B = 1 \mid do(D = 1))$ given $P(\mathbf{V})$ and the graph $\mathcal{G}$ (displayed again in Fig. 20 for convenience). If we were using the expression wrapper, we would need an estimator for each of the L_1 and L_2 distributions such $P(\mathbf{V}), P(\mathbf{V}_{D=1}), P(\mathbf{V}_{S=1})$, and so on.

However, unlike the NCM, the expression wrapper does not automatically incorporate the constraints implied by $\mathcal{G}$. Here is a list of some constraints used in the do-calculus derivation in Example 5:

1. $P(B \mid do(D), S) = P(B \mid do(D, S))$
2. $P(B \mid do(D, S)) = P(B \mid do(S))$
3. $P(S \mid do(D)) = P(S \mid D)$
4. $P(D \mid do(S)) = P(D)$
5. $P(B \mid do(S), D) = P(B \mid S, D)$

This list is not exhaustive, and there are more constraints induced by $\mathcal{G}$. This list is also compressed, since intervening on different values for the same variables result in different distributions. Note that even with just 3 binary variables, the number of constraints is quite long when enumerated.

Maintaining all of these constraints while fitting all of the desired distributions is the key difficulty of using the L_2-expression wrapper. While fitting $P(\mathbf{V})$, the expression wrapper would have to also generate some other distributions from layer 2 like $P(\mathbf{V}_{D=1})$, keeping constraints in mind. One cannot make such a choice arbitrarily. For instance, setting $P(\mathbf{V}_{D=1})$ to $P(\mathbf{V})$, but with all values of D set to 1, would still violate constraint 3 from the above list. This would be infeasible for any tractable optimization procedure. ■

This illustrates the benefits of the properties of the NCM. Not only is the NCM L_3-expressive, a non-trivial property to achieve, but it also is easily able to incorporate structural constraints.

E Frequently Asked Questions

Q1. Why are the sets of noise variables for each function not necessarily independent in the SCM and NCM?

Answer. Each structural causal model includes a set of observed (endogenous) and unobserved (exogenous) variables, $\mathbf{V}$ and $\mathbf{U}$, respectively. The set $\mathbf{U}$ accounts for the unmodeled sources of variation that generate randomness in the system. On the one hand, an exogenous variable $U \in \mathbf{U}$ could affect all endogenous variables in the system. On the other hand, each $U_i \in \mathbf{U}$ could affect only one observable $V_i \in \mathbf{V}$ and be independent to all other variables in $\mathbf{U} \setminus \{U_i\}$. These two represent opposite sides of a spectrum, where the former means zero assumptions on the presence of unobserved confounding, while the latter represents the strongest assumption known as Markovianity. We operate in between, allowing for all possible cases, which is represented in the causal diagram through bidirected edges.

In the context of identification, Markovianity is usually known as the lack of unobserved confounding. As shown in Corol. 3, the identification problem becomes trivial and causation is always derivable from association in this case, i.e., *all* layer 2 queries are computable from layer 1 data. In practice, identification becomes somewhat more involved in non-Markovian settings. Example 4 in Appendix C shows a case where a causal query is not identifiable even in a 2 variable case. More discussion and examples can be found in [5, Sec. 1.4].

Q2. How come the true SCM cannot be learned from observational data? If the functions are universal approximators, this seems to be easily accomplished.

Answer. This is not possible in general, as highlighted by Corol. 1. The issue is that the learned model, $\widehat{M}$, could generate the distribution of observational data perfectly yet still differ in interventional distributions from the true SCM, $\mathcal{M}^*$. Searching for (or randomly choosing) some SCM that matches the observational distribution will almost surely fail to match the interventional distributions. Since the true SCM, $\mathcal{M}^*$, is unobserved, one would not be able to determine whether the inferences made on the learned model $\widehat{M}$ preserve the validity of the causal claims. See Example 2 in Appendix C for a more detailed discussion.

Q3. Why are feedforward neural networks and uniform distributions used in Def. 3? Would these results not hold for other function approximators and noise distributions?

Answer. We implemented in the body of the paper NCMs with feedforward neural networks and uniform distributions for the sake of clarity of exposition and presentation. The results of the NCM are not limited to this particular architecture choice, and other options may offer different benefits in practice. A general definition of the NCM is introduced in Def. 18, and the corresponding results in this generalized setting are provided in Appendix D.1.

Q4. What is a causal diagram? What is a CBN? Is there any difference between these objects?

Answer. A causal diagram $\mathcal{G}$ is a non-parametric representation of an SCM in the form of a graph, which follows Def. 12, such that nodes represent variables, a directed edge from X to Y indicates that X is an input to f_Y, and a bidirected edge from X to Y indicates that there is unobserved confounding between X and Y. The causal diagram $\mathcal{G}$ is strictly weaker than the generating SCM $\mathcal{M}^*$. Talking about the diagram itself is usually ill-defined, since it is relevant in the context of the constraints it imposes over the space of distributions generated by $\mathcal{M}^*$ (both observed and unobserved).

A Causal Bayesian Network (CBN, Def. 15) is a pair consisting of a causal diagram $\mathcal{G}$ and a collection of layer 2 (interventional) distributions $\mathbf{P}^*$, where the distributions follow a set of constraints represented in the diagram (i.e., they are "compatible"). Remarkably, the causal diagram induced by any SCM, along with its induced layer 2 distributions, together form a valid CBN (Fact 2). Causal diagrams and the constraints encoded in CBNs are often the fuel to causal inference methods (e.g., do-calculus) and used to bridge the gap between layer 1 and layer 2 quantities. We refer readers to Appendix C.2 for a further discussion on these inductive biases and their motivation, and [5, Sec. 1.3] for further elaborations.

Q5. Based on the answer to Q4, it is not entirely clear what an NCM really represents. Is an NCM a causal diagram? Or is it an SCM? Or should I think about it in a different way?

Answer An NCM is not a causal diagram, and although an NCM is technically an SCM, by definition, it may be helpful to think of it as a separate data structure. As discussed above, SCMs contain strictly more information than causal diagrams. Still, NCMs without $\mathcal{G}$-consistency (Def. 5) do not even ascertain the constraints represented in $\mathcal{G}$ about layer 2 distributions. So, even though an NCM seems to appear like an SCM, it is an "empty" data structure on its own. Whenever we impose $\mathcal{G}$-consistency, NCMs may have the capability of generating interventional distributions, contingent on L_1-consistency and the identifiability status of the query.

More broadly, we think it is useful to think about an NCM as a candidate for a proxy model for the true SCM. Unlike the SCM, the NCM's concrete definitions for functions and noise allow the NCM to be optimized (via neural optimization procedures) for learning in practical settings. One additional difference between the NCM and SCM is their purpose in causal inference. The SCM data structure is typically used to represent reality, so in some sense, the SCM has all information of interest from all layers of the PCH. Of course, this reality is unobserved, so the SCM is typically unavailable for use. The NCM is then trained as a proxy for the true SCM, so it has all of the functionality of an SCM while being readily available. However, unlike the SCM, the "quality" of the information in the NCM is dependent on the amount of input information. If an NCM is only trained on data from layer 1, the Neural CHT (Corol. 1) says that distributions induced by the NCM from higher layers will, in general, be incorrect. For that reason, while the NCM has the same interface as the SCM for the PCH, the NCM's higher layers may be considered "underdetermined", in the sense that they do not contain any meaningful information about reality.

Q6. Why do you assume the causal diagram exists instead of learning it? What if the causal diagram is not available?

Answer. As implied by the N-CHT (Corol. 1), cross-layer inferences cannot be accomplished in the general case without additional information. In other words, the causal diagram assumption is out of necessity. In general, one cannot hope to learn the whole causal diagram from observational data either. However, it is certainly not the case that the causal diagram will always be available in real world settings. In such cases, other assumptions may be necessary which may help structural learning, or perhaps the causal diagram can be partially specified to a degree that allows answering the query of interest (see also Footnotes 7 and 8).

Q7. Are NCMs assumed to be acyclic?

Answer. We assume for this work that the true SCM $\mathcal{M}^*$ is *recursive*, meaning that the variables can be ordered in a way such that if $V_i < V_j$, then V_j is guaranteed to not be an input to f_{V_i}. This implies that the causal diagram induced by $\mathcal{M}^*$ is acyclic, so $\mathcal{G}$-constrained NCMs (from Def. 7) must also be recursive. That said, not all works make the recursive assumption, so the general definition of NCMs does not need to be constrained a priori.

Q8. Why is $\mathcal{M}^*$ assumed to be an SCM instead of an NCM in the definition for neural identifiability (Def. 8)?

Answer. $\mathcal{M}^*$ indicates the ground truth model of reality, and we assume that reality is modeled by an SCM, without any constraints in the form of the functions or probability over the exogenous variables. The NCM is a data structure that we constructed to solve causal inference problems, but the inferences of interest are those from the true model. For this reason, several results were needed to connect the properties of NCMs to those from SCMs, including Thm. 2 (showing that NCMs can encode the same constraints as SCMs' causal diagrams) and Thm. 3 (showing that NCMs can represent any $\mathcal{G}$-consistent SCM on layer 2). These results are required to show the validity (or meaning) of the inferences made from the NCM, which otherwise would have no connection to the true model, $\mathcal{M}^*$.

Naturally, checking for identifiability within the space of NCMs defeats the purpose of the work since the model of interest is not (necessarily) an NCM. Conveniently, however, the expressiveness of the NCM (as shown in Thm. 3) shows that if two SCMs agree on $P(\mathbf{V})$ and $\mathcal{G}$ but do not agree on $P(\mathbf{y} \mid do(\mathbf{x}))$, then there must also be two NCMs that behave similarly. This means that attempting to check for identifiability in the space of NCMs will produce the correct result, which is why Alg. 1 can be used. Still, interestingly, this may no longer be the case within a less expressive model class (see Example 7 in Appendix C.3).

Q9. What is the purpose of Thm. 3 if we already have Thm. 1?

Answer. While Thm. 1 shows that NCMs (from Def. 3) can express any SCM on all three layers of the PCH, it does not make any claims about $\mathcal{G}$-constrained NCMs (Def. 7). In fact, a $\mathcal{G}$-constrained NCM is obviously incapable of expressing any SCM that is not $\mathcal{G}$-consistent. Thm. 3 emphasizes that even when encoding the structural constraints in $\mathcal{G}$ (CBN-like), NCMs are still capable of expressing any SCM that is also compatible with $\mathcal{G}$. If this property did not hold, it would not always be possible to use the NCM as a proxy for the SCM, since there may be some SCMs that are $\mathcal{G}$-consistent, but there does not exist an NCM that matches the corresponding distributions.

One important subtlety of Thm. 3 is that the expressiveness is described after applying the constraints of $\mathcal{G}$. It may be relatively simple to use a universal approximator to construct a model class that can express any SCM on layer 2, such as the L_2-expression wrapper (Def. 21 from Appendix D.2). However, the L_2-expression wrapper does not naturally enforce the constraints of $\mathcal{G}$. Incorporating such constraints is nontrivial, and even if the constraints were applied, the model class may no longer retain its expressiveness. Thm. 3 shows that the NCM is the maximally expressive model class that still respects the constraints of $\mathcal{G}$.

Q10. The objective function in Eq. 5 is not entirely clear. What role does λ play?

Answer. The purpose of Alg. 2 is to solve the optimization problem in lines 2 and 3 of Alg. 1. The goal can be described in two parts: maximize/minimize a query of interest, like $P(\mathbf{y} \mid do(\mathbf{x}))$, while simultaneously learning the observational distribution $P(\mathbf{V})$. This is a nontrivial optimization problem, and there are many different ways to approach it. The proposed method penalizes both of these quantities in the objective. The first term of Eq. 5 attempts to maximize log-likelihood, while the second term attempts to maximize/minimize $P(\mathbf{y} \mid do(\mathbf{x}))$. Certainly, we do not want the optimization of the second term to affect the optimization of the first term, since we need accurate learning of $P(\mathbf{V})$ to make any deductions on the identifiability status of $P(\mathbf{y} \mid do(\mathbf{x}))$ and subsequent estimation of it.

Further, the hyperparameter λ is used to balance the amount of attention placed on maximizing/minimizing $P(\mathbf{y} \mid do(\mathbf{x}))$. If λ is too small, then the term would be ignored, and the optimization of $P(\mathbf{y} \mid do(\mathbf{x}))$ could fail even in non-ID cases. On the other hand, if λ is too large, the optimization procedure might attempt to maximize/minimize $P(\mathbf{y} \mid do(\mathbf{x}))$ at the expense of learning $P(\mathbf{V})$ properly, which may result in incorrect inferences in ID cases. Our solution starts λ at a large value to allow the optimization to quickly find parameters that maximize/minimize $P(\mathbf{y} \mid do(\mathbf{x}))$, and then it lowers λ as training progresses so that $P(\mathbf{V})$ is learned accurately by the end of training.

Q11. Why would one want, or prefer, to solve identifiability using NCMs? Would it not be easier to use existing results (e.g., do-calculus) to determine the identifiability of a query, then use the NCM for estimation?

Answer. We provide a more detailed discussion on this topic in Appendix C.4. In summary, when solving causal problems like identification or estimation, we introduced a new framework using NCMs where one can construct a proxy model to mimic the true SCM via neural optimization. Meanwhile, existing works focus on solving the problems at a higher-level of abstraction with the causal diagram, ignoring the true SCM. We call these works "symbolic" since they directly use the constraints of the causal diagram to algorithmically derive a solution. We are not deciding for the causal analyst which method to use when comparing symbolic and optimization-based methods. For instance, we developed Alg. 4 as an alternative to Alg. 1 for researchers who would prefer to use a symbolic approach for identification while using an NCM for estimation.

We note that the identification and estimation tasks typically appear together in practice, and it is remarkable that the proxy-model framework with NCMs is capable of solving the entire causal pipeline. In some sense, the identification task provides the causal reasoning required to perform estimation. Once a query is determined to be identifiable, the estimation task is only a matter of fitting an expression. It may be somewhat unsurprising that the expressiveness of neural networks along with its strong optimization results allow it to fit any expression, but this work shows, for the first time, that neural nets are also capable of performing the causal reasoning required to determine identifiability. We note that Alg. 1 describes a procedure for solving the ID problem that is unique compared to symbolic approaches. Our goal with Alg. 1 is to provide insights to alternative approaches to causal problems under the proxy model framework compatible with optimization methods.

Q12. Why would one want to learn an entire SCM to solve the estimation problem for a specific query? Shouldn't the efforts be focused on estimating just the query of interest?

Answer. Indeed, existing works typically focus on a single interventional distribution and also abstract out the SCM, as shown in Fig. 15. In particular, solutions to the identification problem typically derive an expression for the query of interest using the causal diagram, and solutions to the estimation problem directly evaluate the derived expression.

One of the novelties of this work is the introduction of a method of solving these tasks using a proxy model of the SCM. Compared to existing methods, there are a number of reasons one may want to have a proxy model. For instance, one may want to have generative capabilities of both layer 1 and identifiable layer 2 distributions rather than simply obtaining a probability value. We refer readers to Appendix C.4 for a more detailed discussion.

Q13. It appears that the theory and results only hold if learning is perfect. What happens if there is error in training?

Answer. This work does not make any formal claims regarding cases with error in training. Still, one of our goals with the experiments of Sec. 5 is to show empirically that results tend to be fairly accurate if training error is minimized to an acceptable degree (i.e., L_1-consistency nearly holds). A more refined understanding and detailed theoretical analysis of robustness to training error is an interesting direction for future work.