NEURAL CAUSAL MODELS FOR COUNTERFACTUAL IDENTIFICATION AND ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating hypothetical statements about how the world would be had a different course of action been taken is arguably one key capability expected from modern AI systems. Counterfactual reasoning underpins discussions in fairness, the determination of blame and responsibility, credit assignment, and regret. In this paper, we study the evaluation of counterfactual statements through neural models. Specifically, we tackle two causal problems required to make such evaluations, i.e., counterfactual identification and estimation from an arbitrary combination of observational and experimental data. First, we show that neural causal models (NCMs) are expressive enough and encode the structural constraints necessary for performing counterfactual reasoning. Second, we develop an algorithm for simultaneously identifying and estimating counterfactual distributions. We show that this algorithm is sound and complete for deciding counterfactual identification in general settings. Third, considering the practical implications of these results, we introduce a new strategy for modeling NCMs using generative adversarial networks. Simulations corroborate with the proposed methodology.

1 INTRODUCTION

Counterfactual reasoning is one of human's high-level cognitive capabilities, used across a wide range of affairs, including determining how objects interact, assigning responsibility, credit and blame, and articulating explanations. Counterfactual statements underpin prototypical questions of the form "what if–" and "why–", which inquire about hypothetical worlds that have not necessarily been realized (Pearl & Mackenzie, 2018). If a patient Alice had taken a drug and died, one may wonder, "why did Alice die?"; "was it the drug that killed her?"; "would she be alive had she not taken the drug?". In the context of fairness, why did an applicant, Joe, not get the job offer? Would the outcome have changed had Joe been a Ph.D.? Or perhaps of a different race? These are examples of fundamental questions about *attribution* and *explanation*, which evoke hypothetical scenarios that disagree with the current reality and which not even experimental studies can reconstruct.

We build on the semantics of counterfactuals based on a generative process called *structural causal model* (SCM) (Pearl, 2000). A fully instantiated SCM $\mathcal{M}^*$ describes a collection of causal mechanisms and distribution over exogenous conditions. Each $\mathcal{M}^*$ induces families of qualitatively different distributions related to the activities of seeing (called observational), doing (interventional), and imagining (counterfactual), which together are known as the *ladder of causation* (Pearl & Mackenzie, 2018; Bareinboim et al., 2022); also called the *Pearl Causal Hierarchy* (PCH). The PCH is a containment hierarchy in which distributions can be put in increasingly refined layers: observational content goes into layer 1 ($\mathcal{L}_1$); experimental to layer 2 ($\mathcal{L}_2$); counterfactual to layer 3 ($\mathcal{L}_3$). It is understood that there are questions about layers 2 and 3 that cannot be answered (i.e. are underdetermined), even given all information in the world about layer 1; further, layer 3 questions are still underdetermined given data from layers 1 and 2 (Bareinboim et al., 2022; Ibeling & Icard, 2020).

Counterfactuals represent the more detailed, finest type of knowledge encoded in the PCH, so naturally, having the ability to evaluate counterfactual distributions is an attractive proposition. In practice, a fully specified model $\mathcal{M}^*$ is almost never observable, which leads to the question – how can a counterfactual statement, from $\mathcal{L}_3^*$, be evaluated using a combination of observational and experimental data (from $\mathcal{L}_1^*$ and $\mathcal{L}_2^*$)? This question embodies the challenge of cross-layer inferences, which entail solving two challenging causal problems in tandem, identification and estimation.

In the more traditional literature of causal inference, there are different symbolic methods for solving these problems in various settings and under different assumptions. In the context of identification, there exists an arsenal of results that includes celebrated methods such as Pearl's do-calculus (Pearl, 1995), and go through different algorithmic methods when considering inferences for $\mathcal{L}_2$- (Tian & Pearl, 2002; Shpitser & Pearl, 2006; Huang & Valtorta, 2006; Bareinboim & Pearl, 2012; Lee et al., 2019; Lee & Bareinboim, 2020; 2021) and $\mathcal{L}_3$-distributions (Heckman, 1992; Pearl, 2001; Avin et al., 2005; Shpitser & Pearl, 2009; Shpitser & Sherman, 2018; Zhang & Bareinboim, 2018; Correa et al., 2021). On the estimation side, there are various methods including the celebrated Propensity Score/IPW for the backdoor case (Rubin, 1978; Horvitz & Thompson, 1952; Kennedy, 2019; Kallus & Uehara, 2020), and some more relaxed settings (Fulcher et al., 2019; Jung et al., 2020; 2021), but the literature is somewhat scarcer and less developed. In fact, there is a lack of estimation methods for $\mathcal{L}_3$ quantities in most settings.

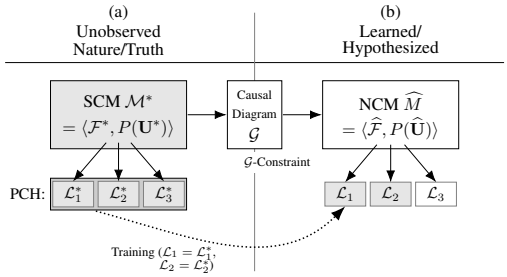


Figure 1: The l.h.s. contains the true SCM $\mathcal{M}^*$ that induces PCH's three layers. The r.h.s. contains a neural model $\widehat{M}$ constrained by inductive bias $\mathcal{G}$ (entailed by $\mathcal{M}^*$) and matching $\mathcal{M}^*$ on $\mathcal{L}_1$ and $\mathcal{L}_2$ through training.

On another thread in the literature, deep learning methods have achieved outstanding empirical success in solving a wide range of tasks in fields such as computer vision (Krizhevsky et al., 2012), speech recognition (Graves & Jaitly, 2014), and game playing (Mnih et al., 2013). One key feature of deep learning is its ability to allow inferences to scale with the data to high dimensional settings. We study here the suitability of the neural approach to tackle the problems of causal identification and estimation while trying to leverage the benefits of these new advances experienced in non-causal settings. [1] The idea behind the approach pursued here is illustrated in Fig. 1. Specifically, we will search for a neural model $\widehat{M}$ (r.h.s.) that has the same generative capability of the true, unobserved SCM $\mathcal{M}^*$ (l.h.s.); in other words, $\widehat{M}$ should be able to generate the same observed/inputted data, i.e., $\mathcal{L}_1 = \mathcal{L}_1^*$ and $\mathcal{L}_2 = \mathcal{L}_2^*$. [2] To tackle this task in practice, we use an inductive bias for the neural model in the form of a *causal diagram* (Pearl, 2000; Spirtes et al., 2000; Bareinboim & Pearl, 2016), which is a parsimonious description of the mechanisms ($\mathcal{F}^*$) and exogenous conditions ($P(\mathbf{U}^*)$) of the generating SCM. [3] The question then becomes: under what conditions can a model trained using this combination of qualitative inductive bias and the available data be suitable to answer questions about hypothetical counterfactual worlds, *as if* we had access to the true $\mathcal{M}^*$?

There exists a growing literature that leverages modern neural methods to solve causal inference tasks.[1] Our approach based on proxy causal models will answer causal queries by direct evaluation through a parameterized neural model $\widehat{M}$ fitted on the data generated by $\mathcal{M}^*$. [4] For instance, some recent work solves the estimation of interventional ($\mathcal{L}_2$) or counterfactual ($\mathcal{L}_3$) distributions from observational ($\mathcal{L}_1$) data in Markovian settings, implemented through architectures such as GANs, flows, GNNs, and VGAEs (Kocaoglu et al., 2018; Pawlowski et al., 2020; Zecevic et al., 2021; Sanchez-Martin et al., 2021). In some real-world settings, Markovianity is a too stringent condition (see discussion in App. D.4) and may be violated, which leads to the separation between layers 1 and 2, and, in turn, issues of causal identification. [5] The proxy approach discussed above was pursued in Xia et al. (2021) to solve the identification and estimation of interventional distributions ($\mathcal{L}_2$) from observational data ($\mathcal{L}_1$) in non-Markovian settings. [6] This work introduced an object we leverage throughout this paper called *Neural Causal Model* (NCM, for short), which is a class of SCMs constrained to neural network functions and fixed distributions over the exogenous variables. While

[1] One of our motivations is that these methods showed great promise at estimating effects from observational data under backdoor/ignorability conditions (Shalit et al., 2017; Louizos et al., 2017; Li & Fu, 2017; Johansson et al., 2016; Yao et al., 2018; Yoon et al., 2018; Kallus, 2020; Shi et al., 2019; Du et al., 2020; Guo et al., 2020).

[2] This represents an extreme case where all $\mathcal{L}_1$- and $\mathcal{L}_2$-distributions are provided as data. In practice, this may be unrealistic, and our method takes as input any arbitrary subset of distributions from $\mathcal{L}_1$ and $\mathcal{L}_2$.

[3] When imposed on neural models, they enforce equality constraints connecting layer 1 and layer 2 quantities, defined formally through the *causal Bayesian network* (CBN) data structure (Bareinboim et al., 2022, Def. 16).

[4] In general, $\widehat{M}$ does not need to, and will not be equal to the true SCM $\mathcal{M}^*$.

[5] Layer 3 differs from lower layers even in Markovian models; see Bareinboim et al. (2022, Ex. 7).

[6] Witty et al. (2021) shows a related approach taking the Bayesian route; further details, see Appendix C.

NCMs have been shown to be able to solve the identification and estimation tasks for $\mathcal{L}_2$ queries, their potential for counterfactual inferences is still largely unexplored, and existing implementations have been constrained to low-dimensional settings.

Despite all the progress achieved so far, no practical methods exist for estimating counterfactual ($\mathcal{L}_3$) distributions in the general setting where an arbitrary combination of observational ($\mathcal{L}_1$) and experimental ($\mathcal{L}_2$) distributions is available, and unobserved confounders exist (i.e. Markovianity does not hold). Hence, in addition to providing the first neural method of counterfactual identification, this paper establishes the first general counterfactual estimation technique even among non-neural methods, leveraging the neural toolkit for scalable inferences. Specifically, our contributions are:
1. We prove that when fitted with a graphical inductive bias, the NCM encodes the $\mathcal{L}_3$-constraints necessary for performing counterfactual inference (Thm. 1), and that they are still expressive enough to model the underlying data-generating model, which is not necessarily a neural network (Thm. 2).
2. We show that counterfactual identification within a neural proxy model setting is equivalent to established symbolic approaches (Thm. 3). We leverage this duality to create an algorithm (Alg. 1) for counterfactual identification and estimation that is both sound and complete (Corol. 2). The approach is general in that it accepts any combination of inputs from $\mathcal{L}_1$ and $\mathcal{L}_2$, it works in any causal diagram setting, and it does not require the Markovianity assumption to hold.
3. We develop a new approach to modeling the NCM using generative adversarial networks (GANs) (Goodfellow et al., 2014), capable of robustly scaling inferences to high dimensions (Alg. 3). We then show how GAN-NCMs can solve the challenging optimization problems in identifying and estimating counterfactuals in practice. Experiments are provided in Sec. 5 and proofs in Appendix A.

Preliminaries. We now introduce the notation and definitions used throughout the paper. We use uppercase letters (X) to denote random variables and lowercase letters (x) to denote corresponding values. Similarly, bold uppercase ($\mathbf{X}$) and lower case ($\mathbf{x}$) letters are used to denote sets of random variables and values respectively. 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 the domain of $\mathbf{X} = \{X_1, \ldots, X_k\}$. We denote $P(\mathbf{X} = \mathbf{x})$ (which we will often shorten to $P(\mathbf{x})$) as the probability of $\mathbf{X}$ taking the values $\mathbf{x}$ under the probability distribution $P(\mathbf{X})$.

We utilize the basic semantic framework of structural causal models (SCMs), as defined in (Pearl, 2000, Ch. 7). An SCM $\mathcal{M}$ consists of endogenous variables $\mathbf{V}$, exogenous variables $\mathbf{U}$ with distribution $P(\mathbf{U})$, and mechanisms $\mathcal{F}$. $\mathcal{F}$ contains a function f_{V_i} for each variable V_i that maps endogenous parents $\mathbf{Pa}_{V_i}$ and exogenous parents $\mathbf{U}_{V_i}$ to V_i. Each $\mathcal{M}$ induces a causal diagram $\mathcal{G}$, where every $V_i \in \mathbf{V}$ is a vertex, there is a directed arrow ($V_j \to V_i$) for every $V_i \in \mathbf{V}$ and $V_j \in \mathbf{Pa}_{V_i}$, 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, see (Bareinboim et al., 2022, Def. 13/16, Thm. 4). The exogenous $\mathbf{U}_{V_i}$'s are not assumed independent (i.e. Markovianity is not required). Our treatment is constrained to *recursive* SCMs, which implies acyclic causal diagrams, with finite domains over $\mathbf{V}$.

Each SCM $\mathcal{M}$ assigns values to each counterfactual distribution as follows:

Definition 1 (Layer 3 Valuation)**.** An SCM $\mathcal{M}$ induces layer $\mathcal{L}_3(\mathcal{M})$, a set of distributions over $\mathbf{V}$, each with the form $P(\mathbf{Y}_*) = P(\mathbf{Y}_{1[\mathbf{x}_1]}, \mathbf{Y}_{2[\mathbf{x}_2],\ldots})$ such that

$$P^{\mathcal{M}}(\mathbf{y}_{1[\mathbf{x}_1]}, \mathbf{y}_{2[\mathbf{x}_2]}, \ldots) = \int_{\mathcal{D}_{\mathbf{U}}} \mathbb{1}\left[\mathbf{Y}_{1[\mathbf{x}_1]}(\mathbf{u}) = \mathbf{y}_1, \mathbf{Y}_{2[\mathbf{x}_2]}(\mathbf{u}) = \mathbf{y}_2, \ldots\right] dP(\mathbf{u}), \tag{1}$$

where $\mathbf{Y}_{i[\mathbf{x}_i]}(\mathbf{u})$ is evaluated under $\mathcal{F}_{\mathbf{x}_i} := \{f_{V_j} : V_j \in \mathbf{V} \setminus \mathbf{X}_i\} \cup \{f_X \leftarrow x : X \in \mathbf{X}_i\}$. ■

Each $\mathbf{Y}_i$ corresponds to a set of variables in a world where the original mechanisms f_X are replaced with constants $\mathbf{x}_i$ for each $X \in \mathbf{X}_i$; this is also known as the mutilation procedure. This procedure corresponds to interventions, and we use subscripts to denote the intervening variables (e.g. $\mathbf{Y}_{\mathbf{x}}$) or subscripts with brackets when the variables are indexed (e.g. $\mathbf{Y}_{1[\mathbf{x}_1]}$). For instance, $P(y_x, y'_{x'})$ is the probability of the joint counterfactual event $Y = y$ had X been x and $Y = y'$ had X been x'.

SCM $\mathcal{M}_2$ is said to be $P^{(\mathcal{L}_i)}$-consistent (for short, $\mathcal{L}_i$-consistent) with SCM $\mathcal{M}_1$ if $\mathcal{L}_i(\mathcal{M}_1) = \mathcal{L}_i(\mathcal{M}_2)$. We will use $\mathbb{Z}$ to denote a set of quantities from Layer 2 (i.e. $\mathbb{Z} = \{P(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$), and we use $\mathbb{Z}(\mathcal{M})$ to denote those same quantities induced by SCM $\mathcal{M}$ (i.e. $\mathbb{Z}(\mathcal{M}) = \{P^{\mathcal{M}}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$).

We use neural causal models (NCMs) as a substitute (proxy) model for the true SCM, as follows:

Definition 2 ($\mathcal{G}$-Constrained Neural Causal Model ($\mathcal{G}$-NCM) (Xia et al., 2021, Def. 7))**.** Given a causal diagram $\mathcal{G}$, a $\mathcal{G}$-constrained Neural Causal Model (for short, $\mathcal{G}$-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 $\widehat{\mathbf{U}} = \{\widehat{U}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G})\}$, where $\mathbb{C}(\mathcal{G})$ is the set of all maximal cliques over bidirected edges of $\mathcal{G}$, and $\mathcal{D}_{\widehat{U}} = [0, 1]$ for all $\widehat{U} \in \widehat{\mathbf{U}}$; $\widehat{\mathcal{F}} = \{\hat{f}_{V_i} : V_i \in \mathbf{V}\}$, 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 $\mathbf{U}_{V_i} = \{\widehat{U}_{\mathbf{C}} : \widehat{U}_{\mathbf{C}} \in \widehat{\mathbf{U}} \text{ s.t. } V_i \in \mathbf{C}\}$ and $\mathbf{Pa}_{V_i} = Pa_{\mathcal{G}}(V_i)$; $\widehat{P}(\widehat{\mathbf{U}})$ is defined s.t. $\widehat{U} \sim \text{Unif}(0, 1)$ for each $\widehat{U} \in \widehat{\mathbf{U}}$. ■

2 Neural Causal Models for Counterfactual Inference

We first recall that inferences about higher layers of the PCH generated by the true SCM $\mathcal{M}^*$ cannot be made in general through an NCM $\widehat{M}$ trained only from lower layer data (Bareinboim et al., 2022; Xia et al., 2021). This impossibility motivated the use of the inductive bias in the form of a causal diagram $\mathcal{G}$ in the construction of the NCM in Def. 2, which ascertains that the $\mathcal{G}$-consistency property holds. (See App. D.1 for further discussion.) We next define consistency w.r.t. to each layer, which will be key for a more fine-grained discussion later on.

Definition 3 ($\mathcal{G}^{(\mathcal{L}_i)}$-Consistency)**.** Let $\mathcal{G}$ be the causal diagram induced by the SCM $\mathcal{M}^*$. For any SCM $\mathcal{M}$, $\mathcal{M}$ is said to be $\mathcal{G}^{(\mathcal{L}_i)}$-consistent (w.r.t. $\mathcal{M}^*$) if $\mathcal{L}_i(\mathcal{M})$ satisfies all layer i equality constraints implied by $\mathcal{G}$. ■

This generalization is subtle since regardless of which $\mathcal{L}_i$ is used with the definition, the causal diagram $\mathcal{G}$ generated by $\mathcal{M}^*$ is the same. The difference lies in the implied constraints. For instance, if an SCM $\mathcal{M}$ is $\mathcal{G}^{(\mathcal{L}_1)}$-consistent, that means that $\mathcal{G}$ is a Bayesian network for the observational distribution of $\mathcal{M}$, implying independences readable through d-separation Pearl (1988). If $\mathcal{M}$ is $\mathcal{G}^{(\mathcal{L}_2)}$-consistent, that means that $\mathcal{G}$ is a *Causal Bayesian network* (CBN) (Bareinboim et al., 2022, Def. 16) for the interventional distributions of $\mathcal{M}$. While several SCMs could share the same d-separation constraints as $\mathcal{M}^*$, there are fewer that share all $\mathcal{L}_2$ constraints encoded by the CBN. $\mathcal{G}$-consistency at higher layers imposes a stricter set of constraints, narrowing down the set of compatible SCMs. There also exist constraints of layer 3 that are important for counterfactual inferences.

To motivate the use of such constraints, consider an example inspired by the multi-armed bandit problem. A casino has 3 slot machines, labeled "0", "1", and "2". Every day, the casino assigns one machine a good payout, one a bad payout, and one an average payout, with chances of winning represented by exogenous variables U_+, U_-, and $U_=$, respectively. A customer comes every day and plays a slot machine. X represents their choice of machine, and Y is a binary variable representing whether they win. Suppose a data scientist creates a model of the situation, and she hypothesizes that the casino predicts the customer's choice based on their mood (U_M) and will always assign the predicted machine the average payout to maintain profits. Her model is described by the SCM $\mathcal{M}'$:

$$\mathcal{M}' = \begin{cases} \mathbf{U} & = \{U_M, U_+, U_=, U_-\}, U_M \in \{0,1,2\}, U_+, U_=, U_- \in \{0,1\} \\ \mathbf{V} & = \{X, Y\}, X \in \{0,1,2\}, Y \in \{0,1\} \\ \mathcal{F} & = \begin{cases} f_X(u_M) & = u_M \\ f_Y(x, u_M, u_+, u_=, u_-) & = \begin{cases} u_= & x = u_M \\ u_- & x = (u_M - 1)\%3 \\ u_+ & x = (u_M + 1)\%3 \end{cases} \end{cases} \\ P(\mathbf{U}): & P(U_M = i) = \frac{1}{3}, P(U_+ = 1) = 0.6, P(U_= = 1) = 0.4, P(U_- = 1) = 0.2 \end{cases} \tag{2}$$

It turns out that in this model $P(y_x) = P(y \mid x)$. For example, $P(Y = 1 \mid X = 0) = P(U_= = 1) = 0.4$, and $P(Y_{X=0} = 1) = P(U_M = 0)P(U_= = 1) + P(U_M = 1)P(U_- = 1) + P(U_M = 2)P(U_+ = 1) = \frac{1}{3}(0.4) + \frac{1}{3}(0.2) + \frac{1}{3}(0.6) = 0.4$.

Suppose the true model $\mathcal{M}^*$ employed by the casino (and unknown by the customers and data scientist) induces graph $\mathcal{G} = \{X \to Y\}$. Interestingly enough, $\mathcal{M}'$ would be $\mathcal{G}^{(\mathcal{L}_2)}$-consistent with $\mathcal{M}^*$ since $\mathcal{M}'$ is compatible with all $\mathcal{L}_2$-constraints, including $P(y_x) = P(y \mid x)$ and $P(x_y) = P(x)$. However, and perhaps surprisingly, it would fail to be $\mathcal{G}^{(L_3)}$-consistent. A further constraint implied by $\mathcal{G}$ on the third layer is that $P(y_x \mid x') = P(y_x)$, which is not true of $\mathcal{M}'$. To witness, note that $P(Y_{X=0} = 1 \mid X = 2) = P(U_+ = 1) = 0.6$ in $\mathcal{M}'$, which means that if the customer chose machine 2, they would have had higher payout had they chosen machine 0. This does not match $P(Y_{X=0} = 1) = 0.4$, computed earlier, so $\mathcal{M}'$ fails to encode the $\mathcal{L}_3$-constraints implied by $\mathcal{G}$.

In general, the causal diagram encodes a family of $\mathcal{L}_3$-constraints which we leverage to make cross-layer inferences. A more detailed discussion can be found in Appendix D. We show next that NCMs encodes all of the equality constraints related to $\mathcal{L}_3$, in addition to the known $\mathcal{L}_2$-constraints.

Theorem 1 (NCM $\mathcal{G}^{(\mathcal{L}_3)}$-Consistency)**.** *Any $\mathcal{G}$-NCM $\widehat{M}(\boldsymbol{\theta})$ is $\mathcal{G}^{(\mathcal{L}_3)}$-consistent.* ■

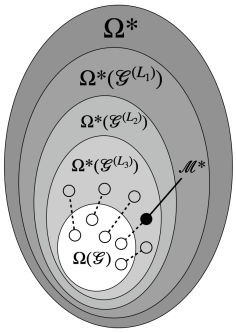


Figure 2: Model-theoretic visualization of Thms. 1 and 2.

This will be a key result for performing inferences at the counterfactual level. Similar to how constraints about layer 2 distributions help bridge the gap between layers 1 and 2, layer 3 constraints allow us to extend our inference capabilities into layer 3. (In fact, most of $\mathcal{L}_3$'s distributions are not obtainable through experimentation.) While this graphical inductive bias is powerful, the set of NCMs constrained by $\mathcal{G}$ is no less expressive than the set of SCMs constrained by $\mathcal{G}$, as shown next.

Theorem 2 ($\mathcal{L}_3$-$\mathcal{G}$ Expressiveness)**.** *For any SCM $\mathcal{M}^*$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-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 $\mathcal{L}_3$-consistent w.r.t. $\mathcal{M}^*$.* ■

This result ascertains that the NCM class is as expressive, and therefore, contains the same generative capabilities as the original generating model. More interestingly, even if the original SCM $\mathcal{M}^*$ does not belong to the NCM class, but from the higher space, there exists a NCM $\widehat{M}(\boldsymbol{\theta})$ that will be capable of expressing the collection of distributions from all layers of the PCH induced by it.

A visual representation of these two results is shown in Fig. 2. The space of all SCMs is called Ω^*, and the subspace that contains all SCMs $\mathcal{G}^{((\mathcal{L}_i)}$-consistent w.r.t. the true SCM $\mathcal{M}^*$ (black dot) is called $\Omega^*(\mathcal{G}^{(\mathcal{L}_i)})$. Note that the $\mathcal{G}^{(\mathcal{L}_i)}$ space shrinks with higher layers, indicating a more constrained space with fewer SCMs. Thm. 1 states that all $\mathcal{G}$-NCMs ($\Omega(\mathcal{G})$) are within $\Omega^*(\mathcal{G}^{(\mathcal{L}_3)})$, and Thm. 2 states that all SCMs in $\Omega^*(\mathcal{G}^{(\mathcal{L}_3)})$ can be represented by a corresponding $\mathcal{G}$-NCM on all three layers.

It may seem intuitive that the $\mathcal{G}$-NCM has these two properties by construction, but these properties are nontrivial and, in fact, not enjoyed by many model classes. Examples can be found in Appendix D. Together, these two theorems ensure that the NCM has both the constraints and the expressiveness necessary for counterfactual inference, elaborated further in the next section.

3 Neural Counterfactual Identification

The problem of identification is concerned with determining whether a certain quantity is computable from a combination of assumptions, usually encoded in the form of a causal diagram, and a collection of distributions (Pearl, 2000, p. 77). This challenge stems from the fact that even though the space of SCMs (or NCMs) is constrained upon assuming a certain causal diagram, the quantity of interest may still be underdetermined. In words, there are many SCMs compatible with the same diagram $\mathcal{G}$ but generate different answers for the target distribution. In this section, we investigate the problem of identification and decide whether counterfactual quantities (from $\mathcal{L}_3$) can be inferred from a combination of a subset of $\mathcal{L}_2$ and $\mathcal{L}_1$ datasets together with $\mathcal{G}$, as formally defined next.

Definition 4 (Neural Counterfactual Identification)**.** Consider an SCM $\mathcal{M}^*$ and the corresponding causal diagram $\mathcal{G}$. Let $\mathbb{Z} = \{P(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$ be a collection of available interventional (or observational if $\mathbf{Z}_k = \emptyset$) distributions from $\mathcal{M}^*$. The counterfactual query $P(\mathbf{Y}_* = \mathbf{y}_* \mid \mathbf{X}_* = \mathbf{x}_*)$ is said to be neural identifiable (identifiable, for short) from the set of $\mathcal{G}$-constrained NCMs $\Omega(\mathcal{G})$ and $\mathbb{Z}$ if and only if $P^{\widehat{M}_1}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\widehat{M}_2}(\mathbf{y}_* \mid \mathbf{x}_*)$ for every pair of models $\widehat{M}_1, \widehat{M}_2 \in \Omega(\mathcal{G})$ s.t. they match $\mathcal{M}^*$ on all distributions in $\mathbb{Z}$ (i.e. $\mathbb{Z}(\mathcal{M}^*) = \mathbb{Z}(\mathcal{M}_1) = \mathbb{Z}(\mathcal{M}_2) > 0$). ■

From a symbolic standpoint, a counterfactual quantity $P(\mathbf{y}_* \mid \mathbf{x}_*)$ is identifiable from $\mathcal{G}$ and $\mathbb{Z}$ if all SCMs that induce the distributions of $\mathbb{Z}$ and abide by the constraints of $\mathcal{G}$ also agree on $P(\mathbf{y}_* \mid \mathbf{x}_*)$. This is illustrated in Fig. 3. In the definition above, the search is constrained to the NCM subspace (shown in light gray) within the space of SCMs (dark gray). It may be concerning that the true SCM $\mathcal{M}^*$ might not be an NCM, as we alluded to earlier. The next result ascertains that identification within the constrained space of NCMs is actually equivalent to identification in the original SCM-space.

Theorem 3 (Counterfactual Graphical-Neural Equivalence (Dual ID))**.** *Let Ω^*, Ω be the spaces including all SCMs and NCMs, respectively. Consider the true SCM $\mathcal{M}^*$ and the corresponding*

causal diagram $\mathcal{G}$. Let $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$ be the target query and $\mathbb{Z}$ the set of observational and interventional distributions available. Then, Q is neural identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$ if and only if it is identifiable from $\mathcal{G}$ and $\mathbb{Z}$. ■

Interestingly, this result connects the new concept of neural counterfactual identification (Def. 4) with established non-neural results. If a counterfactual quantity is determined to be neural identifiable, then it is also identifiable from $\mathcal{G}$ and $\mathbb{Z}$ through other non-neural means, and vice versa.[7] Practically speaking, counterfactual inference can be performed while constrained in the NCM space, and the obtained results will be faithful to existing symbolic approaches. This broadens the previous results connecting NCMs to classical identification.

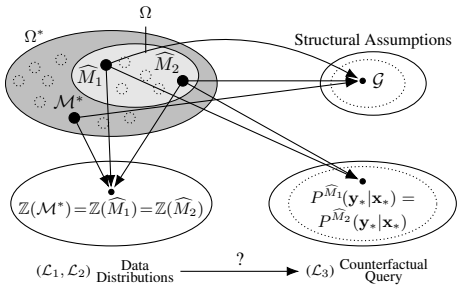


Figure 3: $P(\mathbf{y}_*)$ is identifiable from $\mathbb{Z}$ and $\Omega(\mathcal{G})$ if for any SCM $\mathcal{M}^* \in \Omega^*$ and NCMs $\widehat{M}_1, \widehat{M}_2 \in \Omega$ (top left), $\widehat{M}_1, \widehat{M}_2, \mathcal{M}^*$ match in $\mathbb{Z}$ (bottom left) and $\mathcal{G}$ (top right), then the NCMs $\widehat{M}_1$, $\widehat{M}_2$ also match in $P(\mathbf{y}_*)$ (bottom right).

Corollary 1 (Neural Counterfactual Mutilation (Operational ID))**.** *Consider the true SCM $\mathcal{M}^* \in \Omega^*$, causal diagram $\mathcal{G}$, a set of available distributions $\mathbb{Z}$, and a target query Q equal to $P^{\mathcal{M}^*}(\mathbf{y}_* \mid \mathbf{x}_*)$. Let $\widehat{M} \in \Omega(\mathcal{G})$ be a $\mathcal{G}$-constrained NCM such that $\mathbb{Z}(\widehat{M}) = \mathbb{Z}(\mathcal{M}^*)$. If Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$, then Q is computable via Eq. 1 from $\widehat{M}$.* ■

Corol. 1 states that once identification is established, the counterfactual query can be inferred through the NCM $\widehat{M}$, as if it were the true SCM $\mathcal{M}^*$, by directly applying layer 3's definition to $\widehat{M}$ (Def. 1). Remarkably, this result holds even if $\mathcal{M}^*$ does not match $\widehat{M}$ in either the mechanisms $\mathcal{F}$ or the exogenous dist. $P(\mathbf{U})$, and it only requires some specific properties: $\mathcal{G}^{(\mathcal{L}_3)}$-consistency, matching $\mathbb{Z}$, and identifiability. Without these properties, inferences performed on $\widehat{M}$ would bear no meaningful information about the ground truth. To understand this subtlety, refer to examples in App. D.

Building on these results, we demonstrate through the procedure **NeuralID** (Alg. 1) how to decide the identifiability of counterfactual quantities. The specific optimization procedure searches explicitly in the space of NCMs for two models that respectively minimize and maximize the target query while maintaining consistency with the provided data distributions in $\mathbb{Z}$. If the two models match in the target query Q, then the effect is identifiable, and the value is returned; otherwise, the effect is non-identifiable.

Algorithm 1: **NeuralID** – Identifying/estimating counterfactual queries with NCMs.

Input : query $Q = P(\mathbf{y}_*|\mathbf{x}_*)$, $\mathcal{L}_2$ datasets $\mathbb{Z}(\mathcal{M}^*)$, and causal diagram $\mathcal{G}$

Output : $P^{\mathcal{M}^*}(\mathbf{y}_*|\mathbf{x}_*)$ if identifiable, `FAIL` otherwise.

$\widehat{M} \leftarrow$ `NCM`$(\mathbf{V}, \mathcal{G})$ `// from Def. 2`

$\boldsymbol{\theta}^*_{\min} \leftarrow \arg\min_{\boldsymbol{\theta}} P^{\widehat{M}(\boldsymbol{\theta})}(\mathbf{y}_*|\mathbf{x}_*)$ s.t. $\mathbb{Z}(\widehat{M}(\boldsymbol{\theta})) = \mathbb{Z}(\mathcal{M}^*)$

$\boldsymbol{\theta}^*_{\max} \leftarrow \arg\max_{\boldsymbol{\theta}} P^{\widehat{M}(\boldsymbol{\theta})}(\mathbf{y}_*|\mathbf{x}_*)$ s.t. $\mathbb{Z}(\widehat{M}(\boldsymbol{\theta})) = \mathbb{Z}(\mathcal{M}^*)$

if $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y}_*|\mathbf{x}_*) \neq P^{\widehat{M}(\boldsymbol{\theta}^*_{\max})}(\mathbf{y}_*|\mathbf{x}_*)$ **then**

  return `FAIL`

else

  return $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y}_*|\mathbf{x}_*)$ `// choose min or max arbitrarily`

The implementation of how to enforce these consistency constraints in practice is somewhat challenging. We note two nontrivial details that are abstracted away in the description of Alg. 1. First, although training to fit a single, observational dataset is straightforward, it is not as clear how to simultaneously maintain consistency with the multiple datasets in $\mathbb{Z}$. Second, unlike with simpler interventional queries, it is not clear how to search the parameter space in a way that maximizes or minimizes a counterfactual query, which may be more involved due to nesting (e.g. $P(Y_{Z_{X=0}})$) or evaluating the same variable in multiple worlds (e.g. $P(Y_{X=0}, Y_{X=1})$). The details of how to solve these issues are discussed in Sec. 4.

Interestingly, this approach is qualitatively different than the case of classical, symbolic methods that avoid operating in the space of SCMs directly. Still, in principle, this alternative approach does not imply any loss in functionality, as evident from the next result.

Corollary 2 (Soundness and Completeness)**.** *Let Ω^* be the set of all SCMs, $\mathcal{M}^* \in \Omega^*$ be the true SCM inducing causal diagram $\mathcal{G}$, $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$ be a query of interest, and $\widehat{Q}$ be the result from*

[7] We say identification from $\mathcal{G}$ and $\mathbb{Z}$ instead of $\Omega^*(\mathcal{G})$ and $\mathbb{Z}$ because existing symbolic approaches (e.g. do-calculus) directly solve the identification problem on top of the graph instead of the space of SCMs.

running Alg. 1 with inputs $\mathbb{Z}(\mathcal{M}^*) > 0$, $\mathcal{G}$, 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 $\widehat{Q} = P^{\mathcal{M}^*}(\mathbf{y}_* \mid \mathbf{x}_*)$.* ■

In words, the procedure **NeuralID** is both necessary and sufficient in this very general setting, implying that for any instances involving any arbitrary diagram, datasets, or queries, the identification status of the query is always classified correctly by this algorithm.

4 Neural Counterfactual Estimation

We developed a procedure for identifying counterfactual quantities that is both sound and complete under ideal conditions – e.g., unlimited data, perfect optimization, which is encouraging. In this section, we build on these results and establish a more practical approach to solving these tasks under imperfect optimization and finite samples.

Algorithm 2: NCM Counterfactual Sampling

Input : NCM $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$, counterfactual $\mathbf{Y}_*$, conditional $\mathbf{X}_* = \mathbf{x}_*$, number of samples m

Output : m samples from $P^{\widehat{M}}(\mathbf{Y}_* | \mathbf{x}_*)$

```
Function M̂.sample(Y*, x*, m):
    S ← ∅
    while |S| < m do
        û ← P(Û).sample()
        if X*^M̂(θ)(û) = x* then
            S.add(Y*^M̂(θ)(û))
    return S;
```

Consider a $\mathcal{G}$-NCM $\widehat{M}$ constructed as specified by Def. 2. Any counterfactual statement $\mathbf{Y}_* = (\mathbf{Y}_{1[\mathbf{x}_1]}, \mathbf{Y}_{2[\mathbf{x}_2]}, \dots)$ can be evaluated from an NCM $\widehat{M}$ for a specific setting of $\widehat{\mathbf{U}} = \widehat{\mathbf{u}}$ by computing the corresponding values of $\mathbf{Y}_i$ in the mutilated submodel $\widehat{M}_{\mathbf{x}}$ for each i. That is,

$$\mathbf{Y}_*^{\widehat{M}}(\mathbf{u}) = (\mathbf{Y}_{1[\mathbf{x}_1]}^{\widehat{M}}(\mathbf{u}), \mathbf{Y}_{2[\mathbf{x}_2]}^{\widehat{M}}(\mathbf{u}), \dots) \tag{3}$$

Then, sampling can be done following the natural approach delineated by Alg. 2. In words, the distribution $P(\mathbf{Y}_* \mid \mathbf{x}_*)$ can be sampled from NCM $\widehat{M}$ by (1) sampling instances of $P(\widehat{\mathbf{U}})$, (2) computing their corresponding value for $\mathbf{X}_*$ (via Eq. 3) while rejecting cases that do not match $\mathbf{x}_*$, and (3) returning the corresponding value for $\mathbf{Y}_*$ (via Eq. 3) for the remaining instances.

Following this procedure, a counterfactual $P(\mathbf{Y}_* = \mathbf{y}_* \mid \mathbf{X}_* = \mathbf{x}_*)$ can be estimated from the NCM through a Monte-Carlo approach, instantiating Eq. 4 as follows:

$$P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) \approx \frac{\sum_{j=1}^{m} \mathbb{1}\left[\mathbf{Y}_*^{\widehat{M}}(\widehat{\mathbf{u}}_j) = \mathbf{y}_*, \mathbf{X}_*^{\widehat{M}}(\widehat{\mathbf{u}}_j) = \mathbf{x}_*\right]}{\sum_{j=1}^{m} \mathbb{1}\left[\mathbf{X}_*^{\widehat{M}}(\widehat{\mathbf{u}}_j) = \mathbf{x}_*\right]}, \tag{4}$$

where $\{\widehat{\mathbf{u}}_j\}_{j=1}^m$ are a set of m samples from $P(\widehat{\mathbf{U}})$.

Alg. 3 demonstrates how to solve the challenging optimization task in lines 2 and 3 of Alg. 1. The first step is to learn parameters such that the distributions induced by the NCM $\widehat{M}$ match the true distributions in $\mathbb{Z}$. While Alg. 1 describes the inputs in the form of $\mathcal{L}_2$-distributions, $\mathbb{Z}(\mathcal{M}^*) = \{P^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$, in most settings, one has the empirical versions of such distributions in the form of finite datasets, $\{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k}) = \{\mathbf{v}_{\mathbf{z}_k,i}\}_{i=1}^{n_k}\}_{k=1}^{\ell}$.

One way to train $\widehat{M}$ to match $\mathcal{M}^*$ in the distribution $P(\mathbf{V}_{\mathbf{z}_k})$ is to compare the distribution of data points in $\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})$ with the distribution of samples from $\widehat{M}$, $\widehat{P}^{\widehat{M}}(\mathbf{V}_{\mathbf{z}_k})$. The two empirical distributions can be compared using a divergence function $\mathbb{D}_P$, which returns a smaller value when the two distributions are "similar". The goal is then to minimize $\mathbb{D}_P(\widehat{P}^{\widehat{M}}(\mathbf{V}_{\mathbf{z}_k}), \widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k}))$ for each $k \in \{1, \dots, \ell\}$. In this work, a generative adversarial approach (Goodfellow et al., 2014) is taken to train the NCM, and $\mathbb{D}_P$ is computed using a discriminator network.

In addition to fitting the datasets, the second challenge of Alg. 1 is to simultaneously maximize or minimize the query of interest $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$. This can be done by first computing samples of $P(\mathbf{Y}_* \mid \mathbf{x}_*)$ from $\widehat{M}$ via Alg. 2, denoted $\widehat{Q}$, and then minimizing (or maximizing) the "distance" between Q and $\widehat{Q}$. Essentially, samples from $\mathbf{Y}_*$ are penalized based on how similar they are to the correct values $\mathbf{y}_*$. For example, if the query to maximize is $P(Y = 1)$, and a value of 0.6 is sampled for Y from $\widehat{M}$, then the goal could be to minimize squared error, $(1 - 0.6)^2$. In general, a distance metric $\mathbb{D}_Q$ is used to compute the distance between $\widehat{Q}$ and Q, and we use log loss for $\mathbb{D}_Q$ as our experiments involve binary variables.

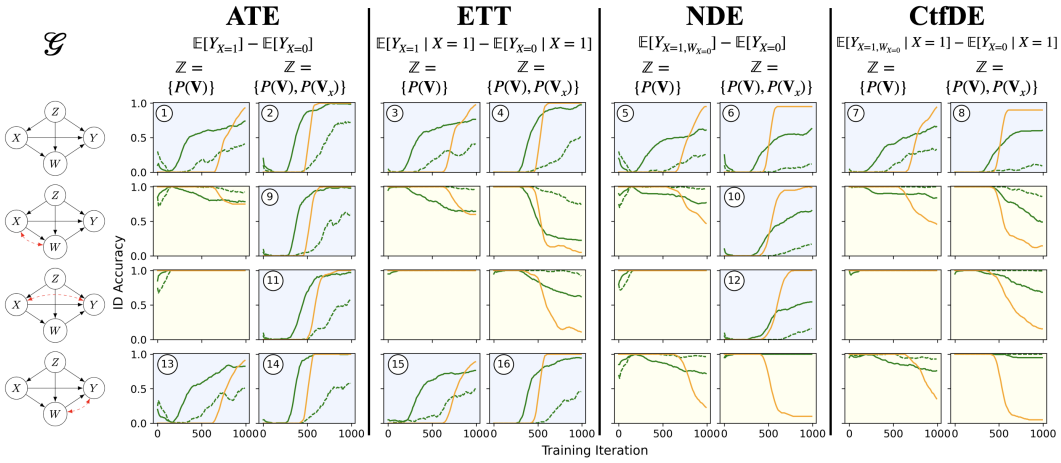


Figure 4: Experimental results on deciding identifiability on counterfactual queries with NCMs. GAN-NCM (green) is compared with MLE-NCM (orange) for settings with $d = 1$. GAN-NCM performance is also shown for $d = 16$ (dashed green). Blue (resp. yellow) backgrounds on plots correspond to a ground truth of ID (resp. non-ID). ID cases are numbered for reference in later plots.

For this reason, an NCM trained with this approach will be referred as a *GAN-NCM*. [8] More details about architecture and hyperparameters used throughout this work can be found in Appendix B.

Putting $\mathbb{D}_P$ and $\mathbb{D}_Q$ together, we can write that the objective $L\left(\widehat{M}, \{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}\right)$ is

$$\left(\sum_{k=1}^{\ell} \mathbb{D}_P\left(\widehat{P}^{\widehat{M}}_{\mathbf{z}_k}, \widehat{P}^{\mathcal{M}^*}_{\mathbf{z}_k}\right)\right) \pm \lambda \mathbb{D}_Q\left(\widehat{Q}, Q\right), \tag{5}$$

where λ is initially set to a high value, and decreases during training. Optimization may be done using gradient descent. After training, the two values of Q induced by $\widehat{M}(\boldsymbol{\theta}_{\min})$ and $\widehat{M}(\boldsymbol{\theta}_{\max})$ are compared with a hypothesis testing procedure to decide identifiability. Eq. 4 is used as Q's estimate, whenever identifiable.

Algorithm 3: Training Model

Input : Data $\{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k}) = \{\mathbf{v}_{\mathbf{z}_k,i}\}_{i=1}^{n_k}\}_{k=1}^{\ell}$, query $Q = P(\mathbf{y}_*|\mathbf{x}_*)$, causal diagram $\mathcal{G}$, number of Monte Carlo samples m, regularization constant λ, learning rate η, training epochs T

$\widehat{M} \leftarrow$ `NCM`$(\mathbf{V}, \mathcal{G})$ `// from Def. 2`
Initialize parameters $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$
for $t \leftarrow 1$ **to** T **do**
  $L_{\min} \leftarrow 0, L_{\max} \leftarrow 0$
  for $k \leftarrow 1$ **to** ℓ **do**
    `// Sample via Alg. 2`
    $\widehat{P}_{\min}(\mathbf{V}_{\mathbf{z}_k}) \leftarrow \widehat{M}(\boldsymbol{\theta}_{\min})$`.sample(`$\mathbf{V}_{\mathbf{z}_k}, n_k$`)`
    $\widehat{P}_{\max}(\mathbf{V}_{\mathbf{z}_k}) \leftarrow \widehat{M}(\boldsymbol{\theta}_{\max})$`.sample(`$\mathbf{V}_{\mathbf{z}_k}, n_k$`)`
    $L_{\min} \leftarrow L_{\min} + \mathbb{D}_P\left(\widehat{P}_{\min}(\mathbf{V}_{\mathbf{z}_k}), \widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\right)$
    $L_{\max} \leftarrow L_{\max} + \mathbb{D}_P\left(\widehat{P}_{\max}(\mathbf{V}_{\mathbf{z}_k}), \widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\right)$
  $\widehat{Q}_{\min} \leftarrow \widehat{M}(\boldsymbol{\theta}_{\min})$`.sample(`$\mathbf{Y}_*, m$`)`
  $\widehat{Q}_{\max} \leftarrow \widehat{M}(\boldsymbol{\theta}_{\max})$`.sample(`$\mathbf{Y}_*, m$`)`
  `// L from Eq. 5`
  $L_{\min} \leftarrow L_{\min} - \lambda \mathbb{D}_Q\left(\widehat{Q}_{\min}, Q\right)$
  $L_{\max} \leftarrow L_{\max} + \lambda \mathbb{D}_Q\left(\widehat{Q}_{\max}, Q\right)$
  $\boldsymbol{\theta}_{\min} \leftarrow \boldsymbol{\theta}_{\min} - \eta \nabla L_{\min}$
  $\boldsymbol{\theta}_{\max} \leftarrow \boldsymbol{\theta}_{\max} - \eta \nabla L_{\max}$

5 Experimental Evaluation

We first evaluate the NCM's ability to identify counterfactual distributions through Alg. 3. Each setting consists of a target query (Q), a causal diagram ($\mathcal{G}$), and a set of input distributions ($\mathbb{Z}$). In total, we test 32 variations. Specifically, we evaluate the identifiability of four queries Q: (1) Average Treatment Effect (ATE), (2) Effect of Treatment on the Treated (ETT) (Pearl, 2000, Eq. 8.18), (3) Natural Direct Effect (NDE) (Pearl, 2001, Eq. 6), and (4) Counterfactual Direct Effect (CtfDE) (Zhang & Bareinboim, 2018, Eq. 3); each expression is shown on the top of Fig. 4. The four graphs used are shown on the figure's left side, and represent general structures found throughout the mediation and fairness literature (Pearl, 2001; Zhang & Bareinboim, 2018). The variable X encodes the treatment/decision, Y the outcome, Z observed features, and W mediating variables. Lastly, we consider a setting in which only the observational data is available ($\mathbb{Z} = \{P(\mathbf{V})\}$) and another in which additional experimental data on X is available ($\mathbb{Z} = \{P(\mathbf{V}), P(\mathbf{V}_x)\}$). In the experiments shown, all variables are 1-dimensional binary variables except Z, whose dimensionality d is adjusted

[8] Other choices of $\mathbb{D}_P$ include KL divergence, f-divergence, or Maximum Mean Discrepancy (MMD) (Gretton et al., 2012).

in experiments. The background color of each setting indicates that the query Q is identifiable (blue) or is not identifiable (yellow) from the inputted $\mathcal{G}$ and $\mathbb{Z}$. Given the sheer volume of variations, we summarize the experiments below and provide further discussion and details in Appendix B.

We implement two approaches to ground the discussion around NCMs, one based on GANs (*GAN-NCM*), and another based on maximum likelihood (*MLE-NCM*). The former was discussed in the previous section and the latter is quite natural in statistical settings. The experiments (Fig. 4) show that GAN-NCM has on average higher accuracy. The MLE-NCM performs slightly better in ID cases (blue), but the performance drops significantly for non-ID cases (yellow), suggesting it may be biased in returning ID for all cases. The GAN-NCM is also shown to achieve decent performance in 16-d, where the MLE-NCM fails to work. We plot the run time of these two approaches in Fig. 5, which shows that the MLE-NCM scales poorly compared to the GAN-NCM; this pattern is observed in all settings. Intuitively, this is not surprising since the MLE-NCM explicitly computes a likelihood for every value in every variable domain, the size of which grows exponentially w.r.t. the dimensionality (d), while the GAN-NCM avoids this by implicitly fitting distributions through $P(\widehat{\mathbf{U}})$ and $\widehat{\mathcal{F}}$ and directly outputting samples.

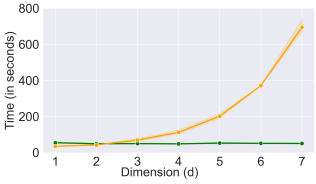


Figure 5: Results comparing run times of 100 epochs of training between GAN-NCM (green) and MLE-NCM (orange) in the first graph of Fig. 4 as the dimensionality d of Z scales higher.

For the identifiable cases (blue background), the target Q is estimated through Eq. 4 after training. Results are shown in Fig. 6. The MLE-NCM serves as a benchmark for 1-dimensional cases since, intuitively, the data distributions can be learned more accurately when modeled explicitly. Still, even when $d = 1$, the GAN-NCM achieves competitive results in most settings and consistently achieves an error under 0.05 with more samples. The GAN-NCM is able to maintain this consistency even at $d = 16$, demonstrating its robustness scaling to higher dimensions. The code will be made available after publication.

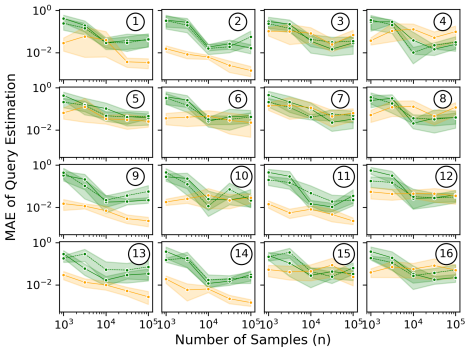


Figure 6: Results on estimating identifiable cases from Fig. 4 (corresponding numbers shown on the right). Mean Absolute Error (MAE) is plotted (with 95% confidence) for each setting for varying sample sizes. Results are shown for GAN-NCM (solid green) and MLE-NCM (orange) with $d = 1$ and also GAN-NCM with $d = 16$ (dashed green).

After all, the GAN-NCM is shown to be effective at identifying and estimating counterfactual distributions even in high dimensions. As expected, the MLE-NCM may achieve lower error in some 1-d settings, but the GAN-NCM may be preferred for scalability. Moreover, an incorrect ID conclusion in a non-ID case may be dangerous for downstream decision-making as the resulting estimation will likely be incorrect or misleading. The GAN-NCM is evidently more robust in such non-ID cases while still performing competitively in ID cases. Further experiments and discussions are provided in App. B.

6 Conclusions

We developed in this work a neural approach to the problems of counterfactual identification and estimation using neural causal models (NCMs). Specifically, we first showed that with the graphical inductive bias, NCMs are capable of encoding counterfactual ($\mathcal{L}_3$) constraints and are still expressive so as to represent any generating SCM (Thms. 1, 2). We then showed that NCMs have the ability of solving any counterfactual identification instance (Thm. 3, Corol 1). Given these theoretical properties, we introduced a sound and complete algorithm (Alg. 1, Corol. 2) for identifying and estimating counterfactuals in general non-Markovian settings given arbitrary datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$. We developed an approach based on GANs to implement this algorithm in practice (Alg. 3) and empirically demonstrated its ability to scale inferences. From a neural perspective, counterfactual reasoning under a causal inductive bias allows for deep models to be trained with an improved understanding of interpretability and generalizability. From a causal perspective, neural nets can now provide tools to solve counterfactual inference problems previously only understood in theory.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

C Avin, Ilya Shpitser, and Judea Pearl. Identifiability of Path-Specific Effects. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pp. 357–363, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Elias Bareinboim and Judea Pearl. Causal Inference by Surrogate Experiments: z-Identifiability. In Nando de Freitas Murphy and Kevin (eds.), *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 113–120. AUAI Press, 2012.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

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

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14 (11), 2013.

Juan Correa, Sanghack Lee, and Elias Bareinboim. Nested counterfactual identification from arbitrary surrogate experiments. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6856–6867. Curran Associates, Inc., 2021.

Xin Du, Lei Sun, Wouter Duivesteijn, Alexander Nikolaev, and Mykola Pechenizkiy. Adversarial balancing-based representation learning for causal effect inference with observational data, 2020.

William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020.

Isabel R Fulcher, Ilya Shpitser, Stella Marealle, and Eric J Tchetgen Tchetgen. Robust inference on population indirect causal effects: the generalized front door criterion. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2019.

David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundation of Science*, 3(1):151–182, 1998.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 881–889, Lille, France, 07–09 Jul 2015. PMLR.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, Inc., 2014.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

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

Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. A survey of learning causality with data. *ACM Computing Surveys*, 53(4):1–37, Sep 2020. ISSN 1557-7341. doi: 10.1145/3397269.

J J Heckman. Randomization and Social Policy Evaluation. In C Manski and I Garfinkle (eds.), *Evaluations: Welfare and Training Programs*, pp. 201–230. Harvard University Press, Cambridge, MA, 1992.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Yimin Huang and Marco Valtorta. Pearl's calculus of interventions is complete. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI2006)*, 2006.

Duligur Ibeling and Thomas Icard. Probabilistic reasoning across the causal hierarchy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10170–10177, 2020.

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

A. Jaber, J. Zhang, and E. Bareinboim. Identification of conditional causal effects under Markov equivalence. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 11512–11520. Curran Associates, Inc., 2019.

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

A. Jaber, A. Ribeiro, J. Zhang, and E. Bareinboim. Causal identification under markov equivalence: Calculus, algorithm, and completeness. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022, forthcoming.

Fredrik D. Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 3020–3029. JMLR.org, 2016.

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

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

Nathan Kallus. DeepMatch: Balancing deep covariate representations for causal inference using adversarial training. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5067–5077. PMLR, 13–18 Jul 2020.

Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. In *Proceedings of the 38th International Conference on Machine Learning*, 2020.

Andrej Karpathy. pytorch-made. `https://github.com/karpathy/pytorch-made` [Source Code], 2018.

Edward H Kennedy. Nonparametric causal effects based on incremental propensity score interventions. *Journal of the American Statistical Association*, 114(526):645–656, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Experimental design for learning causal graphs with latent variablesjaber. In *Advances in Neural Information Processing Systems 30*, pp. 7018–7028. Curran Associates, Inc., 2017.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105. Curran Associates, Inc., 2012.

Sanghack Lee and Elias Bareinboim. Causal effect identifiability under partial-observability. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119. PMLR, 2020.

Sanghack Lee and Elias Bareinboim. Causal identification with matrix equations. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9468–9479, 2021.

Sanghack Lee, Juan D Correa, and Elias Bareinboim. General Identifiability with Arbitrary Surrogate Experiments. In *Proceedings of the Thirty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence*, Corvallis, OR, 2019. AUAI Press.

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

Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6449–6459, Red Hook, NY, USA, 2017. Curran Associates Inc.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*. 2013.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

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

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33: 857–869, 2020.

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

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2000.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Judea Pearl and Dana Mackenzie. *The Book of Why*. Basic Books, New York, 2018.

Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

Donald B Rubin. {B}ayesian inference for causal effects: The role of randomization. *Annals of Statistics*, 7(1):34–58, 1978.

Pablo Sanchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Design of variational graph autoencoders for interventional and counterfactual queries. *arXiv preprint arXiv:2110.14690*, 2021. URL `https://arxiv.org/abs/2110.14690`.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3076–3085, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Claudia Shi, David M. Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2503–2513, 2019.

Ilya Shpitser and Judea Pearl. Identification of Joint Interventional Distributions in Recursive semi-Markovian Causal Models. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, volume 2, pp. 1219–1226, 2006.

Ilya Shpitser and Judea Pearl. What Counterfactuals Can Be Tested. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, pp. 352–359. AUAI Press, Vancouver, BC, Canada, 2007.

Ilya Shpitser and Judea Pearl. Effects of Treatment on the Treated: Identification and Generalization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Montreal, Quebec, 2009. AUAI Press.

Ilya Shpitser and Eli Sherman. Identification of Personalized Effects Associated With Causal Pathways. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, pp. 530–539, 2018.

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

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

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Sam Witty, David Jensen, and Vikash Mansinghka. A simulation-based test of identifiability for bayesian causal inference. *arXiv preprint arXiv:2102.11761*, 2021. URL `https://arxiv.org/abs/2102.11761`.

Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The causal-neural connection: Expressiveness, learnability, and inference. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10823–10836. Curran Associates, Inc., 2021.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 2633–2643. Curran Associates, Inc., 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Matej Zecevic, Devendra Singh Dhami, Petar Velickovic, and Kristian Kersting. Relating graph neural networks to structural causal models. *arXiv preprint arXiv:2109.04173*, 2021.

J. Zhang and E. Bareinboim. Fairness in decision-making–the causal explanation formula. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 2037–2045, 2018.

Junzhe Zhang, Tian Jin, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In *Proceedings of the 39th International Conference on Machine Learning (ICML-22)*, 2022.

A PROOFS

A.1 PROOF OF THEOREM 1

First, we bring forth the longer and more formal definition of causal diagrams.

Definition 5 (Causal Diagram (Bareinboim et al., 2022, 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 ($V_j \leftarrow\!\!-\!\!\rightarrow V_i$) 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). ■

Def. 3 states that an SCM is $\mathcal{G}^{(\mathcal{L}_i)}$-consistent if its Layer i distributions satisfy all Layer i equality constraints implied by $\mathcal{G}$. To clarify, we formalize this notion in the following definitions.

Definition 6 (Equality Constraints)**.** Let $\mathbb{P}$ be a collection of distributions. We denote $C(\mathbb{P}) = \{P_1 = P_2 : P_1, P_2 \in \mathbb{P}^*\}$ (where $\mathbb{P}^*$ is the set of all marginal and conditional distributions of $\mathbb{P}$) as the set of all equality constraints that hold between two distributions obtained from $\mathbb{P}$. ■

Definition 7 ($\mathcal{L}_i$ Equality Constraints)**.** For a graph $\mathcal{G}$, let $\Omega^*(\mathcal{G})$ be the set of all SCMs that induce $\mathcal{G}$ as a causal diagram. Let $\mathcal{C}^{(\mathcal{L}_i)} = \bigcap_{\mathcal{M} \in \Omega^*(\mathcal{G})} C(\mathcal{L}_i(\mathcal{M}))$ be the set of all equality constraints that hold for the $\mathcal{L}_i$ distributions of every SCM in $\Omega^*(\mathcal{G})$. Then, for any collection of distributions $\mathbb{P}$, we say that $\mathbb{P}$ satisfies all Layer i equality constraints implied by $\mathcal{G}$ if and only if $\mathcal{C}^{(\mathcal{L}_i)} \subseteq C(\mathbb{P})$. ■

A more detailed discussion on the intuition of these constraints can be found in Appendix D.2. Naturally, it is intuitive that if an SCM induces $\mathcal{G}$ as a causal diagram, it must satisfy all of its constraints, as shown in the following Lemma.

Lemma 1. *Let $\mathcal{M}$ be an SCM and $\mathcal{G}$ be its induced causal diagram. Then $\mathcal{L}_3(\mathcal{M})$ satisfies all Layer 3 equality constraints implied by $\mathcal{G}$.* ■

Proof. If $\Omega^*(\mathcal{G})$ is the set of all SCMs that induce $\mathcal{G}$ as a causal diagram, then $\mathcal{M}$ must be an element of $\Omega^*(\mathcal{G})$. Since $\mathcal{C}^{(\mathcal{L}_3)} = \bigcap_{\mathcal{M}' \in \Omega^*(\mathcal{G})} C(\mathcal{L}_3(\mathcal{M}'))$ is the set of all equality constraints that hold for the $\mathcal{L}_3$ distributions of every SCM in $\Omega^*(\mathcal{G})$, there will not be any constraints in $\mathcal{C}^{(\mathcal{L}_3)}$ that is not in $C(\mathcal{L}_3(\mathcal{M}))$. Hence, $\mathcal{C}^{(\mathcal{L}_3)} \subseteq C(\mathcal{L}_3(\mathcal{M}))$, so $C(\mathcal{L}_3(\mathcal{M}))$ satisfies all Layer i equality constraints implied by $\mathcal{G}$. □

Moreover, the graph induced by any $\mathcal{G}$-constrained NCM is simply $\mathcal{G}$ itself.

Fact 1 ((Xia et al., 2021, Lem. 6))**.** *Let $\widehat{M}(\boldsymbol{\theta}) = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$ be a $\mathcal{G}$-constrained NCM. Let $\widehat{\mathcal{G}}$ be the causal diagram induced by $\widehat{M}$. Then $\widehat{\mathcal{G}} = \mathcal{G}$.*

These two results prove that $\mathcal{G}$-constrained NCMs satisfy all Layer 3 constraints of $\mathcal{G}$.

Theorem 1 (NCM $\mathcal{G}^{(\mathcal{L}_3)}$-Consistency)**.** *Any $\mathcal{G}$-NCM $\widehat{M}(\boldsymbol{\theta})$ is $\mathcal{G}^{(\mathcal{L}_3)}$-consistent.* ■

Proof. This follows directly from Fact 1 and Lem. 1. □

A.2 PROOF OF THEOREM 2

For this proof, we expand on the technical results provided in Zhang et al. (2022). The paper works with a subclass of SCMs known as *discrete SCMs* and proves strong results about its expressiveness. Similar to this paper, we will assume that in any SCM, the variables in $\mathbf{U}$ are independent, and unobserved confounding is modeled by sharing the same variable from $\mathbf{U}$ in the functions of multiple variables in $\mathbf{V}$.

Definition 8 (Discrete SCM (Zhang et al., 2022, Def. 2))**.** An SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U}) \rangle$ is said to be a discrete SCM if $\mathcal{D}_U$ is discrete for all $U \in \mathbf{U}$ and $\mathcal{D}_V$ is both discrete and finite for all $V \in \mathbf{V}$. ■

Fact 2 ((Zhang et al., 2022, Thm. 1))**.** *Let Ω^* be the set of all SCMs and Ω' the set of discrete SCMs. For any SCM $\mathcal{M}^* \in \Omega^*$ inducing causal diagram $\mathcal{G}$, there exists a discrete SCM $\mathcal{M}' \in \Omega'$ with finite $|\mathcal{D}_U|$ for all $U \in \mathbf{U}$ such that $\mathcal{M}'$ is $\mathcal{G}^{(\mathcal{L}_3)}$-consistent and is $\mathcal{L}_3$-consistent with $\mathcal{M}^*$.* ■

With this result, we need only show that the $\mathcal{G}$-NCM can represent any discrete SCM inducing $\mathcal{G}$ with finite domains in $\mathbf{U}$. We will focus on feedforward neural networks, specifically multi-layer perceptrons (MLPs) with the binary step activation function.

Definition 9 (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:

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

This is simply one choice of activation function which always outputs a binary result.

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 note that while Def. 9 is undefined for non-numerical inputs and outputs, any kind of categorical data can be considered if first converted into a numerical representation. We utilize the following results about the expressivity of MLPs to demonstrate the overall expressiveness of an NCM constructed with MLPs.

Fact 3 (MLP Representation (Xia et al., 2021, Lem. 4))**.** *For any function $f : \mathbf{X} \to Y$ mapping set of variables $\mathbf{X}$ to variable Y, all from finite numerical domains, there exists an equivalent MLP $\hat{f}$ using binary step activations.* ■

Fact 4 (Neural Inverse Probability Integral Transform (Discrete) (Xia et al., 2021, Lem. 5))**.** *For any probability mass function $P(\mathbf{X})$ over finite domains, there exists an MLP $\hat{f}$ which maps $\text{Unif}(0, 1)$ to $P(\mathbf{X})$.* ■

We can now combine these results to achieve an expressiveness result on the counterfactual level for NCMs.

Theorem 2 ($\mathcal{L}_3$-$\mathcal{G}$ Expressiveness)**.** *For any SCM $\mathcal{M}^*$ that induces causal diagram $\mathcal{G}$, there exists a $\mathcal{G}$-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 $\mathcal{L}_3$-consistent w.r.t. $\mathcal{M}^*$.* ■

Proof. By Fact 2, there must exist a discrete SCM $\mathcal{M}' = \langle \mathbf{U}', \mathbf{V}, \mathcal{F}', P(\mathbf{U}') \rangle$ such that $\mathcal{M}'$ is $\mathcal{G}$-consistent, is $\mathcal{L}_3$-consistent with $\mathcal{M}^*$, and $|\mathcal{D}_U|$ is finite for all $U \in \mathbf{U}$. Let $\widehat{M} = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$ be a $\mathcal{G}$-NCM such that each $\hat{f}_V \in \widehat{\mathcal{F}}$ is an MLP composed of smaller MLPs as defined next.

For $U \in \mathbf{U}'$, let $\mathbf{V}_U \subseteq \mathbf{V}$ denote the set of endogenous variables such that for all $V \in \mathbf{V}_U$, $\hat{f}_V$ takes U as an argument. Let $\mathbb{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$, with $\mathbf{C}_1, \ldots \mathbf{C}_k \subseteq \mathbf{V}$ denote the maximal confounded

cliques of $\mathcal{G}$, that is, for all $V_1, V_2 \in \mathbf{C}_i$, there is a bidirected edge between V_1 and V_2 in $\mathcal{G}$, and there is no $\mathbf{C}_j$ such that $\mathbf{C}_i \subseteq \mathbf{C}_j$. Let $\mathbf{U}'_{\mathbf{C}_1}, \ldots, \mathbf{U}'_{\mathbf{C}_k}$ be a partition over $\mathbf{U}'$ such that if $U \in \mathbf{U}'_{\mathbf{C}_i}$, then $\mathbf{V}_U \subseteq \mathbf{C}_i$. If for any $U \in \mathbf{U}'$, there exist multiple cliques such that this applies, one can be chosen arbitrarily. For each V, let $\mathbb{C}(V) \subseteq \mathbb{C}$ denote the set of cliques that contain V. Then we note that

$$\mathbf{U}'_V \subseteq \bigcup_{\mathbf{C} \in \mathbb{C}(V)} \mathbf{U}'_{\mathbf{C}},$$

where $\mathbf{U}'_V \subseteq \mathbf{U}'$ denotes the exogenous parents of V in $\mathcal{M}'$.

By construction of the $\mathcal{G}$-NCM in Def. 2, $\widehat{\mathbf{U}}$ consists of a $\mathrm{Unif}(0, 1)$ random variable $\widehat{U}_{\mathbf{C}_i}$ for each maximal confounded clique $\mathbf{C}_i$. By Fact 4, we can construct MLP $\hat{f}^{(\mathbf{U}')}_{\mathbf{C}_i}$ mapping $\widehat{U}_{\mathbf{C}_i}$ to $\mathbf{U}'_{\mathbf{C}_i}$ for each $\mathbf{C}_i$. Then by Fact 3, we can construct MLP $\hat{f}^{(\mathcal{F}')}_V$ to map $\mathbf{U}'_V$ and $\mathbf{Pa}_V$ to V, matching f'_V. Combining these two results, $\hat{f}_V(\mathbf{pa}_V, \widehat{\mathbf{u}}_V)$ is defined as $\hat{f}^{(\mathcal{F}')}_V(\mathbf{pa}_V, \hat{f}^{(\mathbf{U}')}_{\mathbf{C}_{i_1}}(\widehat{u}_{\mathbf{C}_{i_1}}), \ldots, \hat{f}^{(\mathbf{U}')}_{\mathbf{C}_{i_\ell}}(\widehat{u}_{\mathbf{C}_{i_\ell}}))$, where $\mathbb{C}(V) = \{\mathbf{C}_{i_1}, \ldots, \mathbf{C}_{i_\ell}\}$. [9]

Altogether, $\widehat{\mathcal{F}}^{(\mathbf{U}')} = \{\hat{f}^{(\mathbf{U}')}_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}\}$ collectively forms a neural mapping from $\widehat{\mathbf{U}}$ to $\mathbf{U}'$, and $\widehat{\mathcal{F}}^{(\mathcal{F}')} = \{\hat{f}^{(\mathcal{F}')}_V : V \in \mathbf{V}\}$ collectively forms a neural mapping from $\mathbf{U}'$ to $\mathbf{V}$. We use the notation $\mathcal{F}_{\mathbf{x}_i}(\mathbf{U}) \models \mathbf{y}_i$ to be equivalent to the statement $\mathbf{Y}_{i[\mathbf{x}_i]}(\mathbf{U}) = \mathbf{y}_i$, which is the random event (w.r.t. $\mathbf{U}$) that the variables of $\mathbf{Y}_i$ under intervention $\mathbf{X}_i = \mathbf{x}_i$ takes value $\mathbf{y}_i$.

Then, for any counterfactual query $Q = P(\mathbf{y}_*) = P\left(\bigwedge_i \mathbf{Y}_{i[\mathbf{x}_i]} = \mathbf{y}_i\right)$, we have

$$\begin{aligned}
P^{\widehat{M}}(\mathbf{y}_*) &= P\left(\bigwedge_i \widehat{\mathbf{Y}}_{i[\mathbf{x}_i]}(\widehat{\mathbf{U}}) = \mathbf{y}_i\right) \\
&= P\left(\bigwedge_i \widehat{\mathcal{F}}_{\mathbf{x}_i}(\widehat{\mathbf{U}}) \models \mathbf{y}_i\right) \\
&= P\left(\bigwedge_i \widehat{\mathcal{F}}^{(\mathcal{F}')}_{\mathbf{x}_i}\left(\widehat{\mathcal{F}}^{(\mathbf{U}')}(\widehat{\mathbf{U}})\right) \models \mathbf{y}_i\right) \\
&= P\left(\bigwedge_i \widehat{\mathcal{F}}^{(\mathcal{F}')}_{\mathbf{x}_i}(\mathbf{U}') \models \mathbf{y}_i\right) \\
&= P\left(\bigwedge_i \mathcal{F}'_{\mathbf{x}_i}(\mathbf{U}') \models \mathbf{y}_i\right) \\
&= P\left(\bigwedge_i \mathbf{Y}'_{i[\mathbf{x}_i]}(\mathbf{U}') = \mathbf{y}_i\right) \\
&= P^{\mathcal{M}'}(\mathbf{y}_*) \\
&= P^{\mathcal{M}^*}(\mathbf{y}_*).
\end{aligned}$$

Hence, we have shown that for any SCM $\mathcal{M}^*$ inducing causal diagram $\mathcal{G}$, we can construct a $\mathcal{G}$-NCM that matches $\mathcal{M}^*$ on the third layer. □

A.3 Proof of Theorem 3 and Corollaries 1 and 2

Thms. 1 and 2 provide the key properties that result in the proof of Thm. 3.

[9] There are a few subtleties here to align with Def. 9. First, although $\bigcup_{\mathbf{C} \in \mathbb{C}(V)} \mathbf{U}'_{\mathbf{C}}$ may contain elements of $\mathbf{U}'$ not in $\mathbf{U}'_V$, $\hat{f}^{(\mathcal{F}')}_V$ can be constructed to accept them as input and not use them (weight of 0, identity activation function). Secondly, although Def. 9 does not allow additional inputs inbetween layers, $\mathbf{pa}_V$ can simply be provided as an input to $\hat{f}^{(\mathbf{U}')}_{\mathbf{C}}$ and passed forward to the next layer without modification (weight of 1, identity activation function). Thirdly, while Def. 9 is not defined to have multiple nested MLPs in the same layer, the same result can be achieved by nesting them iteratively and passing the relevant outputs through the nested layers without modification. The presentation in the proof is made for the sake of clarity.

Theorem 3 (Counterfactual Graphical-Neural Equivalence (Dual ID))**.** *Let Ω^*, Ω be the spaces including all SCMs and NCMs, respectively. Consider the true SCM $\mathcal{M}^*$ and the corresponding causal diagram $\mathcal{G}$. Let $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$ be the target query and $\mathbb{Z}$ the set of observational and interventional distributions available. Then, Q is neural identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$ if and only if it is identifiable from $\mathcal{G}$ and $\mathbb{Z}$.* ■

Proof. If Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$, it must also be identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$ simply because $\Omega \subset \Omega^*$. If all pairs of SCMs $\mathcal{M}_1, \mathcal{M}_2 \in \Omega^*$ that induce $\mathcal{G}$ and match $\mathcal{M}^*$ in $\mathbb{Z}$ (i.e. $\mathbb{Z}(\mathcal{M}_1) = \mathbb{Z}(\mathcal{M}_2) = \mathbb{Z}(\mathcal{M}^*)$) also match in Q (i.e. $P^{\mathcal{M}_1}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\mathcal{M}_2}(\mathbf{y}_* \mid \mathbf{x}_*)$), this should still hold if $\mathcal{M}_1$ and $\mathcal{M}_2$ are $\mathcal{G}$-NCMs, as by Thm. 1, the set of all $\mathcal{G}^{(\mathcal{L}_3)}$-consistent SCMs includes the set of all $\mathcal{G}$-NCMs.

If Q is not identifiable from $\mathcal{G}$ and $\mathbb{Z}$, there must exist SCMs $\mathcal{M}_1, \mathcal{M}_2 \in \Omega^*$ such that $\mathcal{M}_1$ and $\mathcal{M}_2$ both induce $\mathcal{G}$, $\mathbb{Z}(\mathcal{M}_1) = \mathbb{Z}(\mathcal{M}_2) = \mathbb{Z}(\mathcal{M}^*)$, but $P^{\mathcal{M}_1}(\mathbf{y}_* \mid \mathbf{x}_*) \neq P^{\mathcal{M}_2}(\mathbf{y}_* \mid \mathbf{x}_*)$. By Thm. 2, there must exist corresponding $\mathcal{G}$-NCMs $\widehat{M}_1, \widehat{M}_2 \in \Omega(\mathcal{G})$ such that $\mathcal{L}_3(\widehat{M}_1) = \mathcal{L}_3(\mathcal{M}_1)$ and $\mathcal{L}_3(\widehat{M}_2) = \mathcal{L}_3(\mathcal{M}_2)$. This implies that $\mathbb{Z}(\widehat{M}_1) = \mathbb{Z}(\widehat{M}_2) = \mathbb{Z}(\mathcal{M}^*)$ and that $P^{\widehat{M}_1}(\mathbf{y}_* \mid \mathbf{x}_*) \neq P^{\widehat{M}_2}(\mathbf{y}_* \mid \mathbf{x}_*)$, since $\widehat{M}_1$ and $\widehat{M}_2$ share all layer 3 distributions with $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. However, this provides an example of two NCMs that match in $\mathcal{G}$ and the correct $\mathbb{Z}$ but do not match in Q. This proves that Q is not identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$ if it is not identifiable from $\mathcal{G}$ and $\mathbb{Z}$

Therefore, since both directions hold, it must be the case that Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$ if and only if it is identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$. □

Consequently, any identifiable query can be computed from any proxy NCM $\widehat{M}$ trained to match $\mathcal{M}^*$ in $\mathbb{Z}$ as if it were $\mathcal{M}^*$ itself.

Corollary 1 (Neural Counterfactual Mutilation (Operational ID))**.** *Consider the true SCM $\mathcal{M}^* \in \Omega^*$, causal diagram $\mathcal{G}$, a set of available distributions $\mathbb{Z}$, and a target query Q equal to $P^{\mathcal{M}^*}(\mathbf{y}_* \mid \mathbf{x}_*)$. Let $\widehat{M} \in \Omega(\mathcal{G})$ be a $\mathcal{G}$-constrained NCM such that $\mathbb{Z}(\widehat{M}) = \mathbb{Z}(\mathcal{M}^*)$. If Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$, then Q is computable via Eq. 1 from $\widehat{M}$.* ■

Proof. By Thm. 2, there must exist $\mathcal{G}$-NCM $\widehat{M}' \in \Omega(\mathcal{G})$ such that $\mathcal{L}_3(\widehat{M}') = \mathcal{L}_3(\mathcal{M}^*)$, which implies that $\widehat{M}'$ matches $\mathcal{M}^*$ in both $\mathbb{Z}$ and Q. By Thm. 3, since Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$, it must also be identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$, which implies that all $\mathcal{G}$-NCMs that match in $\mathbb{Z}$ with $\mathcal{M}^*$ must also match $\widehat{M}'$ in Q. In other words, for any $\mathcal{G}$-NCM $\widehat{M} \in \Omega(\mathcal{G})$ such that $\mathbb{Z}(\widehat{M}) = \mathbb{Z}(\mathcal{M}^*)$, it must be the case that $P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\widehat{M}'}(\mathbf{y}_* \mid \mathbf{x}_*)$. Hence, all such NCMs will match $\mathcal{M}^*$ in Q. Finally, $P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) = \frac{P^{\widehat{M}}(\mathbf{y}_*, \mathbf{x}_*)}{P^{\widehat{M}}(\mathbf{x}_*)}$ can be computed from $\widehat{M}$ using Def. 1. □

These two results lead to the soundness and completeness of Alg. 1.

Corollary 2 (Soundness and Completeness)**.** *Let Ω^* be the set of all SCMs, $\mathcal{M}^* \in \Omega^*$ be the true SCM inducing causal diagram $\mathcal{G}$, $Q = P(\mathbf{y}_* \mid \mathbf{x}_*)$ be a query of interest, and $\widehat{Q}$ be the result from running Alg. 1 with inputs $\mathbb{Z}(\mathcal{M}^*) > 0$, $\mathcal{G}$, 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 $\widehat{Q} = P^{\mathcal{M}^*}(\mathbf{y}_* \mid \mathbf{x}_*)$.* ■

Proof. Thm. 2 states that Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$ if and only if Q is identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$. By definition, Q is identifiable from $\Omega(\mathcal{G})$ and $\mathbb{Z}$ if and only if for all pairs of $\mathcal{G}$-NCMs, $\widehat{M}_1, \widehat{M}_2 \in \Omega(\mathcal{G})$ with $\mathbb{Z}(\widehat{M}_1) = \mathbb{Z}(\widehat{M}_2) = \mathbb{Z}(\mathcal{M}^*)$, we have $P^{\widehat{M}_1}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\widehat{M}_2}(\mathbf{y}_* \mid \mathbf{x}_*)$. This holds if and only if $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\widehat{M}(\boldsymbol{\theta}^*_{\max})}(\mathbf{y}_* \mid \mathbf{x}_*)$. If they are not equal, then $\widehat{M}(\boldsymbol{\theta}^*_{\min})$ and $\widehat{M}(\boldsymbol{\theta}^*_{\max})$ are examples of two NCMs that match in $\mathcal{G}$ and the correct $\mathbb{Z}$ but not in Q. Otherwise, all $\mathcal{G}$-NCMs that match in $\mathbb{Z}$ must also match in Q, since any NCM that induces a result for Q that is less than $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y}_* \mid \mathbf{x}_*)$ or greater than $P^{\widehat{M}(\boldsymbol{\theta}^*_{\min})}(\mathbf{y}_* \mid \mathbf{x}_*)$ would contradict the optimality of $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$.

If Q is identifiable, then Corol. 1 guarantees that any $\mathcal{G}$-NCM $\widehat{M}$ that induces the correct $\mathbb{Z}$ will match $\mathcal{M}^*$ in Q (i.e. $P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) = P^{\mathcal{M}^*}(\mathbf{y}_* \mid \mathbf{x}_*)$). □

B EXPERIMENTAL DETAILS

This section provides the detailed information about our experiments. Our models are primarily written in PyTorch (Paszke et al., 2017), and training is facilitated using PyTorch Lightning (Falcon & Cho, 2020).

B.1 ADDITIONAL RESULTS

The accuracies from Fig. 4 are computed from the max-min gaps (i.e. the difference between the induced query of the maximizing model and the minimizing model) via a hypothesis testing procedure (see Sec. 4). The actual values of these gaps are plotted below.

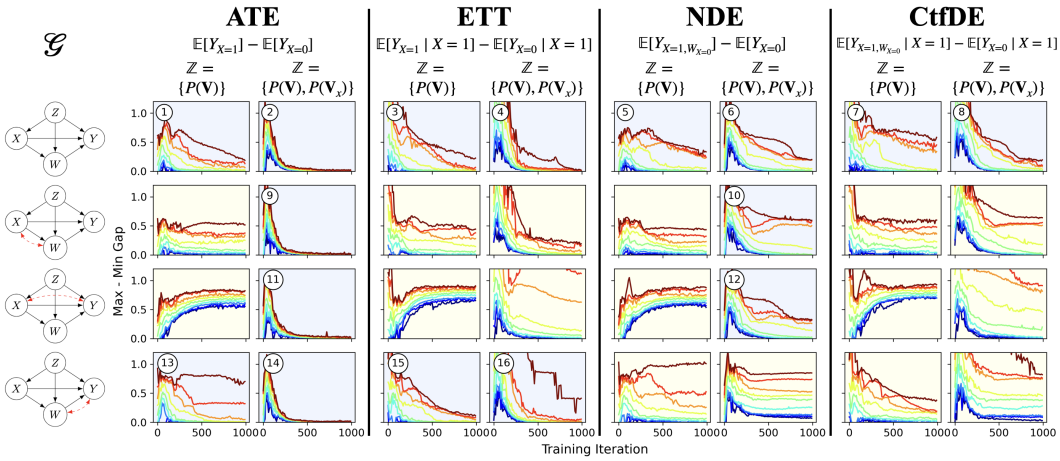


Figure 7: (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentiles of max-min gaps of 1-dimensional (1D) GAN-NCM in ID experiments shown in Fig. 4.

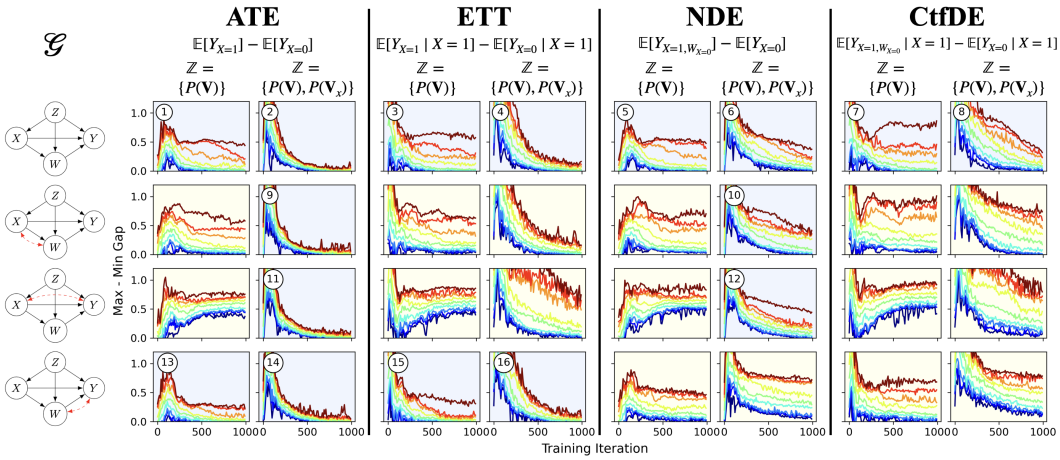


Figure 8: (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentiles of max-min gaps of 16-dimensional (16D) GAN-NCM in ID experiments shown in Fig. 4.

Note from the GAN-NCM results in Fig. 7 that in both ID and non-ID cases, the gaps are initially large due to the large value of λ. As training progresses and λ decreases, the gaps tighten in ID cases but remain wide in non-ID cases. In terms of the accuracy curves in Fig. 4, the result is that accuracy starts low but slowly increases in ID cases, while accuracy remains high the entire time in non-ID cases. Fig. 8 shows similar patterns, but gaps tend to be larger in general due to the higher complexity of having higher dimensions. Positivity of data distributions may be more difficult to achieve with a smaller dataset, so the GAN-NCM leans on the safer side and predicts "non-ID" more often.

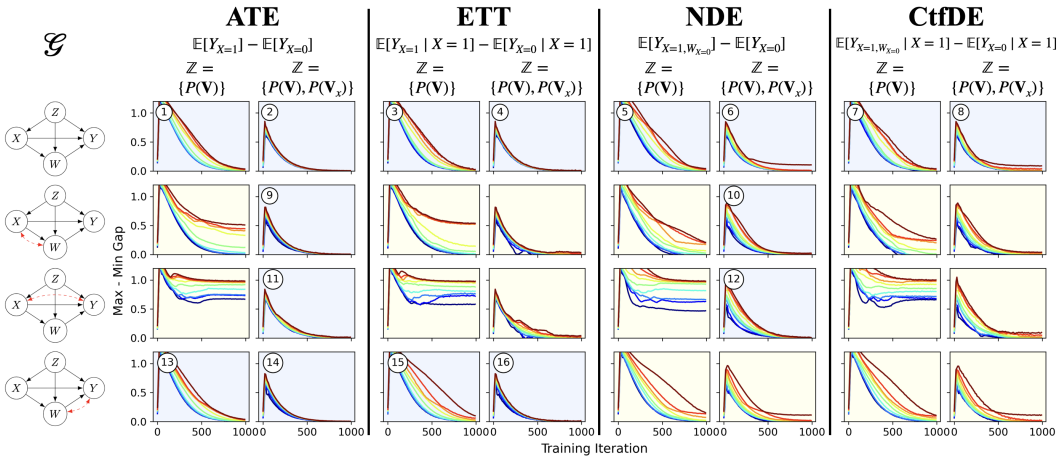


Figure 9: (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentiles of max-min gaps of 1-dimensional (1D) MLE-NCM in ID experiments shown in Fig. 4.

On the other hand, when looking at the MLE-NCM gaps from Fig. 9, that gaps often tighten during training even in non-ID cases, suggesting that MLE-NCM may have trouble finding the proper bounds in non-trivial non-ID cases.

For the estimation experiments shown in Fig. 6, we plot the total KL divergence comparing the induced distributions of the trained models to the true data distributions in Fig. 10. Note the downward trend in all three cases, suggesting that GAN-NCM and MLE-NCM improve in modeling the true distribution with more data (a good sanity check). MLE-NCM is more consistent and achieves slightly lower KL for the same amount of data, which also reflects in its accuracy of the induced query. GAN-NCM is evidently not as statistically efficient due to its implicit design and less stable training method, but it is capable of learning the data distribution even in high dimensions.

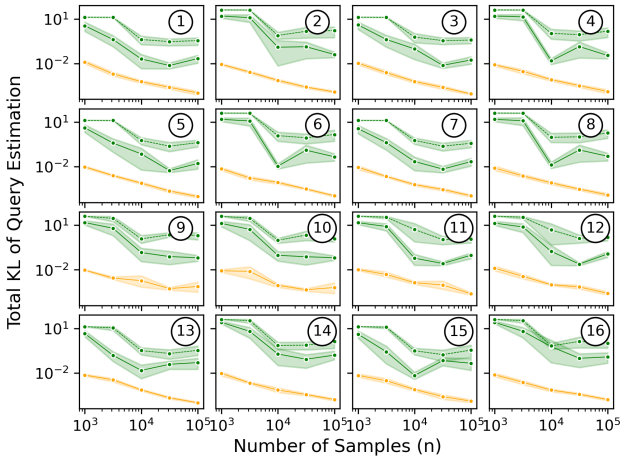


Figure 10: Total KL Divergence of estimation experiments shown in Fig. 6. Results are shown for GAN-NCM (solid green) and MLE-NCM (orange) with $d = 1$ and also GAN-NCM with $d = 16$ (dashed green).

To demonstrate that this work subsumes results from layer 2, we demonstrate the ability of the GAN-NCM to solve the ID and estimation problems in the same 8 settings from Fig. 4 of (Xia et al., 2021). Results are shown in Fig. 11, compared alongside the MLE-NCM which is the original approach used. Recall that the query tested is the average treatment effect (ATE) equal to $P(Y_{X=1} = 1) - P(Y_{X=0} = 1)$. Results are comparable to results from Fig. 4. Like the MLE-NCM, the GAN-NCM starts with large gaps, which slowly tighten in ID cases but stay wide in non-ID cases. Accuracy is also similar.

Fig. 12 shows the estimation results from estimating the four identifiable cases from Fig. 11. Estimation results are similar to that of the counterfactual cases shown in Fig. 6. As expected, KL divergence decreases with data for all three settings, but training is slightly more efficient with the MLE-NCM. This is reflected in the slightly reduced error on the query under the same amount of data. Still, GAN-NCM is a viable method for estimating ATE in these cases, even in high dimensions.

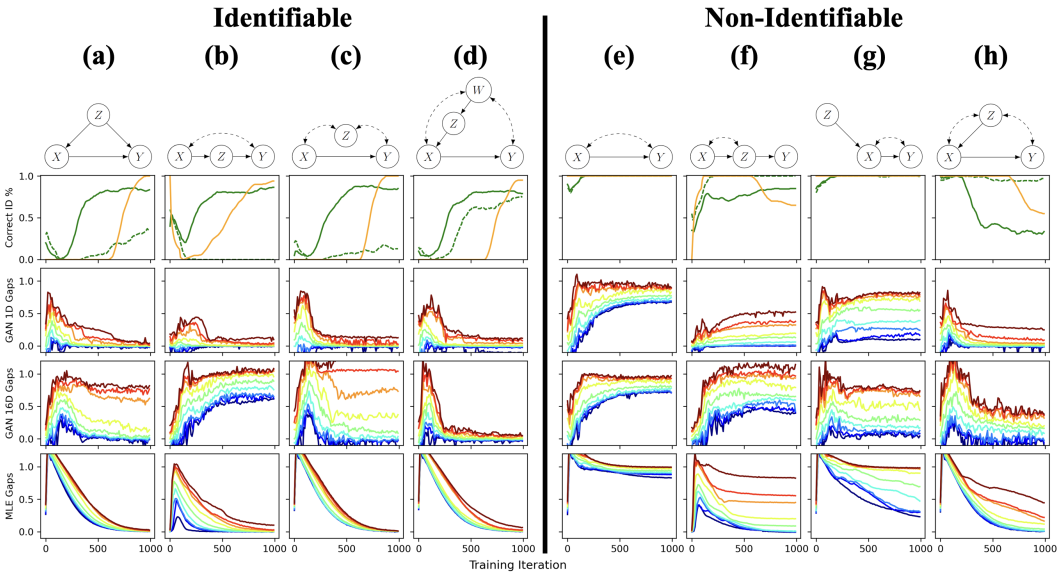


Figure 11: Identification results of the ATE query in the 8 settings from (Xia et al., 2021). The graphs at the top from left to right are: (ID cases) back-door, front-door, M, napkin; (not ID cases) bow, extended bow, IV, bad M. In the top row of plots, the running average of classification accuracy over 1000 epochs of both GAN-NCM (green) and MLE-NCM (orange) are shown. The second, third, and fourth rows of plots show the (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentiles of max-min gaps of 1D GAN-NCM, 16D GAN-NCM, and MLE-NCM respectively.

As mentioned earlier, the MLE-NCM may be problematic for solving the ID problem due to its apparent bias for returning ID conclusions even in non-ID cases. This is obviously undesirable, especially since attempting to estimate a quantity in a non-ID case will likely produce a misleading result. This behavior is not evident in Fig. 11, so one may be tempted to conclude that this is only an issue for complicated counterfactual quantities and not simple layer 2 quantities like ATE. However, this is not the case.

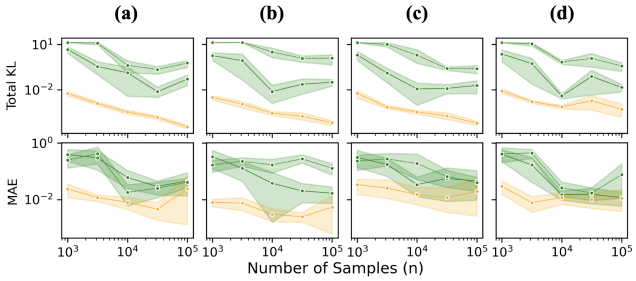


Figure 12: Estimation results for identifiable cases shown in Fig. 11. Results are shown for GAN-NCM (solid green) and MLE-NCM (orange) with $d = 1$ and also GAN-NCM with $d = 16$ (dashed green). Top row is KL divergence between the true $P(\mathbf{V})$ and the one induced by the trained NCM. Bottom row is mean absolute error (MAE) of the induced query, ATE.

To witness, consider the back-door + M graph in Fig. 13, created by combining the edges of the back-door and M graphs (graphs a and c in Fig. 11). It is also the bad M graph (graph h in Fig. 11) with an additional edge from Z to Y. In this setting, $P(y_x)$ (and therefore ATE) is not identifiable from $P(\mathbf{V})$ and $\mathcal{G}$. We test the GAN-NCM and MLE-NCM to identify this setting in Fig. 14.

Evidently, the same bias of the MLE-NCM of incorrectly returning an identifiable result occurs here. As training progresses, the gaps begin to close (shown in Fig. 14d) despite the query being non-identifiable. This results in a very poor accuracy towards the end of training (shown through orange line in Fig. 14a). The GAN-NCM handles this robustly in both 1 and 16 dimensions, having large gaps all the way throughout training (Figs. 14b and 14c) and maintaining a high accuracy (green lines in Fig. 14a).

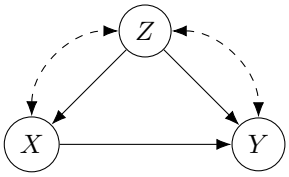
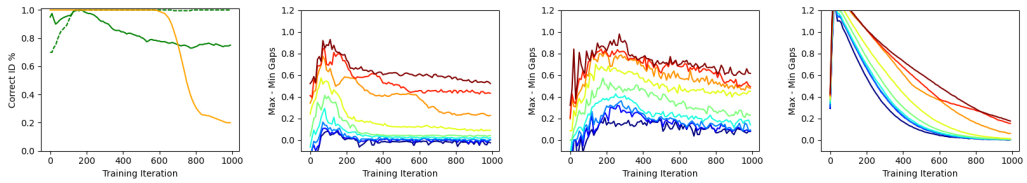


Figure 13: Back-door + M graph.



(a) ID accuracy. (b) GAN-NCM 1D gaps. (c) GAN-NCM 16D gaps. (d) MLE-NCM 1D gaps.

Figure 14: Identification results for the ATE query from $P(\mathbf{V})$ and the graph $\mathcal{G}$ in Fig. 13. The four plots shown from left to right are ID accuracy and (1, 5, 10, 25, 50, 75, 90, 95, 99)-percentile max-min gaps for 1D GAN-NCM (green), 16D GAN-NCM (dashed green), and 1D MLE-NCM (orange) over 1000 training epochs.

We do not make any theoretical claim that explains this phenomenon.
From experimentation, it seems that MLE-NCM is more easily able to model associations between variables through directed edges instead of bidirected edges. That is, if $X \to Y$ and $X \leftrightarrow Y$, the MLE-NCM may be biased to modeling the association between X and Y by directly using the value of x in $\hat{f}_Y$ as opposed to training $\hat{f}_X$ and $\hat{f}_Y$ to use the shared U_{XY} to model this association. As a result, the MLE-NCM may be treating Fig. 13 as a simple back-door case, and the optimization term to maximize/minimize the query may not be strong enough to override this bias.

B.2 Nested Counterfactuals

A more general definition of counterfactuals can be defined as shown below (see (Correa et al., 2021) for a more detailed discussion).

Definition 10 (Layer 3 Valuation (General))**.** An SCM $\mathcal{M}$ induces layer $\mathcal{L}_3(\mathcal{M})$, a set of distributions over $\mathbf{V}$, each with the form $P(\mathbf{Y}_*) = P(\mathbf{Y}_{1[\widetilde{\mathbf{X}}_1]}, \mathbf{Y}_{2[\widetilde{\mathbf{X}}_2],\dots})$ such that

$$P^{\mathcal{M}}(\mathbf{y}_{1[\widetilde{\mathbf{X}}_1]}, \mathbf{y}_{2[\widetilde{\mathbf{X}}_2]}, \dots) = \int_{\mathcal{D}_{\mathbf{U}}} \mathbb{1}\left[\mathbf{Y}_{1[\widetilde{\mathbf{X}}_1]}(\mathbf{u}) = \mathbf{y}_1, \mathbf{Y}_{2[\widetilde{\mathbf{X}}_2]}(\mathbf{u}) = \mathbf{y}_2, \dots\right] dP(\mathbf{u}), \tag{6}$$

where $\widetilde{\mathbf{X}}_i$ is a collection of functions defined over variables $\mathbf{X}_i$ (such that we have $\tilde{f}_X$ mapping from some $\widetilde{\mathbf{U}}_X \subseteq \mathbf{U}$ to $\mathcal{D}_X$ for each $\tilde{f}_X \in \widetilde{\mathbf{X}}_i$), and $\mathbf{Y}_{i[\widetilde{\mathbf{X}}_i]}(\mathbf{u})$ (also called *potential response* (Pearl, 2000, Def. 7.4.1)) is the solution for $\mathbf{Y}$ after evaluating $\mathcal{F}_{\widetilde{\mathbf{X}}_i} := \{f_{V_j} : V_j \in \mathbf{V} \setminus \mathbf{X}\} \cup \{f_X \leftarrow \tilde{f}_X : \tilde{f}_X \in \widetilde{\mathbf{X}}_i\}$ with $\mathbf{u}$.

■

$\mathbf{Y}_{\widetilde{\mathbf{X}}}$ is considered a natural intervention if each $\tilde{f}_X \in \widetilde{\mathbf{X}}$ is a potential response for X in $\mathcal{M}$. If $\tilde{f}_X$ for variable X is set to some fixed value x, then that corresponds to an atomic intervention, which is what is considered in Def. 10. However, one could also have a natural intervention where $\tilde{f}_X := X_{\widetilde{\mathbf{Z}}}(\mathbf{u})$ for some $\widetilde{\mathbf{Z}}$, defined recursively. In other words, the counterfactual terms may be *nested*.

For example, consider the term $P(Y_{X=1,W_{X=0}} = 1)$ in the natural direct effect (NDE) query. This would be evaluated as

$$\begin{aligned} P(Y_{X=1,W_{X=0}} = 1) &= \int_{\mathcal{D}_{\mathbf{U}}} \mathbb{1}\left[Y_{X=1,W_{X=0}}(\mathbf{u}) = 1\right] dP(\mathbf{u}) \\ &= \int_{\mathcal{D}_{\mathbf{U}}} \mathbb{1}\left[Y_{X=1,W=W_{X=0}(\mathbf{u})}(\mathbf{u}) = 1\right] dP(\mathbf{u}). \end{aligned}$$

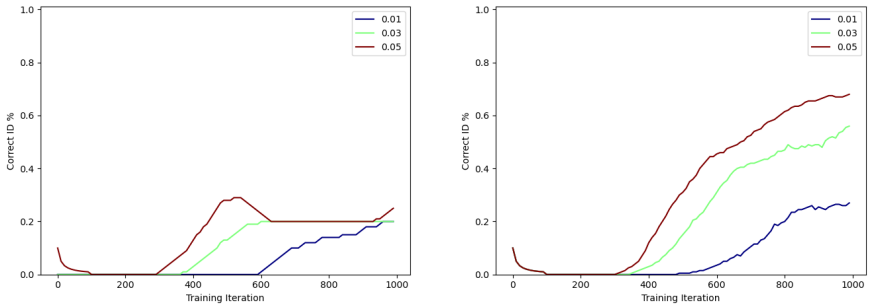
Intuitively, we are aggregating all settings of $\mathbf{u}$ in which $Y = 1$ under the intervention where X is fixed to 1 and W is fixed to the value that it would take under the same setting of $\mathbf{u}$ under the intervention where X is fixed to 0.

It turns out that all nested counterfactuals can be unrolled into an unnested form via the Counterfactual Unnesting Theorem (CUT).

Fact 5 (Counterfactual Unnesting Theorem, (Correa et al., 2021, Thm. 1))**.** *Let* $\widetilde{\mathbf{X}}$*,* $\widetilde{\mathbf{Z}}$ *be any natural interventions on disjoint sets* $\mathbf{X}, \mathbf{Z} \subseteq \mathbf{V}$*. Then for* $\mathbf{Y} \subseteq \mathbf{V}$ *disjoint from* $\mathbf{X}$ *and* $\mathbf{Z}$*, we have*

$$P(\mathbf{Y}_{\widetilde{\mathbf{Z}},\widetilde{\mathbf{X}}} = \mathbf{y}) = \sum_{\mathbf{x} \in \mathcal{D}_{\mathbf{X}}} P(\mathbf{Y}_{\widetilde{\mathbf{Z}},\mathbf{x}} = \mathbf{y}, \widetilde{\mathbf{X}} = \mathbf{x}) \tag{7}$$

This theorem simplifies the scope of theoretical studies on counterfactual identification, and indeed, one could perform inferences on such queries using the NCM by performing it on the terms of the unnested version. However, the proxy model nature of the NCM allows it to directly evaluate nested counterfactuals according to the definition. In the case of $P(Y_{X=1,W_{X=0}} = 1)$, for example, this can be estimated by sampling several instances of $\widehat{\mathbf{U}}$, evaluating W in those instances under the intervention $X = 0$, and then subsequently evaluating Y under the intervention $X = 1$ and with W set to the corresponding values obtained previously. This dramatically simplifies the evaluation of nested counterfactuals, which otherwise would have required an analytical understanding to unnest the query manually, followed by requiring an evaluation on all terms in the sum in the r.h.s. of Eq. 7. In practice, this sum takes a long time (possibly intractable in continuous cases) to compute, with compounding error in each term. A visualization of the improvement is shown in Fig. 15.



(a) Unnested Evaluation

(b) Natural Evaluation

Figure 15: A comparison of identifying NDE from $\mathbb{Z} = \{P(\mathbf{V}), P(\mathbf{V}_x)\}$ and $\mathcal{G}$, with $\mathcal{G}$ being the second graph of Fig. 4. Three different choices of τ are shown. Left figure shows results from evaluating the unnested form of NDE, while right figure shows results from performing the natural evaluation of NDE by definition.

B.3 Data Generation

To generate data for our experiments, we use an implementation of what we call a *regional canonical model*, inspired by canonical SCMs (Zhang et al., 2022).

Definition 11 (Regional Canonical Model (RCM))**.** Given a causal diagram $\mathcal{G}$, a *regional canonical model* $\mathcal{M}_{\text{RCM}}$ associated with $\mathcal{G}$ is an SCM $\langle \mathbf{V}, \mathbf{U}_{\text{RCM}}, \mathcal{F}_{\text{RCM}}, P(\mathbf{U}_{\text{RCM}}) \rangle$ where:

1. $\mathbf{U}_{\text{RCM}} = \{R_{\mathbf{C}} : \forall \mathbf{C} \in \mathbb{C}(\mathcal{G})\}$, where $\mathbb{C}(\mathcal{G})$ is the set of maximally confounded cliques of $\mathcal{G}$ and $R_{\mathbf{C}} \sim \text{Unif}(0, 1)$ for all $R_{\mathbf{C}} \in \mathbf{U}_{\text{RCM}}$.

2. For each $V \in \mathbf{V}$, denote $\mathbf{R}_V = \{R_{\mathbf{C}} : \forall \mathbf{C} \text{ s.t. } V \in \mathbf{C}\}$. Then, each V is associated with a set of regions $\mathbf{I}_V = \{\mathbf{I}_{V,i}\}_{i=1}^{r}$ such that each $\mathbf{I}_{V,i} = \{[a, b]_{\mathbf{C},i} : R_{\mathbf{C}} \in \mathbf{R}_V\}$ with $a, b \in [0, 1]$, $a < b$. In words, each $\mathbf{I}_{V,i}$ is a collection of intervals within 0 to 1 for each of the exogenous parents in $\mathbf{R}_V$. Furthermore, each $\mathbf{I}_{V,i}$ is associated with a corresponding function $h_{V,i} : \mathcal{D}_{\mathbf{Pa}_V} \to \mathcal{D}_V$ mapping to values of V from its parents (in $\mathcal{G}$). Here, r is a hyperparameter which decides the complexity of the model.

3. Each $f_V^{\text{RCM}} \in \mathcal{F}_{\text{RCM}}$ is defined as

$$f_V^{\text{RCM}}(\mathbf{pa}_V, \mathbf{r}_V) = \begin{cases} h_{V,i^*}(\mathbf{pa}_V) & i^* \text{ is the largest } i \text{ s.t. } r_{\mathbf{C}} \in [a, b]_{\mathbf{C},i} \quad \forall r_{\mathbf{C}} \in \mathbf{r}_V \\ h_{V,\text{DEFAULT}}(\mathbf{pa}_V) & \text{if no such } i^* \text{ exists.} \end{cases}$$

Explaining this model in words, a uniformly random exogenous variable is assigned to each confounded clique ($R_{\mathbf{C}}$), and all variables within that clique will take that variable as an exogenous input to its functions. Each variable V is then associated with a set of exogenous variables $\mathbf{R}_V$ depending on the cliques of which it is a member. The space of the domains of $\mathbf{R}_V$ form a unit hypercube with dimension $|\mathbf{R}_V|$, and each V is assigned r rectangular "regions" in this hypercube ($\mathbf{I}_V$). Each region is associated with a function for V ($h_{V,i}$) given its parents, which is the behavior that describes f_V^{RCM} when the inputted $\mathbf{r}_V$ falls within that region. If $\mathbf{r}_V$ does not fall into any of the r regions, then f_V^{RCM} will take a default behavior described by $h_{V,\text{DEFAULT}}$ instead.

For our experimental settings, we randomly sample an RCM given the graph $\mathcal{G}$ to be the ground truth, $\mathcal{M}^*$, with $r = 20$. Each interval $[a, b]_{\mathbf{C},i}$ is sampled by sampling two numbers from $\text{Unif}(0, 1)$ and sorting them. Since all variables are binary vectors, $h_{V,\text{DEFAULT}}$ and each $h_{V,i}$ map each possible input of $\mathbf{pa}_V$ to a randomly sampled binary vector (each bit distributed $\mathsf{Bernoulli}(0.5)$) corresponding to a value of V. All variables in the simulations are single-dimensional binary variables except for Z, which is a d-dimensional binary vector depending on the setting of d in the experiment. To simulate the ground truth, samples can be drawn from the canonical SCM using the same procedure from Def. 1.

Similar to the canonical SCM in Zhang et al. (2022), the RCM assigns different functional behavior to each variable given its endogenous parents depending on the value of their exogenous parents. As implicated in the proof of Thm. 2.3 in Zhang et al. (2022), such a model is very expressive with sufficiently large variation in the space of exogenous parents, eventually being able to express any SCM on all three layers. In our case, this can be achieved with a large enough choice of r. This provides a richer experimental environment when compared to choosing a fixed ground truth $\mathcal{M}^*$ or sampling $\mathcal{M}^*$ from a more restricted parametric family of models (which may produce biased results depending on the family).

B.4 Hypothesis Testing

Alg. 1 is a search within the $\Omega(\mathcal{G})$ space for two NCM parameterizations, $\boldsymbol{\theta}^*_{\min}$ and $\boldsymbol{\theta}^*_{\max}$, that minimizes/maximizes the desired query, respectively. After optimization is finished, the corresponding $P(\mathbf{y}_* \mid \mathbf{x}_*)$ of the two models can be compared to determine whether an effect is identifiable. With perfect optimization and unbounded resources, these two quantities will be equal in identifiable cases. In practice, the two quantities are unlikely to be exactly equal due to noise in the training procedure, so we rely on a hypothesis testing step such as

$$|f(\widehat{M}(\boldsymbol{\theta}_{\max})) - f(\widehat{M}(\boldsymbol{\theta}_{\min}))| < \tau \tag{8}$$

for quantity of interest f and a certain threshold τ. This threshold is similar to a significance level in statistics and can be adjusted depending on the severity of implications of false positives or negatives. In general, τ can be determined by analyzing empirical results in which the identification ground truth is known, and we choose $\tau = 0.05$ in our experiments for all cases.

B.5 GAN-NCM Experiments

In this section, we describe the architecture of the implicit GAN-NCM as described in Sec. 4.

B.5.1 Model Architecture

The implementation of the GAN-NCM used in the experiments specifically utilizes Wasserstein GANs with gradient penalty (Gulrajani et al., 2017). This choice was made to avoid the problem of mode collapse, a frequent problem that arises when training GANs. Wasserstein GANs (WGANs) (Arjovsky et al., 2017) minimize Earth-Mover distance via the Kantorovich-Rubinstein duality (Villani, 2009), producing the objective:

$$\min_{G} \max_{D \in \mathcal{D}_D} \mathbb{E}_{\mathbf{x} \sim P_r}[D(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim P_g}[D(\tilde{\mathbf{x}})], \tag{9}$$

where G is a generating model, D is a discriminatory model (called a critic in this instance), $\mathcal{D}_D$ is the set of 1-Lipshitz functions, P_r is a real distribution (i.e. from $\mathcal{M}^*$), and P_g is the distribution induced by G.

Wasserstein GAN with gradient penalty (WGAN-GP) removes the hard 1-Lipschitz constraint over $\mathcal{D}_D$ and enforces this constraint softly by using an additional penalty term on the gradient of the critic in the objective, namely

$$L = \mathbb{E}_{\tilde{\mathbf{x}} \sim P_g}[D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim P_r}[D(\mathbf{x})] + \lambda_{\text{GP}} \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\text{GP}}} \left[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2 \right], \tag{10}$$

where data points from P_{GP} is sampled uniformly along straight lines between pairs of points sampled from P_g and P_r, and λ_{GP} is a hyperparameter regulating the strength of this penalty. This objective is minimized in the optimization of the critic model. In our case, we use a generalized version of this objective:

$$L = \mathbb{E}_{\tilde{\mathbf{x}} \sim P_g}[D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim P_r}[D(\mathbf{x})] + \lambda_{\text{GP}} \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\text{GP}}} \left[(\|\max(0, \nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}))\|_2 - c)^2 \right], \tag{11}$$

where the desired Lipschitz constant c can be adjusted, and only gradients larger than c are penalized. For our experiments, we use $c = 0.01$ and $\lambda_{\text{GP}} = 10$, which produced the best results while hyperparameter tuning.

In a GAN-NCM $\widehat{M} = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$, the noise $P(\widehat{\mathbf{U}})$ and functions $\mathcal{F}$ collectively form a generative model over $\mathbf{V}$. With the mutilation procedure, the same noise and functions form a generative model over $\mathbf{V}_{\mathbf{x}}$, where the functions for variables in $\mathbf{X}$ are set to fixed values. Hence, we treat the parameters of $\widehat{\mathcal{F}}$ as the parameters of the generator G in the GAN objective. Another fully-connected multi-layer perceptron (MLP) is used as the discriminator/critic, D.

When fitting a specific data distribution $P(\mathbf{V}_{\mathbf{x}})$, the generator ($\widehat{\mathcal{F}}_{\mathbf{x}}$) attempts to maximize $\mathbb{E}_{\tilde{\mathbf{x}} \sim P^{\widehat{M}}(\mathbf{V}_{\mathbf{x}})}[D(\tilde{\mathbf{x}})]$, the expected score assigned by the critic for generated samples. The critic (D) attempts to minimize the objective in Eq. 11.

In cases where multiple datasets are fitted ($|\mathbb{Z}| > 1$), one could create a critic network for each dataset, but for the sake of computational efficiency, we use the same critic network D for all datasets, with an additional input of an index clarifying which dataset is being evaluated. We did not find this to have any significant impact to performance.

In our experiments, each function $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ is a fully connected multi-layer-perceptron (MLP) with 2 hidden layers of size 64, layer normalization (Ba et al., 2016) applied after each layer, and ReLU activations (Nair & Hinton, 2010) after each hidden layer. These choices were made as best practices for implementing WGAN-GP. More layers of larger size can be used for more complicated data. Since we work with binary data, a sigmoid activation function is placed at the end to constrain values between 0 and 1. Values are rounded at evaluation time. To simplify training, we increase the dimensionality of each $\widehat{U} \in \widehat{\mathbf{U}}$ to 2. We recommend using larger dimensions for cases where bidirected cliques are very large, and a single $\widehat{U}$ variable is inputted to several functions.

For the critic D, we use an MLP with 2 hidden layers of size $64 \times |\mathbf{V}|$, which scales depending on the number of variables in the dataset. We also apply layer normalization after each hidden layer and use ReLU activations. We do not use an activation after the last layer.

Weights of all networks are initialized using Glorot initialization (Glorot & Bengio, 2010). The models are trained using the Adam optimizer (Kingma & Ba, 2015).

B.5.2 Query Loss

Since variables in our experiments are binary vectors, we compute $\mathbb{D}_Q$ as in Eq. 5 by using log loss. For example, if $Q = P(Y = 1)$, let $\widehat{Q} = \{\hat{y}_j\}_{j=1}^m$ be m samples of Y from the NCM $\widehat{M}$. Note that all outputs from $\widehat{M}$ are real numbers between 0 and 1 as a result of the sigmoid activation. If the goal was to maximize the query, we would compute

$$\mathbb{D}_Q(\widehat{Q}, Q) = \mathbb{D}_Q(\{\hat{y}_j\}_{j=1}^m, 1) = \sum_{j=1}^{m} -\log(\hat{y}_j)$$

which would approach 0 as $\hat{y}_j \to 1$ and approach $+\infty$ as $\hat{y}_j \to 0$. Penalizing this term would encourage the optimization process to generate samples of $\hat{y}_j$ closer to 1. In the case of minimization, we would replace $-\log(\hat{y}_j)$ with $-\log(1 - \hat{y}_j)$.

More generally, denote $\hat{y}_{X=x|Z=z}$ as a sample from $Y^{\widehat{M}}_{X=x}(\widehat{\mathbf{U}})$ under the condition that $Z^{\widehat{M}}(\widehat{\mathbf{U}}) = z$. In other words, $\hat{y}_{X=x|Z=z}$ is a sample from the NCM under training, $\widehat{M}$, under the intervention $X = x$ such that under the same sample of $\widehat{\mathbf{U}}$, Z evaluates to z (obtained via rejection sampling). Then, for each Monte Carlo sample of the query, we add the corresponding loss term in Table 1 depending on the query and whether we are minimizing or maximizing.

Query	Calculation	Min Loss	Max Loss
ATE	$P(Y_{X=1} = 1)$ $-P(Y_{X=0} = 1)$	$-\log(1 - \hat{y}_{X=1})$ $-\log(\hat{y}_{X=0})$	$-\log(\hat{y}_{X=1})$ $-\log(1 - \hat{y}_{X=0})$
ETT	$P(Y_{X=1} = 1 \mid X = 1)$ $-P(Y_{X=0} = 1 \mid X = 1)$	$-\log(1 - \hat{y}_{X=1\|X=1})$ $-\log(\hat{y}_{X=0\|X=1})$	$-\log(\hat{y}_{X=1\|X=1})$ $-\log(1 - \hat{y}_{X=0\|X=1})$
NDE	$P(Y_{X=1,W_{X=0}} = 1)$ $-P(Y_{X=0} = 1)$	$-\log(1 - \hat{y}_{X=1,W_{X=0}})$ $-\log(\hat{y}_{X=0})$	$-\log(\hat{y}_{X=1,W_{X=0}})$ $-\log(1 - \hat{y}_{X=0})$
CtfDE	$P(Y_{X=1,W_{X=0}=1} \mid X = 1)$ $-P(Y_{X=0} = 1 \mid X = 1)$	$-\log(1 - \hat{y}_{X=1,W_{X=0}\|X=1})$ $-\log(\hat{y}_{X=0\|X=1})$	$-\log(\hat{y}_{X=1,W_{X=0}\|X=1})$ $-\log(1 - \hat{y}_{X=0\|X=1})$

Table 1: Log loss terms for $\mathbb{D}_Q$ in Eq. 5 for each of the queries in Fig. 4. Note that if Y is a binary variable, then $\mathbb{E}[Y] = P(Y = 1)$.

B.5.3 Identification Experiment Procedure

We ran 20 trials for each setting, each with 4 reruns for the sake of hypothesis testing (see Sec. B.4). One trial of the experiment begins with by sampling $n = 10^4$ data points from a randomly generated RCM (see Sec. B.3). In the 1D case, we enforce positivity to accurately reflect our assumptions. In the 16D case, positivity is unreasonable for low sample sizes, so this is not enforced. This may be reflected in a lower accuracy in ID cases, since some identifiable cases are no longer identifiable when positivity does not hold.

For each rerun, one model is initialized and trained to fit the data while simultaneously maximizing the query, then a second model is initialized and trained to fit the data with simultaneously minimizing the query. Results are computed afterwards from the two models.

Each training epoch follows the procedure in Alg. 3. For each dataset in $\mathbb{Z}$, a batch is sampled from the real dataset alongside a batch from the corresponding distribution from the NCM, $\widehat{M}$. These batches are used to train the critic, D, as discussed in Sec. B.5.1. Following this, another batch is sampled from $\widehat{M}$ and evaluated by the critic. The resulting loss is summed across all datasets and added to the query loss explained in Sec. B.5.2, multiplied by λ. This loss is used to train the generator, or in other words, the functions of $\widehat{\mathcal{F}}$.

We use a learning rate of $\eta = 2 \cdot 10^{-5}$ and a batch size of 256 chosen from hyperparameter tuning results, but performance is relatively stable with other choices of learning rate and batch size when using the Adam optimizer. We run for 1000 epochs, but more epochs can be used for better results. We start λ at 10^{-4} and exponentially decrease to 10^{-5} by the end of training. A wider range can be used for λ given more epochs, but using values that are too extreme may disrupt the GAN training process. We found that with more datasets in $\mathbb{Z}$, the total sum of losses from $\mathbb{D}_P$ overpowers the query loss from $\mathbb{D}_Q$ compared to when $\mathbb{Z}$ has fewer datasets. We account for this by multiplying λ by a factor of $|\mathbb{Z}|^2$. We use $m = 10^4$ Monte Carlo samples for estimating Eq. 3 during training, but more can be used for a more accurate result at the expense of computational time.

B.5.4 Estimation Experiment Procedure

For estimation experiments, the procedure is largely the same as the identification experiments except only one model is trained, the query loss $\mathbb{D}_Q$ is not computed, and optimization is purely to fit the datasets in $\mathbb{Z}$. For each setting in Fig. 6, we ran 5 different settings of n, increasing exponentially from 10^3 to 10^5. We ran all settings for $d = 1$ and $d = 16$. We ran 10 trials for each setting, with means and confidence intervals computed across these 10 trials.

We use a learning rate of $\eta = 2 \cdot 10^{-5}$, a batch size of 1000, and run for up to 3000 epochs with early stopping. We use $m = 10^6$ Monte Carlo samples to evaluate the query from $\widehat{M}$ via Eq. 3 after training.

B.6 MLE-NCM Experiments

B.6.1 Explicit Counterfactual Modeling

The GAN-NCM is an *implicit* version of the NCM, where it outputs direct samples of the data. This is in contrast to an *explicit* version of the NCM such as the maximum likelihood estimation and which explicitly outputs the probabilities and likelihood values (Xia et al., 2021). An explicit approach requires modifications to the NCM in contrast to what is stated in Def. 2. Notably, the functions must be adapted to output probabilities instead of values from the variable domains, and sampling is not as straightforward (e.g., a trick through Gumbel random variables may be needed, and samples will not have gradients). This means that the method outlined in Alg. 3 is incompatible, and accomplishing the same task using an explicit model will require a significantly more complicated procedure. First, recall the construction of an MLE-NCM (generalized to high dimensions):

$$\widehat{M}(\boldsymbol{\theta}) := \begin{cases} \mathbf{V} & := \mathbf{V},\ \widehat{\mathbf{U}} := \{U_{\mathbf{C}} : \mathbf{C} \in \mathbb{C}(\mathcal{G})\} \cup \{G_{V_i} : V_i \in \mathbf{V}\}, \\ \widehat{\mathcal{F}} & := \left\{ f_{V_i} := \arg\max_{v_i \in \mathcal{D}_{V_i}} g_{V_i = v_i} + \log \sigma(\phi_{V_i}(\mathbf{pa}_{V_i}, \mathbf{u}^c_{V_i}; \theta_{V_i}))_{v_i} : V_i \in \mathbf{V} \right\}, \\ P(\widehat{\mathbf{U}}) & := \{U_{\mathbf{C}} \sim \text{Unif}(0,1) : U_{\mathbf{C}} \in \mathbf{U}\} \cup \\ & \quad \{G_{V_i = v_i} \sim \text{Gumbel}(0,1) : V_i \in \mathbf{V}, v_i \in \mathcal{D}_{V_i}\}, \end{cases} \tag{12}$$

where $\sigma : \mathbb{R}^d \to (0,1)^d$ is the softmax activation function; $\mathbb{C}(\mathcal{G})$ is the set of maximal confounded cliques of $\mathcal{G}$; each $G_{V_i = v_i}$ is a standard Gumbel random variable (Gumbel, 1954); 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 $\mathbf{U}^c_{V_i} := \{U_{\mathbf{C}} : U_{\mathbf{C}} \in \mathbf{U} \text{ s.t. } V_i \in \mathbf{C}\}$.

Let $P(\mathbf{y}_*) = P(\mathbf{y}_{1[\mathbf{x}_1]}, \mathbf{y}_{2[\mathbf{x}_2]}, \dots)$ be an unconditional counterfactual quantity. Then, one can compute

$$P^{\widehat{M}}(\mathbf{y}_*) = \sum_{\mathbf{v}_j = (\mathbf{w}_j, \mathbf{y}_j) | \forall j, \forall \mathbf{w}_j \in \mathcal{D}_{\mathbf{V} \setminus \mathbf{Y}_j}} P^{\widehat{M}} \left(\bigwedge_j \mathbf{V}_{\mathbf{x}_j} = \mathbf{v}_j \right). \tag{13}$$

Denoting $P^{\widehat{M}}(\mathbf{v}_*) := P^{\widehat{M}} \left(\bigwedge_j \mathbf{V}_{\mathbf{x}_j} = \mathbf{v}_j \right)$ and $\mathbf{v}_j = (v_{1j}, v_{2j}, \dots, v_{|\mathbf{V}|j})$, this can be computed as

$$P^{\widehat{M}}(\mathbf{v}_*) = \mathbb{E}_{P(\mathbf{U}^c)} \left[\prod_{v_{ij} | \forall j, \forall V_i \in \mathbf{V} \setminus \mathbf{X}_j} \sigma_{v_{ij}} \right] \approx \frac{1}{m} \sum_{k=1}^{m} \prod_{v_{ij} | \forall j, \forall V_i \in \mathbf{V} \setminus \mathbf{X}_j} \sigma_{v_{ij}}, \tag{14}$$

where $\sigma_{v_{ij}}$ denotes $\sigma(\phi_{V_i}(\mathbf{pa}_{V_i,j}, \mathbf{u}^c_{V_i,k}; \theta_{V_i}))_{v_i}$ ($\mathbf{pa}_{V_i,j}$ refers to the corresponding values of $\mathbf{Pa}_{V_i}$ in $\mathbf{v}_j$ from Eq. 13), the likelihood output of the model, $P^{\widehat{M}}(V_i = v_{ij} \mid \mathbf{pa}_{V_i,j}, \mathbf{u}^c_{V_i,k})$. The right hand side of the equation is a Monte-Carlo approximation of the expectation over $P(\mathbf{U}^c)$ using $k \in \{1, \dots, m\}$ samples of $\mathbf{U}^c$, the set of uniform random variables of $\widehat{\mathbf{U}}$.

A conditional counterfactual can then be computed as

$$P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) = \frac{P^{\widehat{M}}(\mathbf{y}_*, \mathbf{x}_*)}{P^{\widehat{M}}(\mathbf{x}_*)}. \tag{15}$$

We work in the log space for numerical stability. Denote $\{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$ as the data sets of $\mathbb{Z} = \{P^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}$, where $\mathbf{v}_{\mathbf{z}_k,i}$ refers to the ith data point of $\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})$ out of n_k total data points. Denote $Q = P(\mathbf{y} \mid \mathbf{x})$ as the query of interest. The MLE-NCM can be trained to fit the data sets as well as maximize the query via the loss

$$L\left(\widehat{M}, \{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k})\}_{k=1}^{\ell}\right) := \sum_{k=1}^{\ell} \frac{1}{n_k} \sum_{i=1}^{n} -\log P^{\widehat{M}}(\mathbf{v}_{\mathbf{z}_k,i}) - \lambda \log P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*) \tag{16}$$

Algorithm 4: Training MLE-NCM

Input : Data $\{\widehat{P}^{\mathcal{M}^*}(\mathbf{V}_{\mathbf{z}_k}) = \{\mathbf{v}_{\mathbf{z}_k,i}\}_{i=1}^{n_k}\}_{k=1}^{\ell}$, query $Q = P(\mathbf{y}_*|\mathbf{x}_*)$, causal diagram $\mathcal{G}$, number of Monte Carlo samples m, regularization constant λ, learning rate η, training epochs T

$\widehat{M} \leftarrow \texttt{NCM}(\mathbf{V}, \mathcal{G})$ `// from Eq. 12`
Initialize parameters $\boldsymbol{\theta}_{\min}$ and $\boldsymbol{\theta}_{\max}$
for $t \leftarrow 1$ **to** T **do**
  for $k \leftarrow 1$ **to** ℓ **do**
    for $i \leftarrow 1$ **to** n **do**
      `// Estimate from Eq. 15`
      $\hat{p}_{\min} \leftarrow \texttt{Estimate}(P^{\widehat{M}(\boldsymbol{\theta}_{\min})}(\mathbf{v}_{\mathbf{z}_k,i}), m)$
      $\hat{p}_{\max} \leftarrow \texttt{Estimate}(P^{\widehat{M}(\boldsymbol{\theta}_{\max})}(\mathbf{v}_{\mathbf{z}_k,i}), m)$
      $\hat{q}_{\min} \leftarrow \texttt{Estimate}(P^{\widehat{M}(\boldsymbol{\theta}_{\min})}(\mathbf{y}_* \mid \mathbf{x}_*), m)$
      $\hat{q}_{\max} \leftarrow \texttt{Estimate}(P^{\widehat{M}(\boldsymbol{\theta}_{\max})}(\mathbf{y}_* \mid \mathbf{x}_*), m)$
      `// L from Eq. 16`
      $L_{\min} \leftarrow -\log \hat{p}_{\min} - \lambda \log(1 - \hat{q}_{\min})$
      $L_{\max} \leftarrow -\log \hat{p}_{\max} - \lambda \log \hat{q}_{\max}$
      $\boldsymbol{\theta}_{\min} \leftarrow \boldsymbol{\theta}_{\min} - \eta \nabla L_{\min}$
      $\boldsymbol{\theta}_{\max} \leftarrow \boldsymbol{\theta}_{\max} - \eta \nabla L_{\max}$

where λ is set initially to a high value and decreases during training. The left term is the negative log-likelihood of each data point in each data set, and the right term is the negative log-likelihood of the query. To minimize the query, we instead subtract $\lambda \log \left(1 - P^{\widehat{M}}(\mathbf{y}_* \mid \mathbf{x}_*)\right)$ in the right term. The entire training procedure is shown in Alg. 4.

Nested counterfactuals can be computed as described in Appendix B.2, but in training, all computations of nested terms will not be able to have gradients due to the sampling requirement.

B.6.2 Model Architecture

Following the discussion in Sec. B.6.1, we construct each $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ using a masked autoencoder for density estimation (MADE) (Germain et al., 2015). We use Andrej Karpathy's implementation of MADE (Karpathy, 2018), which is provided with the MIT License. Each function has 2 hidden layers of size 64, and outputs are passed through a log sigmoid activation to be converted into a probability. Weights are initialized using Glorot initialization. The models are trained using the Adam optimizer.

B.6.3 Identification Experiment Procedure

The identification procedure largely matches that of the GAN-NCM. We ran 20 trials for each setting, each with 4 reruns, and $n = 10^4$ data points from the generating RCM. Positivity is enforced.

For each rerun, one model is initialized and trained to fit the data while simultaneously maximizing the query, then a second model is initialized and trained to fit the data with simultaneously minimizing the query. Results are computed afterwards from the two models.

Each training epoch follows the procedure in Alg. 4. For each dataset in $\mathbb{Z}$, a likelihood value for each datapoint type is evaluated from the NCM, $\widehat{M}$. These likelihood values are maximized, weighted by the frequency of the datapoint. For the query loss, the likelihood values of the query distribution are either minimized or maximized. The full batch is used each epoch.

We use a learning rate of $\eta = 4 \cdot 10^{-3}$ and run for 1000 epochs. We start λ at 1 and exponentially decrease to 10^{-3} by the end of training. We use $m = 10^4$ Monte Carlo samples for estimating Eq. 15 during training.

B.6.4 ESTIMATION EXPERIMENT PROCEDURE

As with the GAN-NCM, for estimation experiments, only one model is trained, the query loss $\mathbb{D}_Q$ is not computed, and optimization is purely to fit the datasets in $\mathbb{Z}$. For each setting in Fig. 6, we ran 5 different settings of n, increasing exponentially from 10^3 to 10^5, but only for $d = 1$ due to tractability issues with higher dimensions. We ran 10 trials for each setting, with means and confidence intervals computed across these 10 trials.

We use the full batch per epoch. We use a learning rate of $\eta = 4 \cdot 10^{-3}$, and run for up to 3000 epochs with early stopping. We use $m = 10^6$ Monte Carlo samples to evaluate the query from $\widehat{M}$ via the Gumbel-max trick.

B.7 RUNTIME EXPERIMENT PROCEDURE

For the runtime experiments procedure, we conducted NCM-GAN and NCM-MLE on the same dataset generated in the first graph of Fig. 4 as the dimensionality d of Z scales from 1 to 7. Similar to estimation experiments, only one model is trained, the query loss is not computed, and optimization is purely to fit the datasets. We record the first 100 epochs of training time for both methods. For each setting in Fig. 5, positivity is enforced when generating data and $n = 10000$. We ran 4 trials for each setting, with means and confidence intervals computed across these 4 trials.

For GAN-NCM, we use a learning rate of $\eta = 2 \cdot 10^{-5}$ and a batch size of 1000. For MLE-NCM, we use a learning rate of $\eta = 4 \cdot 10^{-3}$ and the full batch per epoch.

B.8 HARDWARE

The MLE-NCM was trained on NVIDIA Tesla V100 GPUs provided by Amazon Web Services. All other experiments are run via jobs on a cluster, and each thread may be assigned either an NVIDIA K80 or NVIDIA P100 GPU. In total, approximately 10000 GPU hours were used for experiments.

C CONNECTION TO RELATED WORKS

This paper evaluates causal queries directly through a parameterized model in the form of an SCM trained on available data from the true SCM $\mathcal{M}^*$, which we call a proxy model-based approach. In this section, we discuss the differences between the NCM presented in this paper and the models used in other works that also implement a proxy approach.

First, we note that many of these works implicitly benefited from the constraints of enforcing a causal diagram into the model structure without proving any properties about these constraints. One contribution of our work is to provide some explicit results about these constraints via Thms. 1 and 2. Second, and more importantly, most of these works operated under different assumptions and problem settings, to be discussed in individual sections. When applied to our problem setting, these methods may not be as general as the method discussed in this paper. Some of these losses of generality are highlighted in the following propositions and illustrated through the further examples provided in App. D.

Proposition 1. *Any model in which the Markovianity assumption (also known as causal sufficiency) is enforced is not $\mathcal{L}_3$-$\mathcal{G}$ expressive (i.e. an equivalent statement of Thm. 2 cannot be proven about such a model class).* ■

Proof. See counterexample provided in Example 6 in Sec. D.3. □

Proposition 2. *There exist settings in which an $\mathcal{L}_3$ query that is not identifiable from $\mathcal{L}_1$ data ($P(\mathbf{V})$) alone may be identifiable given data sets from $\mathcal{L}_2$.* ■

Proof. Examples include ID settings 9, 10, 11, and 12 from Fig. 4, in which their counterparts without the $P(\mathbf{V}_x)$ datasets are not identifiable. Another example is shown in Example 9 in Sec. D.3. □

Proposition 3. *There exist settings in which an $\mathcal{L}_3$ query is not identifiable even in the Markovian case and given both $\mathcal{G}$ and all datasets from $\mathcal{L}_2$.* ■

Proof. Example is shown in Example 8 from Sec. D.3. Further, this is an example of a case where the query cannot even be bounded.

□

We discuss more specifically the differences with each paper in the following subsections. Whenever notation differs between the two works, we use our notation to maintain consistency and make the comparison more transparent.

C.1 KOCAOGLU ET AL. (2018)

Similar to our work, Kocaoglu et al. (2018) used an implicit approach to fit a proxy SCM on the data via generative adversarial networks. The method was specifically trained on observational ($\mathcal{L}_1$) data and used to evaluate interventional ($\mathcal{L}_2$) queries in the Markovian setting. Like for a $\mathcal{G}$-NCM, their approach used a proxy model $\widehat{M} = \langle \widehat{\mathbf{U}}, \mathbf{V}, \widehat{\mathcal{F}}, P(\widehat{\mathbf{U}}) \rangle$, where $\widehat{\mathbf{U}}$ consists of independent fixed noise variables and each $\hat{f}_{V_i} \in \widehat{\mathcal{F}}$ is defined such that $\hat{f}_{V_i}$ takes $\mathbf{Pa}_{V_i}$ and $\widehat{\mathbf{U}}_{V_i}$ as input. Here, $\mathbf{Pa}_{V_i}$ is the set of parents of V_i in the graph $\mathcal{G}$, and $\widehat{\mathbf{U}}_{V_i}$ is a set of noise variables specific to variable V_i.

This work introduced two novel GAN architectures for $\mathcal{L}_2$ inference, CausalGAN and CausalBEGAN, focused on applications in image generation. For this challenging task, the CausalGAN architecture was designed in a unique fashion in which the model was separated into two parts: one for generating the labels, and another for generating the image conditioned on these labels. This was in contrast to having one generator learning the entire joint distribution of labels and images. The approach for learning the image generator is similar to a conditional GAN in that the objective of the generator includes an additional term to maximize the prediction accuracy of a labeling network. However, this often results in a problem they described as *label-conditioned mode collapse*, where the generator may have been complacent in generating a small number of fixed samples that are easy to label. Hence, in addition to this labeler, CausalGAN also introduced an anti-labeler network. The anti-labeler also attempts to label the generated images, but the objective of CausalGAN simultaneously

maximizes the accuracy of the labeler while minimizing the accuracy of the anti-labeler. This prevents label-conditioned mode collapse because images that are too easily labeled will fail to minimize the accuracy of the anti-labeler. More formally, for a binary label l with probability $\rho = P(l = 1)$, the objective of the CausalGAN generator is:

$$\begin{aligned}
&\min_G \mathbb{E}_{x\sim p_g(x)}\left[\log\left(\frac{1-D(x)}{D(x)}\right)\right] \\
&- \rho\mathbb{E}_{x\sim p_g^1(x)}[\log(D_{LR}(x)] - (1-\rho)\mathbb{E}_{x\sim p_g^0(x)}[log(1-D_{LR}(x))] \\
&+ \rho\mathbb{E}_{x\sim p_g^1(x)}[\log(D_{LG}(x)] + (1-\rho)\mathbb{E}_{x\sim p_g^0(x)}[log(1-D_{LG}(x))],
\end{aligned} \tag{17}$$

where G is the generator, D is the discriminator, $p_g(x)$ is the distribution induced by the generator, $p_g^i(x)$ is the distribution conditioned on $l = i$, D_{LR} is the labeler, and D_{LG} is the anti-labeler. The loss is composed of the loss from the discriminator subtracted with the loss from the labeler and adding the loss from the anti-labeler. The paper proved that the generator successfully matched the true conditional image distribution when G was the optimum of this objective (Kocaoglu et al., 2018, Thm. 2). CausalBEGAN was another architecture introduced where optimization considered image quality margin and label margin. In both cases, interventions were computed using the mutilation procedure on the generating model $\widehat{M}$.

CausalGAN and CausalBEGAN were empirically shown to successfully learn the data distribution in several image generation tasks. Experiments demonstrated several settings in which intervening on a label resulted in a different image compared to conditioning on the label.

The reference produced impressive results in generating images under causal interventions using novel implicit GAN architectures. While the approach is similar to our work in that a GAN proxy model is used, there are a number of differences with our work.

1. The paper worked in the space of inferring $\mathcal{L}_2$ quantities given observational data from $\mathcal{L}_1$. Our work generalizes this setting to inferring any counterfactual query from $\mathcal{L}_3$ given any combination of datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$.
2. Unlike this work, we do not make the Markovianity assumption, meaning we allow unobserved confounding between variables. Many real world settings often have unobserved confounding, so the Markovianity assumption can be limiting. In a situation where the true SCM has unobserved confounding, a model class which enforces Markovianity will not be expressive enough to model the true SCM (Prop. 1).
3. By (Kocaoglu et al., 2018, Def. 1), the architecture of CausalGAN took the form of a causal diagram $\mathcal{G}$, similar to how the $\mathcal{G}$-constrained NCM is constructed in Def. 2. In our work, we discuss the theoretical properties of the implied constraints via Thms. 1 and 2. While $\mathcal{G}^{(\mathcal{L}_2)}$-consistency likely held for CausalGAN due to the nature of its construction, this property was not explicitly proven or discussed.
4. The fact that all $\mathcal{L}_2$ quantities are identifiable from $\mathcal{L}_1$ data was acknowledged (Kocaoglu et al., 2018, Prop. 1). The identification question constitutes an additional challenge in our paper since we consider layer 3 queries and allow for unobserved confounding. It's understood that without Markovianity, even inferences about $\mathcal{L}_2$ queries are not always identifiable from $\mathcal{L}_1$ data.

C.2 Pawlowski et al. (2020)

Pawlowski et al. (2020) implemented a deep proxy SCM approach in which each mechanism $\hat{f}_{V_i}$ models a conditional distribution $P(V_i \mid \mathbf{Pa}_{V_i})$ under the Markovianity assumption (lack of unobserved confounders). Under this assumption, the joint observational distribution $P(\mathbf{V})$ can be factorized as

$$P(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} P(V_i \mid \mathbf{Pa}_{V_i}), \tag{18}$$

so each mechanism represents one conditional distribution in this factorization. They proposed three novel approaches to modeling this conditional distribution.

1. The first approach involved modeling $P(X \mid \mathbf{Pa}_X)$ using normalizing flows. $\widehat{\mathbf{U}}_X$ consists of a noise variable ϵ, and $\hat{f}_X$ is an invertible function such that $x = \hat{f}_X(\epsilon; \mathbf{pa}_X)$. Then the explicit likelihood could be computed as $p(x \mid \mathbf{pa}_x) = p(\epsilon) \cdot |\det \nabla_\epsilon \hat{f}_X(\epsilon; \mathbf{pa}_x)|^{-1}$, where ϵ could be computed as $\hat{f}_X^{-1}(x; \mathbf{pa}_x)$.
2. The second approach extended the first approach to high dimensional settings by splitting ϵ into two independent noise variables, z and u, and splitting $\hat{f}_X$ into two components: a non-invertible function g and an invertible function h. Then $\hat{f}_X(\epsilon, \mathbf{pa}_x) = h(u; g(z; \mathbf{pa}_x), \mathbf{pa}_x)$, where g is expected to capture the high-level aspects of the data. This is helpful in high dimensional settings as the size of the invertible component h can be reduced because working with the outputs of g is easier than directly working with the data. Since $p(x \mid \mathbf{pa}_x)$ is no longer tractable, the model is instead trained using variational inference by maximizing the evidence lower bound (ELBO) with variational distribution $Q(z \mid x, \mathbf{pa}_x)$,
$$\log p(x \mid \mathbf{pa}_x) \geq \mathbb{E}_Q(z \mid x, \mathbf{pa}_x)[\log p(x \mid z, \mathbf{pa}_x)] - D_{\mathrm{KL}}[Q(z \mid x, \mathbf{pa}_x) || P(z)]. \quad (19)$$
3. Although not demonstrated in the paper, a third approach was presented suggesting the use of a GAN-based approach to implicitly model the likelihood.

The method primarily used in the paper was the 2nd approach. They evaluated counterfactual quantities using the 3-step procedure from Pearl (2000, Thm. 7.1.7):

1. **Abduction** – The observations $\mathbf{v}$ of all variables $\mathbf{V}$ are used to predict $P(\mathbf{U} \mid \mathbf{v})$, where $P(\mathbf{U} \mid \mathbf{v})$ decomposes as $\prod_{V_i \in V} P(\epsilon_{V_i} \mid v_i, \mathbf{pa}_{V_i})$ due to the Markovianity assumption. This is then computed as
$$\begin{aligned} P(\epsilon_{V_i} \mid v_i, \mathbf{pa}_{V_i}) &= P(z_{V_i} \mid v_i, \mathbf{pa}_{V_i}) P(u_{V_i} \mid z_{V_i}, v_i, \mathbf{pa}_{V_i}) \\ &\approx Q(z_{V_i} \mid e_{V_i}(v_i; \mathbf{pa}_{V_i})) \delta(u_{V_i}) \end{aligned} \quad (20)$$
where $e(v_i; \mathbf{pa}_{V_i})$ is a deterministic encoder mapping v_i and $\mathbf{pa}_{V_i}$ to z_{V_i}, and δ is a Dirac delta distribution centered at $h_{V_i}^{-1}(v_i; g_{V_i}(z_{V_i}; \mathbf{pa}_{V_i}), \mathbf{pa}_{V_i})$.
2. **Action** – For an intervention over sets of variables $\mathbf{X}$, each mechanism $\hat{f}_X$ for $X \in \mathbf{X}$ is replaced either with a fixed value x (atomic) or a surrogate mechanism $\tilde{f}_X(\epsilon_X \mid \widetilde{\mathbf{pa}}_X)$ (stochastic).
3. **Prediction** – Under this distribution of $P(\mathbf{U} \mid \mathbf{v})$, and with the modified mechanisms, the deep SCM is evaluated to produce a result for the counterfactual quantity $P(\mathbf{V_x} \mid \mathbf{v})$.

The approach was demonstrated empirically on a synthetic MNIST dataset and on a real world brain imaging dataset.

The paper presented a powerful approach to modeling a deep SCM using normalizing flows and variational inference, allowing for efficient and precise training even in high-dimensional cases. Still, there are a number of differences between this paper and our work.

1. Unlike this work, we do not make the Markovianity assumption, meaning we allow unobserved confounding between variables. Many real world settings often have unobserved confounding, so the Markovianity assumption can be limiting. In a situation where the true SCM has unobserved confounding, a model class which enforces Markovianity will not be expressive enough to model the true SCM (Prop. 1). Moreover, it is not a trivial task to extend the normalizing flow approach in this paper to a case with unobserved confounders. In such a case, Eq. 18 would no longer be valid, and it is not immediately clear how to train invertible mechanisms that can share noise.
2. Like our work, this paper assumed that the causal graph $\mathcal{G}$ is provided. However, in our work, we discuss the theoretical properties of the implied constraints via Thms. 1 and 2, while this paper implicitly benefited from these constraints without discussing these properties.
3. The identification problem is not considered in this work while it is a major focus in our work. As shown in Prop. 3, there exist counterfactual quantities that cannot be identified given the graph $\mathcal{G}$, and all available data from $\mathcal{L}_1$ and $\mathcal{L}_2$, even under the Markovianity assumption. If these quantities were to be estimated from a model trained on such data, the result will likely be incorrect and/or misleading. For that reason, we make sure to identify all queries before attempting to estimate them.

4. While this paper only considered input in the form of observational data from $\mathcal{L}_1$, our work considers cases where additional experimental datasets from $\mathcal{L}_2$ are provided (even cases where perhaps the observational distribution $P(\mathbf{V})$ is not provided). As shown in Prop. 2, there do exist counterfactual queries that are not identifiable given observational data but are identifiable given datasets from $\mathcal{L}_2$.
5. This paper evaluated counterfactual quantities using the 3-step procedure of abduction, action, and prediction. While this is a sound algorithm for evaluating applicable quantities, it does not apply generally to all counterfactual quantities. Specifically, it only applies in quantities of the form $P(\mathbf{Y_x} \mid \mathbf{z})$, in which only one intervention is applied before the conditioning bar, and observations are present after the conditioning bar. This excludes quantities such as $P(y_x, y'_{x'})$, which is a joint probability of two counterfactual terms in different worlds (one in $X = x$ and another in $X = x'$), as well as quantities such as $P(y_x \mid y'_{x'})$, in which a term in an intervened world ($X = x'$) is after the conditioning bar.
6. Finally, although an implicit approach via GANs was discussed in (Pawlowski et al., 2020), it was not explored. Our paper realizes an implementation of GAN approach in the non-Markovian setting.

C.3 WITTY ET AL. (2021)

Witty et al. (2021) introduced simulation-based identifiability (SBI), which uses a Bayesian approach to solve the identification problem for $\mathcal{L}_2$ and some $\mathcal{L}_3$ queries given $\mathcal{L}_1$ data. In their approach, the goal was to identify a particular query Q given a fixed observational distribution $P(\mathbf{V})$ and a causal diagram $\mathcal{G}$. An SCM $\mathcal{M}^* = \langle \mathbf{U}^*, \mathbf{V}, \mathcal{F}^*, P(\mathbf{U}^*) \rangle$ is sampled from a prior over SCMs (specifically their functions and exogenous variables), $P(\mathcal{F}, \mathbf{U})$. In each iteration, n samples are drawn from $P^{\mathcal{M}^*}(\mathbf{V})$, the observational dataset of $\mathcal{M}^*$. Similar to our work, two parameterized SCMs, $\widehat{M}_1$ and $\widehat{M}_2$ are optimized to simultaneously fit $P^{\mathcal{M}^*}(\mathbf{V})$ while maximizing the difference in the induced queries of the two SCMs, $\widehat{Q}_1$ and $\widehat{Q}_2$. This is done by maximizing the objective

$$L(\widehat{M}_1, \widehat{M}_2, \{\mathbf{v}_i\}_{i=1}^n) = \sum_{i=1}^{n} \Big(\log P^{\widehat{M}_1}(\mathbf{v}_i) + \log P^{\widehat{M}_2}(\mathbf{v}_i) \Big) + \lambda |\widehat{Q}_1 - \widehat{Q}_2|, \tag{21}$$

where $\{\mathbf{v}_i\}_{i=1}^n$ are a collection of n samples from $P^{\mathcal{M}^*}(\mathbf{V})$. Maximizing this objective is equivalent to jointly maximizing the likelihood of the data for both $\widehat{M}_1$ and $\widehat{M}_2$ as well as maximizing the gap between $\widehat{Q}_1$ and $\widehat{Q}_2$. Identifiability is concluded if, following a hypothesis testing procedure, $|\widehat{Q}_1 - \widehat{Q}_2|$ is under a certain threshold. Non-identifiability is concluded otherwise. They proved that Q is identifiable w.r.t. $\mathcal{M}^*$ if and only if there does not exist any parameterized SCM $\widehat{M}$ fitted to the data from $P^{\mathcal{M}^*}(\mathbf{V})$ such that $P^{\widehat{M}}(\mathbf{V})$ converges to $P^{\mathcal{M}^*}(\mathbf{V})$ as $n \to \infty$, $Q^{\widehat{M}} \neq Q^{\mathcal{M}^*}$, and $\widehat{M}$ is in the support of the prior (Witty et al., 2021, Thm. 3.2).

In non-parametric settings, there exist sound and complete symbolic methods such as do-calculus that could efficiently and correctly identify any $\mathcal{L}_2$ query Pearl (2000). Hence, as we understand, the paper primarily focused its attention on parametric settings, where the model class of SCMs was further restricted to specific function classes (e.g. linear), and symbolic approaches to identification under this constrained class were not as well-studied.

For the identification task, SBI shares many similarities with our approach in Alg. 1, notably that they also trained two proxy SCMs to minimize/maximize the query while simultaneously fitting the data. Still, there are many differences between the two works.

1. The paper considered a general form of the query $Q(\mathbf{V}_{\mathbf{X}=\mathbf{x}}, \mathbf{V}_{\mathbf{X}=\mathbf{x}'})$, which was a function of the counterfactual terms of $\mathbf{V}$ in the two worlds of $\mathbf{X} = \mathbf{x}$ and $\mathbf{X} = \mathbf{x}'$. For example, average treatment effect (ATE) given a binary treatment X could be computed as $Q(\mathbf{V}_{\mathbf{X}=\mathbf{x}}, \mathbf{V}_{\mathbf{X}=\mathbf{x}'}) = \mathbb{E}[Y_{X=1}] - \mathbb{E}[Y_{X=0}]$, which was what was used in their experiments. While in theory, this may have included certain counterfactual queries, these queries were not explored, so the work largely focused on the space of $\mathcal{L}_2$ queries. Moreover, they only considered data from $\mathcal{L}_1$. Our work generalizes this setting to inferring any counterfactual query from $\mathcal{L}_3$ given any combination of datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$.

2. Like our work, this paper assumed that the causal graph $\mathcal{G}$ is provided. However, in our work, we discuss the theoretical properties of the implied constraints via Thms. 1 and 2, while this paper implicitly benefited from these constraints without discussing these properties.
3. The paper was largely concerned with identifying quantities in parametric cases (and functions of SCMs were fixed to certain parametric families), while our work is largely focused on the non-parametric case. We make no claims about expressivity or $\mathcal{G}$-consistency in parametric settings.
4. Our work assumes that there exists a true SCM $\mathcal{M}^*$, which generates a specific dataset from which we aim to identify and estimate our query. In contrast, (Witty et al., 2021) specifically aimed to identify a query without a true SCM in mind and instead matched the distribution with an SCM sampled from a prior distribution over the space of SCMs. While estimation could theoretically be performed after concluding identifiability with SBI, this was not a focus of (Witty et al., 2021), and the estimation would not be meaningful unless $\mathcal{M}^*$ corresponded to some true SCM in a real application.
5. In terms of the approach, (Witty et al., 2021) differs from our work in that the two parametrized models, $\widehat{M}_1$ and $\widehat{M}_2$ are optimized jointly, while in our case, the two models are optimized separately. The goal in SBI is to maximize the gap between the two models in the query, while Alg. 1 does this implicitly when maximizing or minimizing the query in the two models separately. Further, our implementation in practice utilizes GANs, which is not explored in this paper.

C.4 Xia et al. (2021)

Xia et al. (2021) introduced the NCM, which is the model class considered in our paper. They proved that NCMs are expressive on all three layers (Xia et al., 2021, Thm. 1) but then subsequently proved the Neural Causal Hierarchy Theorem (Xia et al., 2021, Corol. 1) which states that despite this expressivity, an NCM trained only on lower layer data would almost never match the true SCM on higher layers. To proceed, they introduced a graphical inductive bias in the form of a causal diagram $\mathcal{G}$ derived from the true SCM. They defined a $\mathcal{G}$-constrained version of the NCM (Def. 2 in our paper).

They proved that the class of $\mathcal{G}$-constrained NCMs is both $\mathcal{G}$-consistent (Xia et al., 2021, Thm. 2) and $\mathcal{L}_2$-$\mathcal{G}$ expressive (Xia et al., 2021, Thm. 3). $\mathcal{G}$-consistency ($\mathcal{G}^{(\mathcal{L}_2)}$-consistency in the context of our work) constrained the class to a smaller set of models which may have potentially agreed on the causal query, while $\mathcal{L}_2$-$\mathcal{G}$ expressiveness guaranteed that $\mathcal{G}$-NCMs could still model any SCM within this constrained space. Still, these results only considered up to layer 2 of the causal hierarchy, and made no claims about layer 3. In contrast, Thms. 1 and 2 in our work are much stronger claims, proving these results on layer 3. That is, $\mathcal{G}$-NCMs have all of the constraints implied on the third layer while still being able to express any true SCM on all three layers within this constrained setting.

The paper then defined the neural identification problem specifically for $\mathcal{L}_2$ queries. They proved a duality stating that neural identification for $\mathcal{L}_2$ quantities is equivalent to graphical identification (Xia et al., 2021, Thm. 4) and showed that the mutilation procedure applied to a proxy $\mathcal{G}$-NCM can be used to estimate identifiable queries (Xia et al., 2021, Corol. 2). This definition of neural identification did not consider $\mathcal{L}_3$ quantities or $\mathcal{L}_2$ input data. In our case, we define neural counterfactual identification (Def. 4) generalizing the neural identification problem to handle $\mathcal{L}_3$ queries given arbitrary datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$. We prove a stronger duality in this setting (Thm. 3) and that counterfactual quantities can be evaluated directly from the proxy $\mathcal{G}$-NCM, akin to the mutilation procedure for $\mathcal{L}_2$ quantities (Corol. 1).

Following these identification results, (Xia et al., 2021) introduced an algorithm (Xia et al., 2021, Alg. 1) which is both sound and complete for identifying and estimating $\mathcal{L}_2$ queries (Xia et al., 2021, Corol. 4). The algorithm was implemented in practice through a maximum likelihood approach (MLE-NCM) through (Xia et al., 2021, Alg. 2). This involved training two $\mathcal{G}$-NCM parameterizations, where one was trained to maximize the query, while the other was trained to minimize it, and both models were trained to fit the observational data $P(\mathbf{V})$. The algorithm in our counterfactual setting (Alg. 1) is similar on the surface, but there are many additional challenges in solving this more general problem. First, while an interventional query might be easy to compute and optimize, it is not as clear how to compute a counterfactual quantity with gradients such that it can be maximized or minimized. After all, counterfactual quantities often have complicated forms such as nesting (e.g.

$P(Y_{Z_X})$) or having the same variable in multiple worlds (e.g. $P(Y_{X=0}, Y_{X=1})$). Second, fitting a single dataset is standard practice, which is what needs to be done when only observational data is available. It is not as clear how to simultaneously fit several datasets. Our work solves both of these problems in practice using the GAN-NCM approach in Sec. 4, but also explains how one might solve this problem using the MLE-NCM in Sec. B.6.1.

To demonstrate the ability of the MLE-NCM to identify and estimate $\mathcal{L}_2$ queries in practice, the paper showed results from identifying average treatment effect (ATE) from 8 typical graphs encountered in the literature. They also provided estimation results for the identifiable cases. While our experiments mainly focused on identifying $\mathcal{L}_3$ queries, we reproduced the experiments of (Xia et al., 2021) in Sec. B.1 using the GAN-NCM to demonstrate that our approach is general and can be applied to the $\mathcal{L}_2$ case as well. Further, although the reason why one might prefer to use an NCM for causal inference is due to the ability of neural models to scale inferences to high dimensions, this ability was not emphasized in the experimental results of this paper. We explicitly showed how the GAN-NCM can perform nearly as well even in 16-dimensional cases while showing that the MLE-NCM scales poorly due to tractability issues (see Fig. 5 in Sec. 5). One additional issue of the MLE-NCM is its bias towards returning ID results even in non-ID cases. This is evident in Fig. 4 and can be potentially problematic if used in practice since attempting to estimate a non-ID query will likely produce an incorrect and misleading result. For more details, see the discussion surrounding Fig. 14 in Sec. B.1.

To summarize, this paper introduced the NCM and proved several properties about it, including results on expressiveness, constraints, and identification. Our work also uses the NCM and proves many analogous results, but there are still several differences:

1. The paper primarily worked in the space of inferring $\mathcal{L}_2$ quantities given observational data from $\mathcal{L}_1$. Our work generalizes this setting to inferring any counterfactual query from $\mathcal{L}_3$ given any combination of datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$. This is reflected in the theoretical results, the algorithms, and the practical implementation. We show that the $\mathcal{G}$-NCM has all $\mathcal{L}_3$ constraints (Thm. 1) but can still express any model on all three layers in this constrained setting (Thm. 2). We show that the $\mathcal{G}$-NCM can be used to solve the identification and estimation problem in the more general counterfactual setting (Thm. 3, Corol. 1, Alg. 1, Corol. 2). These results drastically generalize the capabilities of the NCM to $\mathcal{L}_3$.
2. While the MLE-NCM implemented in this paper is an option for counterfactual identification and estimation (as shown in Sec. B.6.1), we develop a new type of architecture for the NCM using a generative adversarial approach. Our approach, detailed in Sec. 4, has many advantages over the MLE-NCM, notably its design simplicity, its ability to scale to high dimensions, and its robustness in solving the ID problem. We even show that the GAN-NCM can solve all of the experimental settings originally presented in (Xia et al., 2021) in Sec. B.1.

From all of these results, both theoretical and empirical, it is evident that our approach, in fact, subsumes the work in (Xia et al., 2021).

C.5 Zecevic et al. (2021)

Zecevic et al. (2021) related the class of graph neural networks (GNNs) to the class of Markovian SCMs. Given a DAG $\mathcal{G}$ with variables $\mathbf{V}$, they defined GNN layers using the functions

$$h_{V_i} = \phi\left(d_{V_i}, \bigoplus_{V_j \in \mathcal{N}(V_i)} \psi(d_{V_i}, d_{V_j})\right), \tag{22}$$

where d_{V_i} is either data associated with V_i or a feature computed from a previous layer, $\mathcal{N}(V_i)$ is the set of neighbors of V_i in $\mathcal{G}$, $\bigoplus$ refers to an aggregating function such as sum, and h_{V_i} is a feature of V_i for the current layer. They proved (Zecevic et al., 2021, Thm. 1), which states that for any Markovian SCM $\mathcal{M} = \langle \mathbf{U}, \mathbf{V}, \mathcal{F}, P(\mathbf{U})\rangle$, there exists a GNN construction such that $h_{V_i} = f_{V_i}$ for all $f_{V_i} \in \mathcal{F}$, and such a construction (but without shared ϕ and ψ) is therefore a special case of the NCM introduced in (Xia et al., 2021) ((Zecevic et al., 2021, Corol. 1)). Using a GNN interpretation of an SCM, one could compute quantities from an intervention $\mathbf{X} = \mathbf{x}$ through the mutilation approach by evaluating Eq. 22 but with edges into variables of $\mathbf{X}$ removed from $\mathcal{N}$.

Leveraging results from variational graph autoencoders (VGAEs), (Zecevic et al., 2021) demonstrated an approach to learning the observational distribution with a GNN architecture. They introduced

interventional VGAEs (iVGAEs) which defined inferences from VGAEs on layer 2 quantities via the mutilation procedure on the GNN layers. They proved a wealth of results related to this class of models following similar results proven in (Xia et al., 2021): $\mathcal{L}_2$-expressivity (Zecevic et al., 2021, Thm. 3), a causal hierarchy theorem result between layers 1 and 2 (Zecevic et al., 2021, Corol. 2), and equivalence between neural-identification from iVGAEs and classical identification of $\mathcal{L}_2$ quantities (Zecevic et al., 2021, Thm. 4).

(Zecevic et al., 2021) introduced an interesting discussion on the relationship between GNNs, SCMs, and NCMs. The work differs from our work in a variety of ways.

1. Like (Xia et al., 2021), (Zecevic et al., 2021) primarily worked in the space of inferring $\mathcal{L}_2$ quantities given observational data from $\mathcal{L}_1$. Our work generalizes this setting to inferring any counterfactual query from $\mathcal{L}_3$ given any combination of datasets from $\mathcal{L}_1$ and $\mathcal{L}_2$.
2. Unlike this work, we do not make the Markovianity assumption, meaning we allow unobserved confounding between variables. Many real world settings often have unobserved confounding, so the Markovianity assumption can be limiting. In a situation where the true SCM has unobserved confounding, a model class which enforces Markovianity will not be expressive enough to model the true SCM (Prop. 1). The results proven in this paper do not extend trivially to cases with unobserved confounders. Traditional GNNs are typically defined on adjacency matrices, which cannot describe bidirected edges in causal diagrams.
3. Like our work, this paper assumed that the causal graph $\mathcal{G}$ was provided. However, in our work, we discuss the theoretical properties of the implied constraints via Thms. 1 and 2. There does not appear to be a result in the paper demonstrating that $\mathcal{G}$-GNNs or iVGAEs are $\mathcal{G}^{(L_2)}$-consistent. It is not immediately obvious that $\mathcal{G}$-constrained GNNs have all of the $\mathcal{L}_2$ constraints of $\mathcal{G}$ given the unique architecture of GNNs.
4. The identification problem is one of the main focuses of our work. While the paper discussed the identification problem of $\mathcal{L}_2$ queries given $\mathcal{L}_1$ data, they did not provide a method of solving the problem using iVGAEs. Rather, they focused on the estimation problem under the assumption of identifiability. In fact, all $\mathcal{L}_2$ quantities are identifiable from $\mathcal{L}_1$ data under the Markovianity assumption (Bareinboim et al., 2022, Corol. 2), so an approach for solving the identification problem in this setting may not be meaningful.
5. Finally, we note that while this paper explored some novel aspects of using GNN and VGAE architectures for causal inference, our paper focuses on GANs, an entirely different class of models.

C.6 Sanchez-Martin et al. (2021)

Sanchez-Martin et al. (2021) introduced Variational Causal Autoencoders (VACA), a model for estimating interventional and counterfactual quantities under the Markovianity assumption using a variational graph autoencoder architecture (VGAE) and graph neural network (GNN) architecture like (Zecevic et al., 2021). VACA is defined as a tuple $\langle A, p(\mathbf{Z}), p_\theta, q_\phi \rangle$, where

- A is an adjacency matrix representing the causal DAG $\mathcal{G}$;
- $p(\mathbf{Z}) = \prod_{V_i \in \mathbf{V}} p(Z_{V_i})$ is a prior distribution over independent latent variables $\mathbf{Z} = \{Z_{V_i} : V_i \in \mathbf{V}\}$;
- The decoder $p_\theta(\mathbf{V} \mid \mathbf{Z}, A)$ is a GNN parameterized by θ mapping $\mathbf{Z}$ and A to the likelihood of $\mathbf{V}$ given these values;
- The encoder $q_\phi(\mathbf{Z} \mid \mathbf{V}, A)$ is a GNN parameterized by ϕ mapping $\mathbf{V}$ and A to the posterior approximation of $\mathbf{Z}$ given these values.

The model was trained to maximize the likelihood of observational data $P(\mathbf{V})$, which can be expressed as

$$P(\mathbf{V}) \approx \int p_\theta(\mathbf{V} \mid \mathbf{Z}, A) p(\mathbf{Z}) d\mathbf{Z}. \tag{23}$$

Here, $\mathbf{Z}$ is similar to the role of $\mathbf{U}$ in an SCM. Although $\mathbf{Z}$ does not have to be equal to $\mathbf{U}$ from the true SCM, each Z_{V_i} provides a source of noise for variable V_i that captures the same information that

$\mathbf{U}_{V_i}$ contributes to V_i in the true SCM. The difference between $\mathbf{Z}$ and $\mathbf{U}$ does not matter as long as the VACA model can match the true observational distribution.

During evaluation, interventional ($\mathcal{L}_2$) quantities can be estimated via

$$p(\mathbf{V}_{\mathbf{X}=\mathbf{x}}) \approx \int \int p_\theta(\mathbf{V} \mid \tilde{\mathbf{Z}}, \tilde{\mathbf{Z}}^{\mathcal{I}}, A^{\mathcal{I}}) p(\tilde{\mathbf{Z}}) q_\phi(\tilde{\mathbf{Z}}^{\mathcal{I}} \mid A^{\mathcal{I}}, \mathbf{X}) d\tilde{\mathbf{Z}} d\tilde{\mathbf{Z}}^{\mathcal{I}}, \tag{24}$$

where $A^{\mathcal{I}}$ is the adjacency matrix with edges into variables of $\mathbf{X}$ removed, $\tilde{\mathbf{Z}}^{\mathcal{I}}$ is the set of representations of $\mathbf{X}$, and $\tilde{\mathbf{Z}}$ is the subset of $\mathbf{Z}$ corresponding to $\mathbf{V} \setminus \mathbf{X}$. In words, intervened variables are assigned a representation from the encoder under a graph with the edges into $\mathbf{X}$ mutilated, while the remaining variables are assigned the same representation.

Further, the following counterfactual ($\mathcal{L}_3$) quantity can also be estimated:

$$\begin{aligned} &P(\mathbf{V}_{\mathbf{X}=\mathbf{x}} = \mathbf{v}^{CF} \mid \mathbf{v}^F) \\ &\approx \int \int p_\theta(\mathbf{V} = \mathbf{v}^{CF} \mid \tilde{\mathbf{Z}}^F, \tilde{\mathbf{Z}}^{\mathcal{I}}, A^{\mathcal{I}}) q_\phi(\tilde{\mathbf{Z}}^{\mathcal{I}} \mid \mathbf{x}, A^{\mathcal{I}}) q_\phi(\tilde{\mathbf{Z}}^F \mid \mathbf{v}^F, A) d\tilde{\mathbf{Z}}^{\mathcal{I}} d\tilde{\mathbf{Z}}^F, \end{aligned} \tag{25}$$

where $\mathbf{v}^F$ is the observed factual value of $\mathbf{V}$, $\mathbf{v}^{CF}$ is the counterfactual value of $\mathbf{V}$ to be queried, and $\tilde{\mathbf{Z}}^F$ is the set of representations associated with $\mathbf{v}^F$ except for the variables in $\mathbf{X}$. In words, the 3-step procedure for evaluating counterfactuals (Pearl, 2000, Thm. 7.1.7) is applied here. The abduction step is performed by computing the factual representations $\tilde{\mathbf{Z}}^F$ through the encoder $q_\phi(\tilde{\mathbf{Z}}^F \mid \mathbf{v}^F, A)$ applied to the factual observations $\mathbf{v}^F$. The action step is performed by assigning the intervened representations $\tilde{\mathbf{Z}}^{\mathcal{I}}$ as discussed in Eq. 24. Finally, the prediction step is performed by using the interventional representations $\tilde{\mathbf{Z}}^{\mathcal{I}}$ from $\mathbf{X}$ and the factual representations $\tilde{\mathbf{Z}}^F$ from $\mathbf{V} \setminus \mathbf{X}$ in the encoder $p_\theta(\mathbf{V} \mid \tilde{\mathbf{Z}}^F, \tilde{\mathbf{Z}}, A^{\mathcal{I}})$ with the mutilated graph $A^{\mathcal{I}}$ to produce a counterfactual outcome $\mathbf{v}^{CF}$ for $\mathbf{V}$ in this setting.

VACA presented interesting ideas for evaluating interventional and counterfactual quantities in the Markovian case using GNNs and VGAEs. Still, they differ from our work in several ways.

1. Unlike this work, we do not make the Markovianity assumption, meaning we allow unobserved confounding between variables. Many real world settings often have unobserved confounding, so the Markovianity assumption can be limiting. In a situation where the true SCM has unobserved confounding, a model class which enforces Markovianity will not be expressive enough to model the true SCM (Prop. 1). VACA does not extend trivially to cases with unobserved confounders. Traditional GNNs are typically defined on adjacency matrices, which cannot describe bidirected edges in causal diagrams.
2. Like our work, this paper assumed that the causal graph $\mathcal{G}$ was provided. However, in our work, we discuss the theoretical properties of the implied constraints via Thms. 1 and 2. This paper possibly benefited from these constraints implicitly without discussing these properties, but it is not actually clear if VACA is a $\mathcal{G}^{(\mathcal{L}_3)}$-consistent model.
3. The identification problem was not considered in this work while it is a major focus in our work. As shown in Prop. 3 above, there exist counterfactual quantities that cannot be identified given the graph $\mathcal{G}$, and all available data from $\mathcal{L}_1$ and $\mathcal{L}_2$, even under the Markovianity assumption. If these quantities were to be estimated from a model trained on such data, the result will likely be incorrect and misleading. For that reason, we make sure to identify all queries before attempting to estimate them.
4. While this paper only considered input in the form of observational data from $\mathcal{L}_1$, our work considers cases where additional experimental datasets from $\mathcal{L}_2$ are provided (even cases where perhaps the observational distribution $P(\mathbf{V})$ is not provided). As shown in Prop. 2, there do exist counterfactual queries that are not identifiable given observational data but are identifiable given datasets from $\mathcal{L}_2$.
5. This paper evaluated counterfactual quantities using the 3-step procedure of abduction, action, and prediction. While this is a sound algorithm for evaluating applicable quantities, it does not apply generally to all counterfactual quantities. Specifically, it only applies in quantities of the form $P(\mathbf{Y}_{\mathbf{x}} \mid \mathbf{z})$, in which only one intervention is applied before the conditioning bar, and observations are present after the conditioning bar. This excludes

quantities such as $P(y_x, y'_{x'})$, which is a joint probability of two counterfactual terms in different worlds (one in $X = x$ and another in $X = x'$), as well as quantities such as $P(y_x \mid y'_{x'})$, in which a term in an intervened world $(X = x')$ is after the conditioning bar.

6. Finally, we note that while this paper explored some novel aspects of using GNN and VGAE architectures for causal inference, our paper focuses on GANs, a different class of models.

D FURTHER DISCUSSIONS AND EXAMPLES

In this section, we supplement the main points of the paper with examples and expand on some subtleties. In Section D.1, we expand on the impossibility of performing inferences across layers of the PCH in general and why assumptions are necessary. In particular, we discuss the implications of the causal diagram assumption and compare with alternatives. In Section D.2, we expand on the reasoning behind introducing the discussion of constraints separately for each layer (as in Def. 3) using an example. In Section D.3, we ground the theoretical results of Secs. 2 and 3 by walking through examples with actual constructions of NCMs. In Section D.4, we discuss the Markovianity assumption, why it simplifies causal inference, why it unrealistic, and why the approach in this paper is therefore more general for *not* making this assumption.

D.1 NECESSITY OF CAUSAL ASSUMPTIONS

As emphasized at the beginning of Sec. 2, the entire discussion on constraints in this paper is due to the Neural Causal Hierarchy Theorem (N-CHT) (Xia et al., 2021, Corol. 1), following the Causal Hierarchy Theorem from Bareinboim et al. (2022) that states that distributions from lower layers of the hierarchy almost surely underdetermine higher layers in a measure-theoretic sense. In other words, it is not possible to make a statement about a higher layer (e.g. interventions, counterfactuals) with only knowledge from lower layers (e.g. observations). This result is well-understood in the causal inference field, and Pearl eloquently states in the Book of Why that "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 only use rung-one data and do not have a causal model) will never be able to answer questions about interventions (...)" (Pearl & Mackenzie, 2018, p. 32).

To further ground this discussion, consider the following examples.

Example 1. Consider a scenario where the board of education of a city is deciding on policies for school curricula. Data is collected from schools across the city, noting the textbooks (X) used in the schools and the GPA of students from that school (Y). For simplicity, let X and Y be binary variables, where $X = 1$ represents the use of a new textbook, and $Y = 1$ represents a high average GPA. Suppose the true model $\mathcal{M}^*$ describing the relationship between X and Y is specified as follows:

$$\mathcal{M}^* = \begin{cases} \mathbf{U} & = \{U_{XY}, U_{Y1}, U_{Y2}, U_{Y3}\} \\ \mathbf{V} & = \{X, Y\} \\ \mathcal{F} & = \begin{cases} f_X(u_{XY}) & = u_{XY} \\ f_Y(x, u_{XY}, u_{Y1}, u_{Y2}, u_{Y3}) & = \begin{cases} \neg x \wedge u_{Y1} & \text{if } u_{XY} = 0 \\ (x \oplus u_{Y2}) \vee u_{Y3} & \text{if } u_{XY} = 1 \end{cases} \end{cases} \\ P(\mathbf{U}) & = U_{XY}, U_{Y1} \sim Bern(0.2), U_{Y2} \sim Bern(0.5), U_{Y3} \sim Bern(0.6) \end{cases} \tag{26}$$

Here, $\neg$, $\wedge$, $\vee$, and $\oplus$ indicate bitwise negation, AND, OR, and XOR operators, respectively. The board of education would like to see if they should implement a policy to switch all schools to the new textbook. Formally, they will decide to do this if

$$P(Y_{X=1} = 1) > P(Y_{X=0} = 1) \tag{27}$$

that is, if the effect of incorporating the new textbook on high GPA is higher than not incorporating it.

X	Y	$P(X, Y)$
0	0	0.64
0	1	0.16
1	0	0.04
1	1	0.16

Table 2: Observational data for Example 1.

However, the board does not have access to the true SCM $\mathcal{M}^*$. Instead, they have only collected observational data, shown in terms of probabilities in Table 2. One can verify that these are the true observational probabilities as induced by $\mathcal{M}^*$. Given this data, one may be tempted to train a model to match this observational distribution. For example, suppose the board's data science team learns the following model:

$$\widehat{M} = \begin{cases} \widehat{\mathbf{U}} & = \{\widehat{U}_X, \widehat{U}_Y\} \\ \mathbf{V} & = \{X, Y\} \\ \widehat{\mathcal{F}} & = \begin{cases} \hat{f}_X(\widehat{u}_X) & = \mathbb{1}\{\widehat{u}_X < 0.2\} \\ \hat{f}_Y(x, \widehat{u}_Y) & = \begin{cases} \mathbb{1}\{\widehat{u}_Y < 0.2\} & \text{if } x = 0 \\ \mathbb{1}\{\widehat{u}_Y < 0.8\} & \text{if } x = 1 \end{cases} \end{cases} \\ P(\widehat{\mathbf{U}}) & = \widehat{U}_X, \widehat{U}_Y \sim Unif(0, 1) \end{cases}$$

One can verify that $\widehat{M}$ induces the same $P(\mathbf{V})$ and therefore matches $\mathcal{M}^*$ in $\mathcal{L}_1$. Computing the interventional effect from this model leads to:

$$P^{\widehat{M}}(Y_{X=0} = 1) = P(\widehat{U}_Y < 0.2) = 0.2, \tag{28}$$

$$P^{\widehat{M}}(Y_{X=1} = 1) = P(\widehat{U}_Y < 0.8) = 0.8 \tag{29}$$

Since $P^{\widehat{M}}(Y_{X=1} = 1) > P^{\widehat{M}}(Y_{X=0} = 1)$, this seems to suggest that implementing the new textbook is beneficial to the students' population. However, computing the same values from $\mathcal{M}^*$ reveals that

$$P^{\mathcal{M}^*}(Y_{X=0} = 1) = 0.32, \tag{30}$$

$$P^{\widehat{M}}(Y_{X=1} = 1) = 0.16. \tag{31}$$

In other words, implementing the new textbook would actually harm students' performance on average.

The implemented model $\widehat{M}$ assigns $P(y_x) = P(y \mid x)$, failing to account for confounding between X and Y. Still, even if confounding is considered, it is unclear to what extent X and Y are confounded. In fact, even the causal direction is not known. For example, perhaps schools with better-performing students opted to use the new textbook to challenge the students, resulting in a flipped causal relationship. Without further information about the generating model, one cannot infer anything about the interventional distribution $P(y_x)$, even if the trained model matches the true model on the observational level.,

■

Example 2. Consider again the board of education situation from Example 1. Suppose, after conducting further experimental studies, the board discovers that (1) Y has no causal effect on X ($P(x_y) = P(x)$), and (2) $P(Y_{X=0} = 1) = 0.32$ and $P(Y_{X=1} = 1) = 0.16$ as discovered in the previous example. Now, all $\mathcal{L}_2$ information from $\mathcal{M}^*$ is provided as well. The board realizes that implementing the new textbook may not be the best idea and is now wondering whether schools that have already implemented the textbook and have poorly performing students would see improvement if the new textbook was no longer used. Formally, they are wondering about the counterfactual probability $P(Y_{X=0} = 1 \mid X = 1, Y = 0)$. They construct the following model:

$$\widehat{M} = \begin{cases} \widehat{\mathbf{U}} & = \{\widehat{U}_{XY}\} \\ \mathbf{V} & = \{X, Y\} \\ \widehat{\mathcal{F}} & = \begin{cases} \hat{f}_X(\widehat{u}_{XY}) & = \mathbb{1}\{\widehat{u}_{XY} < 0.2\} \\ \hat{f}_Y(x, \widehat{u}_{XY}) & = \begin{cases} 1 & \text{if } \widehat{u}_{XY} < 0.16 \\ 0 & \text{if } 0.16 \leq \widehat{u}_{XY} < 0.84 \\ \neg x & \text{otherwise} \end{cases} \end{cases} \\ P(\widehat{\mathbf{U}}) & = \widehat{U}_{XY} \sim Unif(0, 1) \end{cases} \tag{32}$$

Note that $\widehat{M}$ induces the same $\mathcal{L}_1$ and $\mathcal{L}_2$ distributions as $\mathcal{M}^*$. When querying $\widehat{M}$ with the query of interest, one obtains

$$P^{\widehat{M}}(Y_{X=0} = 1 \mid X = 1, Y = 0) = 0, \tag{33}$$

indicating that removing the book from the curriculum would have no changes on student performance. However, in reality,

$$P^{\mathcal{M}^*}(Y_{X=0} = 1 \mid X = 1, Y = 0) = 1, \tag{34}$$

meaning that removing the book would guarantee an improvement.

Once again, despite fully matching $\mathcal{L}_1$ and $\mathcal{L}_2$ of the true model, $\widehat{M}$ is not able to provide any meaningful information about an $\mathcal{L}_3$ query. ■

These examples are written in SCM notation, but equivalent NCM constructions can be found with the same distributions (following from (Xia et al., 2021, Thm. 1)). These examples illustrate that there can be many models which match on a lower layer but fail to match on a higher layer. Hence, a model (such as an NCM) trained on data from a lower layer cannot be expected to produce correct inferences from higher layers (i.e. crossing between layers is impossible). For more examples, see Examples 7-9 from Bareinboim et al. (2022). Also, Example 2 from Xia et al. (2021) shows a case where four different NCMs all match in the given observational data but have drastically differing outputs for an $\mathcal{L}_2$ query, and none of them are correct. Evidently, NCMs cannot circumvent the impossibility described by the CHT despite their expressiveness, as expressivity does not guarantee that the learned content will match that of the true generating model.

Given this impossibility, the natural course of action to proceed in problems of cross-layer inference is to study under what assumptions inferences could potentially be made. Obviously, the problem could be solved if the mechanisms and noise of the true generating SCM are known, but this is almost never the case in practice. In this work, we assume that we are given partial knowledge of the true generating SCM in the form of a causal diagram, a qualitative description of the underlying mechanisms and noise. This assumption is commonly used in the causal inference literature starting from Pearl (1995), where the field started in its modern form, and later shown in several classic causal inference applications such as in the identification of causal effects with do-calculus (Pearl, 2000, Ch. 4) or counterfactual identification and their axioms (Pearl, 2000, Ch. 7). There are compelling cases in which a data analyst may have a qualitative understanding of the causal relationships of the environment to form a causal graph (Bottou et al., 2013; Pearl, 2000). These relationships are explored in several previous counterfactual studies using symbolic approaches such as Galles & Pearl (1998), Shpitser & Pearl (2007), and Correa et al. (2021), to cite a few.

The reason why the causal diagram assumption can be utilized to solve counterfactual inference problems is due to the constraints implied by it. While many SCMs may agree on lower layer quantities but disagree on higher layer quantities, the pool of plausible models shrinks dramatically after falsifying all SCMs that do not match the given constraints. Specifically these constraints are the $\mathcal{L}_i$ equality constraints discussed in Appendix A.1, which led to the notion of $\mathcal{G}^{(\mathcal{L}_i)}$-consistency discussed in Sec. 2. Still, even with these assumptions, inferring counterfactual quantities is a non-trivial task. Despite having a constrained setting, it is still not guaranteed that a counterfactual quantity is computable from observational and interventional data, nor is it obvious how to compute such a quantity even if it is computable. This results in the counterfactual identification and estimation problems discussed in Secs. 3 and 4.

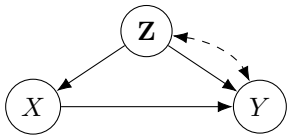


Figure 16: Graph with backdoor set $\mathbf{Z}$.

While assumptions are necessary to perform counterfactual inference, other forms of assumptions aside from the causal diagram may also lead to positive results. For example, many works in the intersection of neural models and causal inference have achieved success under the assumption of conditional ignorability (see references cited in Footnote 1). This assumption states that, under treatment X, outcome Y, and co-variates $\mathbf{Z}$, we have

$$X \perp\!\!\!\perp Y_x \mid \mathbf{Z}, \tag{35}$$

or in words, X is independent of the counterfactual variable Y_x when conditioned on $\mathbf{Z}$. This assumption is also known as back-door admissibility and is encoded in causal diagrams like Fig. 16. In this paper, we take the graphical language to validate assumptions, which may appear in different forms than Eq. 35 above.

Another possibility is to assume the availability of all $\mathcal{L}_2$ data. This data, combined with the faithfulness assumption, can be used to learn the true causal diagram (Kocaoglu et al., 2017). If this is the approach used, then the work in this paper can be applied on top of this learned causal diagram. There are examples of counterfactual queries that are still not identifiable even when given the graph and all $\mathcal{L}_2$ information (see Example 8). Still, this relies on the user having an abundance of experimental data, which is not the most common setting in practice. The work in this paper is agnostic to the approach for acquiring the causal diagram, so it can still be used whether the graph is obtained from experiments or from other means such as through expert knowledge.

Finally, although it is out of the scope of this work, one may envision ways to relax the causal diagram assumption. One example is to work with an equivalence class (EC) of causal diagrams which can be learned from purely observational ($\mathcal{L}_1$) data (Jaber et al., 2020). Performing causal inferences using the EC comes with its own merits and faces different challenges. In terms of merits, the work relies on fewer assumptions as it is entirely data-driven. However, in terms of challenges, many interventional and counterfactual quantities are not identifiable since the equivalence class is strictly weaker than the causal diagram itself (for obvious reasons). Such works are complementary to this work since the data scientist can choose for themselves whether they would prefer to make more assumptions for stronger results.

Ultimately, the goal of this work is not to promote one type of assumption over another but to offer a toolbox to solve counterfactual inference problems once the assumptions are made. The results of this paper are simply the first steps for solving counterfactual inferences in a well-understood setting. A promising direction for future work is to further relax and generalize these approaches when the diagram is not fully known.

D.2 Discussion on $\mathcal{L}_i$ Constraints

(a) True graph. (b) Alternate graph.

Figure 17: Graphs for illustrating equality constraints.

Example 3. To illustrate this concept of graphical constraints on different layers with an example, consider the graphs in Fig. 17. Suppose graph (a) is the causal diagram of the true SCM $\mathcal{M}^*$. Graph (a) implies different constraints across all three layers. For example, on layer 1, we know by d-separation that $P^*(y \mid z, x) = P^*(y \mid z)$. One could easily train a model $\mathcal{M}$ given data on $P^*(X, Y, Z)$ such that the model outputs $P^{\mathcal{M}}(X, Y, Z) = P^*(X, Y, Z)$, and hence, the equality constraint $P^{\mathcal{M}}(y \mid z, x) = P^{\mathcal{M}}(y \mid z)$ holds in $\mathcal{M}$ as well. In this case $\mathcal{M}$ would be considered $\mathcal{G}^{(\mathcal{L}_1)}$-consistent with $\mathcal{M}^*$.

However, this is not to say that $\mathcal{M}$ takes the constraints of $\mathcal{G}$ on higher layers. For example, $\mathcal{M}$ could be another SCM with causal diagram (b), which also implies the constraint $P(y \mid z, x) = P(y \mid z)$. The similarities break down on higher layer constraints. For example, on layer 2, graph (a) implies that $P^*(x_z) = P^*(x)$ (i.e. Z has no causal effect on X) and $P^*(z_x) = P^*(z \mid x)$ (i.e. the causal effect of X on Z is equal to the association). If graph (b) truly reflects the mechanisms of $\mathcal{M}$, then neither of these two constraints would hold for $\mathcal{M}$ since the causal direction is reversed. $\mathcal{M}$ would *not* be $\mathcal{G}^{(\mathcal{L}_2)}$-consistent with $\mathcal{M}^*$, greatly reducing its inferential capabilities on layer 2 quantities induced by $\mathcal{M}^*$.

Layer 1 and layer 2 constraints are formalized through the concepts of bayesian networks and causal bayesian networks, respectively. Although not yet formally studied, there are further constraints encoded by $\mathcal{G}$ corresponding to layer 3 distributions. For example, graph (a) implies that $P^*(y_z \mid z') = P^*(y_z)$ and that $P^*(y_{x'}, z_x) = P^*(y_z, z_x)$. These constraints are vital for performing inference on layer 3, which is why $\mathcal{G}^{(\mathcal{L}_3)}$-consistency is an important property.

■

D.3 PROPERTIES OF THE $\mathcal{G}$-NCM

Sec. 2 emphasizes that the $\mathcal{G}$-NCM has the properties of $\mathcal{G}^{(\mathcal{L}_3)}$-consistency and $\mathcal{L}_3$-$\mathcal{G}$ expressiveness. In this section, we expand on why these two properties are nontrivial.

The proof that the $\mathcal{G}$-NCM is $\mathcal{G}^{(\mathcal{L}_3)}$-consistent arises naturally from its structure and the fact that it is a type of SCM. We first note that any other type of structure may struggle to achieve $\mathcal{G}^{(\mathcal{L}_3)}$-consistency. To illustrate this point, consider the following example.

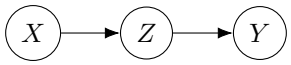


Figure 18: Graph for following examples.

Example 4. When representing a collection of distributions, perhaps the most simple model that comes to mind is an expression wrapper.

Definition 12 (Expression Wrapper)**.** Given a collection of distributions $\mathbb{P}$, an *expression wrapper* $M^{\mathbb{P}}$ defined over $\mathbb{P}$ is a collection of probability models $M_{\mathbb{P}} = \{M_P : P \in \mathbb{P}\}$, where each M_P is an $\mathcal{L}_1$-expressive model such as a normalizing flow that can express any distribution P. ■

It is obvious by construction that regardless of $\mathbb{P}$, there exists an expression wrapper defined over $\mathbb{P}$ that can express all of the distributions in $\mathbb{P}$. If we are interested in an application modeled by true SCM $\mathcal{M}^*$, and $\mathbb{P} = \mathcal{L}_3(\mathcal{M}^*)$, that implies that there exists an expression wrapper that is $\mathcal{L}_3$-consistent with $\mathcal{M}^*$. The trouble arises when we aim to enforce $G^{(\mathcal{L}_3)}$-consistency. Consider the simple 3 variable case where $\mathcal{G}$ is shown in Fig. 18. Even on layer 2, $\mathcal{G}$ encodes several constraints, including but not limited to:

1. $P(x) = P(x_z) = P(x_y) = P(x_{zy})$
2. $P(z) = P(z_y)$
3. $P(y_{xz}) = P(y_z) = P(y \mid z) = P(y \mid x, z)$
4. $P(z_{xy}) = P(z_x) = P(z \mid x)$
5. $P(y_x) = P(y \mid x)$
6. $P(\{x, z\}_y) = P(x, z)$
7. $P(\{y, z\}_x) = P(y, z \mid x)$

There are even more constraints on Layer 3, such as
$P(\{x,z\}_y) = P(x,z) = P(x,z_y) = P(x_y,z) = P(x_y,z_{y'}) = P(x_{y'},z_y) = P(x_{y'},z_{y'}) = P(x_{y'},z_{y''}) = P(x_z,z) = P(x_z,z_y) = P(x_z,z_{y'}) = P(x_{z'},z) = P(x_{z'},z_y) = P(x_{z'},z_{y'}) = P(x_{z,y},z) = P(x_{z,y},z_y) = P(x_{z,y},z_{y'}) = P(x_{z',y},z) = P(x_{z',y},z_y) = P(x_{z',y},z_{y'}) = P(x_{z,y'},z) = P(x_{z,y'},z_y) = P(x_{z,y'},z_{y'}) = P(x_{z,y'},z_{y''}) = P(x_{z',y'},z) = P(x_{z',y'},z_y) = P(x_{z',y'},z_{y'}) = P(x_{z',y'},z_{y''}) = P(x,z_x) = P(x,z_{x,y}) = P(x,z_{x,y'}) = P(x_y,z_x) = P(x_y,z_{x,y}) = P(x_y,z_{x,y'}) = P(x_{y'},z_x) = P(x_{y'},z_{x,y}) = P(x_{y'},z_{x,y'}) = P(x_{y'},z_{x,y''}) = P(x_z,z_x) = P(x_z,z_{x,y}) = P(x_z,z_{x,y'}) = P(x_{z'},z_x) = P(x_{z'},z_{x,y}) = P(x_{z'},z_{x,y'}) = P(x_{z,y},z_x) = P(x_{z,y},z_{x,y}) = P(x_{z,y},z_{x,y'}) = P(x_{z',y},z_x) = P(x_{z',y},z_{x,y}) = P(x_{z',y},z_{x,y'}) = P(x_{z,y'},z_x) = P(x_{z,y'},z_{x,y}) = P(x_{z,y'},z_{x,y'}) = P(x_{z,y'},z_{x,y''}) = P(x_{z',y'},z_x) = P(x_{z',y'},z_{x,y}) = P(x_{z',y'},z_{x,y'}) = P(x_{z',y'},z_{x,y''})$,
among several others. Furthermore, each of these constraints must hold for every value of x, y, z, x', y', z', etc., in their respective domains.

The sheer number of constraints even in a 3-variable case is so large that it is difficult to list them all, let alone train a model to actually enforce all of these constraints. The causal diagram parsimoniously encodes all of these constraints implicitly, and therefore, the NCM enforces these constraints by construction when it is used as an inductive bias. However, these constraints must be enforced manually when using a model such as the expression wrapper. Not only must every distribution $P \in \mathbb{P}$ be trained, which is already a difficult task due to the size of $\mathbb{P}$, all of these exponentially

many constraints must be simultaneously enforced if $\mathcal{G}^{(\mathcal{L}_3)}$-consistency is desired. This is simply not feasible. ■

There are also models that achieve $\mathcal{G}^{(\mathcal{L}_2)}$-consistency but not $\mathcal{G}^{(\mathcal{L}_3)}$-consistency. Consider the following example.

Example 5. Recall the definition of Causal Bayesian networks.

Definition 13 (Causal Bayesian Network (CBN) (Bareinboim et al., 2022, 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 $V_i \in \mathbf{V} \setminus \mathbf{X}$, $\mathbf{W} \subseteq \mathbf{V} \setminus (\mathbf{Pa}_i^{\mathbf{X}+} \cup \mathbf{X} \cup \{V_i\})$:

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

,

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

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

Here, $\mathbf{Pa}_{V_i}^{\mathbf{x}+} = \mathbf{Pa}(\{V \in \mathbf{C}_{\overline{\mathbf{X}}}(V_i) : V \leq V_i\})$, 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. ■

To clarify notation used in this definition, $P(\mathbf{y} \mid do(\mathbf{x}), \mathbf{z}) = P(\mathbf{y}_{\mathbf{x}} \mid \mathbf{z}_{\mathbf{x}})$ for any $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, and $\mathcal{G}_{\overline{\mathbf{X}}}$ refers to the graph $\mathcal{G}$ with the edges into $\mathbf{X}$ removed.

In words, a causal diagram $\mathcal{G}$ is a CBN for a collection of distributions $\mathbf{P}^*$ if $\mathbf{P}^*$ satisfies the $\mathcal{L}_2$ equality constraints induced by $\mathcal{G}$. The CBN explicitly states these constraints, and more details can be found in Bareinboim et al. (2022).

Let $\mathcal{G}$ be some causal diagram induced by true SCM $\mathcal{M}^*$. Let $M_{\mathsf{CBN}} = \langle \mathcal{G}, \mathbf{P}^* \rangle$ be some CBN defined over distributions $\mathbf{P}^*$. Then, by definition, M_{CBN} is $\mathcal{G}^{(\mathcal{L}_2)}$-consistent with $\mathcal{M}^*$. However, M_{CBN} would not be $\mathcal{G}^{(\mathcal{L}_3)}$-consistent with $\mathcal{M}^*$ since the constraints in the definition do not include any distributions outside of layer 2. In fact, $\mathbf{P}^*$ does not even have to be defined with layer 3 distributions.

As an example, suppose $\mathcal{G}$ is the graph in Fig. 18, and that $\mathcal{L}_1(\mathcal{M}^*) = \mathcal{L}_1(M_{\mathsf{CBN}})$, that is, $\mathbf{P}^*$ matches the layer 1 distribution of $\mathcal{M}^*$. Then, since M_{CBN} is $\mathcal{G}^{(\mathcal{L}_2)}$-consistent, one can derive $P(y_x) = P(y \mid x)$ and conclude that $P^{M_{\mathsf{CBN}}}(y_x) = P^{\mathcal{M}^*}(y_x)$, successfully identifying a layer 2 quantity. However, if a model were $\mathcal{G}^{(\mathcal{L}_3)}$-consistent with $\mathcal{M}^*$, one could derive quantities such as $P(y_x, z_x) = P(y_{xz}, z_x) = P(y_z, z_x) = P(y_z)P(z_x) = P(y \mid z)P(z \mid x)$, also computable from observational quantities. This is not guaranteed to be true with the CBN, since counterfactual quantities like $P(y_z, z_x)$ are not guaranteed to be defined in $\mathbf{P}^*$, and even if they were, they could take any values since no part of the definition of CBNs constrains them. ■

There also exist models that may be $\mathcal{G}^{(\mathcal{L}_3)}$-consistent but are not $\mathcal{L}_3$-$\mathcal{G}$ expressive. Any model class in which Markovianity is enforced is an example of such a case, as shown in the following.

(a) True graph. (b) Markovian subgraph.

Figure 19: Graphs for Example 6.

Example 6. Given the causal diagram $\mathcal{G}$ from the true SCM $\mathcal{M}^*$, consider $\widetilde{\Omega}(\mathcal{G})$, the set of all Markovian SCMs that induce causal diagrams with a subset of the edges of $\mathcal{G}$. By construction, $\widetilde{\Omega}(\mathcal{G})$ is $\mathcal{G}^{(\mathcal{L}_3)}$-consistent with $\mathcal{M}^*$ since no constraint can be removed by removing edges, so any SCM in $\widetilde{\Omega}(\mathcal{G})$ has at least as many $\mathcal{L}_3$ constraints as those induced by $\mathcal{G}$.

Suppose $\mathcal{G}$ is the bow graph as shown in Fig. 19, and the true SCM $\mathcal{M}^*$ is defined as

$$\mathcal{M}^* = \begin{cases} \mathbf{U} & = \{U_{XY}, U_Y\}, \text{ both binary} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F} & = \begin{cases} f_X(u_{XY}) & = u_{XY} \\ f_Y(x, u_{XY}, u_Y) & = u_{XY} \oplus u_Y \end{cases} \\ P(\mathbf{U}) & \text{defined such that } U_{XY} \sim \text{Bernoulli}(0.5), U_Y \sim \text{Bernoulli}(0.1) \end{cases}$$

Evidently, X and Y are highly correlated, as $P(Y = 1 \mid X = 1) = 0.9$, but $P(Y = 1) = 0.5$. However, this correlation is not causal, as the input X remains unused by f_Y, implying that $P(Y_{X=1} = 1) = P(Y = 1) = 0.5$. It is clear that no model in $\widetilde{\Omega}(\mathcal{G})$ will be able to represent $\mathcal{M}^*$ on all three layers. For example, one may construct $\mathcal{M}' \in \widetilde{\Omega}(\mathcal{G})$ as follows

$$\mathcal{M}' = \begin{cases} \mathbf{U}' & = \{U_X, U_Y\}, \text{ both binary} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F}' & = \begin{cases} f'_X(u_X) & = u_X \\ f'_Y(x, u_Y) & = x \oplus u_Y \end{cases} \\ P(\mathbf{U}') & \text{defined such that } U_X \sim \text{Bernoulli}(0.5), U_Y \sim \text{Bernoulli}(0.1) \end{cases}$$

This model induces the graph $\mathcal{G}'$ as shown in Fig. 19b, so it is a valid model from $\widetilde{\Omega}(\mathcal{G})$. Moreover, it can even reproduce $\mathcal{L}_1$ quantities from $\mathcal{M}^*$, as it is clear that $P^{\mathcal{M}'}(Y = 1 \mid X = 1) = 0.9$, and $P^{\mathcal{M}'}(Y = 1) = 0.5$. This makes $\mathcal{M}'$ $\mathcal{L}_1$-consistent with $\mathcal{M}^*$. However, on layer 2, the similarities break down. While in the true model, X has no effect on Y, in this case, the Markovian nature of $\mathcal{M}'$ means that $P^{\mathcal{M}'}(Y_{X=1} = 1) = P^{\mathcal{M}'}(Y = 1 \mid X = 1) = 0.9$, resulting in a different value.

Clearly $\mathcal{M}'$ is not $\mathcal{L}_2$-consistent with $\mathcal{M}^*$, let alone $\mathcal{L}_3$-consistent. However, no other model from $\widetilde{\Omega}(\mathcal{G})$ would be able to achieve $\mathcal{L}_2$-consistency either. In a Markovian model, the correlation between X and Y must be modeled through a causal effect from either X to Y or from Y to X, and both of these options would violate $\mathcal{L}_2$-consistency with $\mathcal{M}^*$. ■

Other examples of model classes that are $\mathcal{G}^{(\mathcal{L}_3)}$-consistent but not $\mathcal{L}_3$-$\mathcal{G}$ expressive include parametric SCMs in which the parameterization is not fully expressive, such as linear SCMs.

As shown in Thm. 2, the class of $\mathcal{G}$-NCMs is $\mathcal{L}_3$-$\mathcal{G}$ expressive. It may not be immediately clear why this expressivity holds on all three layers instead of only on layers 1 or 2. The intuition is that with the expressivity of neural networks, a $\mathcal{G}$-NCM can directly model the functions of the true SCM $\mathcal{M}^*$ instead of indirectly matching its distributions. By matching the functions and utilizing the probability integral transform on uniform random variables to match $P(\mathbf{U})$, it is guaranteed that one can construct a $\mathcal{G}$-NCM to directly mimic the behavior of $\mathcal{M}^*$ on the functional level, implying equivalence on all three layers. Consider the following example.

Figure 20: Graph for Example 7.

Example 7. Suppose we are studying the effects of a drug (X) on the recovery of a disease (Y). For simplicity, X and Y are both binary, $X = 1$ represents the drug being taken, and $Y = 1$ represents a full recovery. There are two hypothesized models. In $\mathcal{M}_1$, taking the drug helps people without a certain gene ($U_Y = 1$) but actively harms people with the gene ($U_Y = 0$). In $\mathcal{M}_2$, the drug has no

effect. The two SCMs are clarified as follows:

$$\mathcal{M}_1 = \begin{cases} \mathbf{U}_1 & = \{U_X, U_Y\}, \text{ both binary} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F}_1 & = \begin{cases} f_X^1(u_X) & = u_X \\ f_Y^1(x, u_Y) & = x \oplus u_Y \end{cases} \\ P(\mathbf{U}_1) & \text{defined such that } U_X, U_Y \sim \text{Bernoulli}(0.5) \end{cases}$$

$$\mathcal{M}_2 = \begin{cases} \mathbf{U}_2 & = \{U_X, U_Y\}, \text{ both binary} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F}_2 & = \begin{cases} f_X^2(u_X) & = u_X \\ f_Y^2(x, u_Y) & = u_Y \end{cases} \\ P(\mathbf{U}_2) & \text{defined such that } U_X, U_Y \sim \text{Bernoulli}(0.5) \end{cases}$$

It turns out that these two hypothesized models match in both $\mathcal{L}_1$ and $\mathcal{L}_2$. For example, $P^{\mathcal{M}_1}(Y_{X=1} = 1) = P^{\mathcal{M}_2}(Y_{X=1} = 1) = 0.5$. The difference lies in their induced distributions for $\mathcal{L}_3$. For example, $P^{\mathcal{M}_1}(Y_{X=1} = 1 \mid X = 0, Y = 1) = 0$. In words, in $\mathcal{M}_1$, if it is observed that someone did not take the drug and recovered from the disease, then it is guaranteed that they would fail to recover if they had taken the drug. On the other hand, $P^{\mathcal{M}_2}(Y_{X=1} = 1 \mid X = 0, Y = 1) = P^{\mathcal{M}_2}(Y = 1 \mid Y = 1) = 1$ since X simply has no effect on the disease.

The reasoning for this difference at the functional level is that both f_Y^1 and f_Y^2 have two different behaviors with respect to X depending on the setting of U_Y. When $U_Y = 0$, $f_Y^1(x) = x$, while $f_Y^2(x) = 0$. When $U_Y = 1$, $f_Y^1(x) = 1 - x$, while $f_Y^2(x) = 1$. In both cases, the functions are different between $\mathcal{M}_1$ and $\mathcal{M}_2$. Notably, the output of f_Y^1 depends on X while the output of f_Y^2 does not.

We can reproduce this difference in behavior when constructing $\mathcal{G}$-NCMs to represent these SCMs.

Recall the binary step activation function,

$$\sigma(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0. \end{cases} \tag{36}$$

Using this activation function, we can create various useful neural network functions like the following:

$$\begin{aligned} \hat{f}_{OR}(z_1, \ldots, z_n) &= \sigma\left(\sum_{i=1}^{n} z_i - 1\right) && \text{bitwise OR} \\ \hat{f}_{AND}(z_1, \ldots, z_n) &= \sigma\left(\sum_{i=1}^{n} z_i - n\right) && \text{bitwise AND} \\ \hat{f}_{NOT}(z) &= \sigma(-z) && \text{bitwise NOT} \\ \hat{f}_{\geq p}(z) &= \sigma(z - p) && 1 \text{ if } z \geq p \\ \hat{f}_{< p}(z) &= \hat{f}_{NOT}(\hat{f}_{\geq p}(z)) && 1 \text{ if } z < p \end{aligned}$$

With these functions, consider the following $\mathcal{G}$-NCMs, $\widehat{M}_1$ and $\widehat{M}_2$ that replicate the behavior of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively.

$$\widehat{M}_1 = \begin{cases} \widehat{\mathbf{U}}_1 & = \{\widehat{U}_X, \widehat{U}_Y\} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \widehat{\mathcal{F}}_1 & = \begin{cases} \hat{f}_X^1(\widehat{u}_X) & = \hat{f}_{<0.5}(\widehat{u}_X) \\ \hat{f}_Y^1(x, \widehat{u}_Y) & = \hat{f}_{OR} \begin{cases} \hat{f}_{AND}(x, \hat{f}_{NOT}(\hat{f}_{<0.5}(\widehat{u}_Y))) \\ \hat{f}_{AND}(\hat{f}_{NOT}(x), \hat{f}_{<0.5}(\widehat{u}_Y)) \end{cases} \end{cases} \\ P(\widehat{\mathbf{U}}_1) & \text{defined such that } \widehat{U}_X, \widehat{U}_Y \sim \text{Unif}(0, 1) \end{cases}$$

$$\widehat{M}_2 = \begin{cases} \widehat{\mathbf{U}}_2 & = \{\widehat{U}_X, \widehat{U}_Y\} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F}_2 & = \begin{cases} \hat{f}^2_X(\widehat{u}_X) & = \hat{f}_{<0.5}(\widehat{u}_X) \\ \hat{f}^2_Y(x, \widehat{u}_Y) & = \hat{f}_{<0.5}(\widehat{u}_Y) \end{cases} \\ P(\widehat{\mathbf{U}}_2) & \text{defined such that } \widehat{U}_X, \widehat{U}_Y \sim \text{Unif}(0,1) \end{cases}$$

Note that $\widehat{M}_1$ and $\widehat{M}_2$ match $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively, on all three layers. For example, $P^{\widehat{M}_1}(Y_{X=1} = 1 \mid X = 0, Y = 1) = 0$ and $P^{\widehat{M}_2}(Y_{X=1} = 1 \mid X = 0, Y = 1) = 1$. This is because the construction of $\widehat{M}_1$ and $\widehat{M}_2$ were designed to directly match the functions of $\mathcal{M}_1$ and $\mathcal{M}_2$. ■

As a disclaimer, note that $\mathcal{L}_3$-$\mathcal{G}$ expressiveness means the existence of a model that matches on all three layers, but it does not necessarily mean that such a model can be found. In the case of Example 7, we could create $\mathcal{G}$-NCMs to match the hypothesized SCMs because the full specifications of the SCMs were provided. In practice, this is rarely the case, and it may not be possible to determine the correct NCM from training on data alone, which is why the identification problem is important to determine when inferences can be made from a trained NCM.

Determining whether a counterfactual quantity is identifiable from the given data and constraints is an important problem even in the Markovian case. Consider the following simple example.

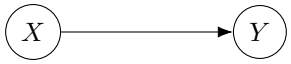


Figure 21: Graph for Example 8.

Example 8. Consider a simple study on the effect of a new drug (X) on survival rate of a sick patient (Y), modeled by the following SCM $\mathcal{M}$:

$$\mathcal{M}^* = \begin{cases} \mathbf{U} & = \{U_X, U_{Y1}, U_{Y2}\}, \text{ all binary} \\ \mathbf{V} & = \{X, Y\}, \text{ both binary} \\ \mathcal{F} & = \begin{cases} f_X(u_X) & = u_X \\ f_Y(x, u_{Y1}, u_{Y2}) & = \begin{cases} x \oplus u_{Y1} & u_{Y2} = 0 \\ u_{Y1} & u_{Y2} = 1 \end{cases} \end{cases} \\ P(\mathbf{U}) & \text{defined such that } U_X, U_{Y1}, U_{Y2} \sim \text{Bernoulli}(0.5) \end{cases}$$

This SCM induces the graph $\mathcal{G}$ in Fig. 21. Evidently from observing $\mathcal{M}^*$, this drug has some effect on patients with this disease. However, $\mathcal{M}^*$ is unobserved, and we only have data. Suppose we have both observational data and experimental data from testing on X. Experiments on Y are not needed since Y has no causal effect on X, so this essentially means we have data on all of layer 1 and layer 2. This data is shown in terms of probabilities in Table. 3.

X	Y	$P(X, Y)$
0	0	0.25
0	1	0.25
1	0	0.25
1	1	0.25

x	$Y_{X=x}$	$P(Y_{X=x})$
0	0	0.5
0	1	0.5
1	0	0.5
1	1	0.5

Table 3: Data for Example 8

The data seems to suggest that the drug has no effect on the disease since $P(y_x) = P(y \mid x) = P(y)$ for all values of y and x. However, just to be sure, we are interested in evaluating the query $P(Y_{X=1} = 1 \mid X = 0, Y = 0)$, the probability that someone who did not take the drug and did not recover *would have* recovered had they taken the drug. This quantity is called the probability of sufficiency (Pearl, 2000, Def. 9.2.2).

Running the procedure in Alg. 1 on $\mathcal{G}$ and $\mathbb{Z} = \{P(\mathbf{V}), P(\mathbf{V}_x)\}$, we may achieve the following two $\mathcal{G}$-NCMs (with the notation from Example 7)

$$\widehat{M}_1 = \begin{cases} \widehat{\mathbf{U}} & = \{\widehat{U}_X, \widehat{U}_Y\} \\ \mathbf{V} & = \{X, Y\} \\ \widehat{\mathcal{F}}_1 & = \begin{cases} \hat{f}_X^{(1)}(\widehat{u}_X) & = \hat{f}_{<0.5}(\widehat{u}_X) \\ \hat{f}_Y^{(1)}(x, \widehat{u}_Y) & = \hat{f}_{<0.5}(\widehat{u}_Y) \end{cases} \\ P(\widehat{\mathbf{U}}) & \text{defined such that } \widehat{U}_X, \widehat{U}_Y \sim \text{Unif}(0,1) \end{cases}$$

$$\widehat{M}_2 = \begin{cases} \widehat{\mathbf{U}} & = \{\widehat{U}_X, \widehat{U}_Y\} \\ \mathbf{V} & = \{X, Y\} \\ \widehat{\mathcal{F}}_2 & = \begin{cases} \hat{f}_X^{(2)}(\widehat{u}_X) & = \hat{f}_{<0.5}(\widehat{u}_X) \\ \hat{f}_Y^{(2)}(x, \widehat{u}_Y) & = \hat{f}_{OR}(\hat{f}_{AND}(x, \hat{f}_{<0.5}(\widehat{u}_Y)), \hat{f}_{AND}(\hat{f}_{NOT}(x), \hat{f}_{\geq 0.5}(\widehat{u}_Y))) \end{cases} \\ P(\widehat{\mathbf{U}}) & \text{defined such that } \widehat{U}_X, \widehat{U}_Y \sim \text{Unif}(0,1) \end{cases}$$

Note that $\hat{f}_Y^{(2)}$ simply resembles the XOR function. Evaluating $P^{\widehat{M}_1}(X, Y)$, $P^{\widehat{M}_2}(X, Y)$, $P^{\widehat{M}_1}(Y_x)$, and $P^{\widehat{M}_2}(Y_x)$, one can verify that they match the results in Table 3. However, for our query, observe that

$$\begin{aligned} P^{\widehat{M}_1}(Y_{X=1} = 1 \mid X = 0, Y = 0) &= \frac{P^{\widehat{M}_1}(Y_{X=1} = 1, X = 0, Y = 0)}{P^{\widehat{M}_1}(X = 0, Y = 0)} \\ &= \frac{P(\widehat{U}_Y < 0.5, \widehat{U}_X \geq 0.5, \widehat{U}_Y \geq 0.5)}{P(\widehat{U}_X \geq 0.5, \widehat{U}_Y \geq 0.5)} \\ &= \frac{0}{0.25} = 0, \end{aligned}$$

and

$$\begin{aligned} P^{\widehat{M}_2}(Y_{X=1} = 1 \mid X = 0, Y = 0) &= \frac{P^{\widehat{M}_2}(Y_{X=1} = 1, X = 0, Y = 0)}{P^{\widehat{M}_2}(X = 0, Y = 0)} \\ &= \frac{P(\widehat{U}_Y < 0.5, \widehat{U}_X \geq 0.5, \widehat{U}_Y < 0.5)}{P(\widehat{U}_X \geq 0.5, \widehat{U}_Y < 0.5)} \\ &= \frac{0.25}{0.25} = 1. \end{aligned}$$

In words, according to $\widehat{M}_1$, someone who did not take the drug or recover is guaranteed *not* to recover even if they had taken the drug. On the other hand, according to $\widehat{M}_2$, the opposite is true, and that person would be guaranteed to recover if they had taken the drug. Remarkably, despite the fact that we have the graph $\mathcal{G}$ and *all* data from $\mathcal{L}_1$ and $\mathcal{L}_2$, we cannot narrow down the value of our query at all, as it can take any possible value between 0 and 1.

Moreover, when we evaluate the true result of the query from $\mathcal{M}^*$, we get

$$\begin{aligned} P^{\mathcal{M}^*}(Y_{X=1} = 1 \mid X = 0, Y = 0) &= \frac{P^{\mathcal{M}^*}(Y_{X=1} = 1, X = 0, Y = 0)}{P^{\mathcal{M}^*}(X = 0, Y = 0)} \\ &= \frac{P(U_X = 0, U_{Y1} = 0, U_{Y2} = 0)}{P(U_X = 0, U_{Y1} = 0)} \\ &= \frac{0.125}{0.25} = \frac{1}{2}. \end{aligned}$$

In other words, neither $\widehat{M}_1$ nor $\widehat{M}_2$ captured the true value of the query. This query is not identifiable from $\mathbb{Z}$ and $\mathcal{G}$. Arbitrarily training a model to fit $\mathbb{Z}$ and $\mathcal{G}$ is almost guaranteed to fail to capture the true query in this case.

■

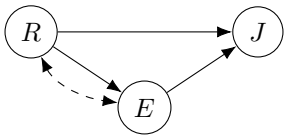


Figure 22: Graph for Example 9.

Although Example 8 shows a compelling case where one would not want to estimate the query due to non-identifiability, the NCM serves as a robust proxy model in cases where the query is identifiable.

Example 9. Suppose we are conducting a study on whether race (R) affects an applicant's chances of getting a certain job (J). We also collect data on the applicant's education background (E). For simplicity, suppose all three are binary variables, $R = 1$ implies that the applicant is from a protected class (e.g. black, hispanic), $E = 1$ implies that the applicant graduated from a prestigious university, and $J = 1$ implies that the applicant got the job. Experts suggest that race may affect application results both directly and indirectly through education background, but race and education may share confounding factors such as location, as shown in Fig. 22. The full relationship between variables is modeled through SCM $\mathcal{M}^*$:

$$\mathcal{M}^* = \begin{cases} \mathbf{U} & = \{U_{RE}, U_E, U_J\}, \text{ all binary} \\ \mathbf{V} & = \{R, E, J\} \\ \mathcal{F} & = \begin{cases} f_R(u_{RE}) & = u_{RE} \\ f_E(r, u_{RE}, u_E) & = (\neg u_{RE} \wedge \neg r) \oplus u_E \\ f_J(r, e, u_J) & = (\neg r \vee e) \oplus u_J \end{cases} \\ P(\mathbf{U}) & \text{ defined such that } U_{RE} \sim \text{Bernoulli}(0.5), U_E, U_J \sim \text{Bernoulli}(0.25) \end{cases}$$

Relevant quantities and their corresponding probability for each setting of $\mathbf{U}$ is shown in Table 4.

$P(\mathbf{U})$	U_{RE}	U_E	U_J	R	E	J	$E_{R=0}$	$E_{R=1}$	$J_{R=0}$	$J_{R=1}$	$J_{R=1,E_{R=0}}$
$p_0 = 9/32$	0	0	0	0	1	1	1	0	1	0	1
$p_1 = 3/32$	0	0	1	0	1	0	1	0	0	1	0
$p_2 = 3/32$	0	1	0	0	0	1	0	1	1	1	0
$p_3 = 1/32$	0	1	1	0	0	0	0	1	0	0	1
$p_4 = 9/32$	1	0	0	1	0	0	0	0	1	0	0
$p_5 = 3/32$	1	0	1	1	0	1	0	0	0	1	1
$p_6 = 3/32$	1	1	0	1	1	1	1	1	1	1	1
$p_7 = 1/32$	1	1	1	1	1	0	1	1	0	0	0

Table 4: Relevant quantities calculated for each setting of $\mathbf{U}$ in $\mathcal{M}^*$ for Example 9.

We are given the causal diagram $\mathcal{G}$ from Fig. 22 as well as the datasets $\mathbb{Z} = \{P(\mathbf{V}), P(\mathbf{V}_{R=0}), P(\mathbf{V}_{R=1})\}$. The observational data from $P(\mathbf{V})$ was collected by simply asking random participants to list their race, education level, and application results. The experimental data from $P(\mathbf{V}_{R=0})$ and $P(\mathbf{V}_{R=1})$ is obtained similarly, except participants are asked to write their race as $R = 0$ or $R = 1$ respectively in their school and job applications regardless of their actual race. This data, computed from Table 4, is shown in Table 5.

When looking at the average treatment effect of race on job success from the data, we see that

$$P(J_{R=1} = 1) - P(J_{R=0} = 1) = \frac{12}{32} - \frac{24}{32} = -\frac{12}{32} = -0.375$$

which is a rather large disparity between the success rates of protected classes compared to others. To avoid accusations of discrimination, the company claims that this disparity is not a result of the company's hiring practices. The company says that they only look at education levels, and the reason why protected classes are less likely to get the job is because the fault lies with the education system for discriminating against protected classes in their admissions process. To verify this claim, we may be interested in the *natural direct effect* query, formally defined in this context as

$$Q = P(J_{R=1,E_{R=0}} = 1) - P(J_{R=0} = 1). \tag{37}$$

R	E	J	$P(\mathbf{V})$
0	0	0	$p_3 = 1/32$
0	0	1	$p_2 = 3/32$
0	1	0	$p_1 = 3/32$
0	1	1	$p_0 = 9/32$
1	0	0	$p_4 = 9/32$
1	0	1	$p_5 = 3/32$
1	1	0	$p_7 = 1/32$
1	1	1	$p_6 = 3/32$

r	$E_{R=r}$	$J_{R=r}$	$P(\mathbf{V}_{R=r})$
0	0	0	$p_3 + p_5 = 4/32$
0	0	1	$p_2 + p_4 = 12/32$
0	1	0	$p_1 + p_7 = 4/32$
0	1	1	$p_0 + p_6 = 12/32$
1	0	0	$p_0 + p_4 = 18/32$
1	0	1	$p_1 + p_5 = 6/32$
1	1	0	$p_3 + p_7 = 2/32$
1	1	1	$p_2 + p_6 = 6/32$

Table 5: Data for Example 9

In words, the first term is the probability that someone will get the job if we change their race to a protected class, but change their education level as if their race was not from a protected class. The second term is simply their natural chance of getting the job if their race was fixed to a non-protected class. In other words, this query calculates the direct effect of race on the hiring process, irrespective of education. We do not have data on the first term of Q, but it turns out that Q is identifiable from $\mathcal{G}$ and $\mathbb{Z}$. Running Alg. 1 would return two NCMs that induce the same result for Q. For example, the first term can be derived symbolically as

$$
\begin{aligned}
&P(J_{R=1,E_{R=0}} = 1) \\
&= P(J_{R=1,E=0} = 1, E_{R=0} = 0) + P(J_{R=1,E=1} = 1, E_{R=0} = 1) && \text{Unnesting} \\
&= P(J_{R=1,E=0} = 1)P(E_{R=0} = 0) + P(J_{R=1,E=1} = 1)P(E_{R=0} = 1) && \text{C-factor decomp.} \\
&= P(J = 1 \mid R = 1, E = 0)P(E_{R=0} = 0) && \text{Rule 2} \\
&+ P(J = 1 \mid R = 1, E = 1)P(E_{R=0} = 1) \\
&= \left(\frac{3}{12}\right)\left(\frac{16}{32}\right) + \left(\frac{3}{4}\right)\left(\frac{16}{32}\right) = \frac{1}{8} + \frac{3}{8} = \frac{1}{2} && \text{From data}
\end{aligned}
$$

From the data, we also have $P(J_{R=0} = 1) = \frac{24}{32} = \frac{3}{4}$. Hence, we can compute

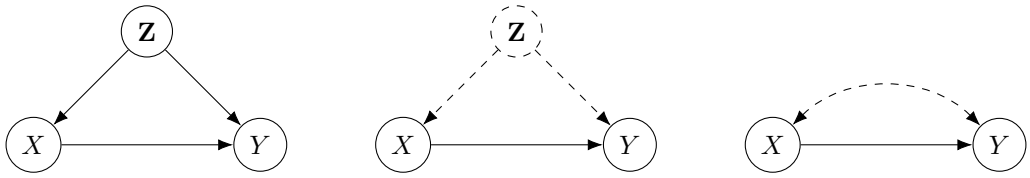
$$
Q = P(J_{R=1,E_{R=0}} = 1) - P(J_{R=0} = 1) = \frac{1}{2} - \frac{3}{4} = -\frac{1}{4} = -0.25,
$$

revealing that while the direct effect is slightly lower than the average treatment effect, the company still shares some responsibility in the discrimination of protected classes in their hiring practices. Indeed, one could construct a $\mathcal{G}$-NCM that matches $\mathcal{M}^*$ on the given data sets, for example:

$$
\widehat{M} = \begin{cases}
\widehat{\mathbf{U}} & = \{\widehat{U}_{RE}, \widehat{U}_J\} \\
\mathbf{V} & = \{R, E, J\} \\
\widehat{\mathcal{F}} & = \begin{cases}
\hat{f}_R(\widehat{u}_{RE}) & = \hat{f}_{\geq 0.5}(\widehat{u}_{RE}) \\
\hat{f}_E(r, \widehat{u}_{RE}) & = \hat{f}_{OR} \begin{cases} \hat{f}_{AND}\left(\hat{f}_{<0.375}(\widehat{u}_{RE}), \hat{f}_{NOT}(r)\right) \\ \hat{f}_{AND}\left(\hat{f}_{<0.125}(\widehat{u}_{RE}), r\right) \\ \hat{f}_{\geq 0.875}(\widehat{u}_{RE}) \end{cases} \\
\hat{f}_J(r, e, \widehat{u}_J) & = \hat{f}_{OR} \begin{cases} \hat{f}_{AND}\left(\hat{f}_{NOT}(r), \hat{f}_{<0.75}(\widehat{u}_J)\right) \\ \hat{f}_{AND}\left(r, \hat{f}_{NOT}(e), \hat{f}_{<0.25}(\widehat{u}_J)\right) \\ \hat{f}_{AND}\left(r, e, \hat{f}_{<0.75}(\widehat{u}_J)\right) \end{cases}
\end{cases} \\
P(\widehat{\mathbf{U}}) & \text{ defined such that } \widehat{U}_{RE}, \widehat{U}_J \sim \text{Unif}(0, 1)
\end{cases}
$$

One can verify that the quantities induced by $\widehat{M}$ match those in Table 5. Interestingly, the functions of $\widehat{M}$ are quite different from those of $\mathcal{M}^*$, and in fact, they will likely not match in non-identifiable quantities. However, if we evaluate $\widehat{M}$ on our query of interest Q, we see that

1. $E_{R=0} = 1$ when $\widehat{U}_{RE} < 0.375$ or $\widehat{U}_{RE} \geq 0.875$, 0 otherwise;



(a) Example of a graph from a Markovian SCM. Note the lack of bidirected edges. Z is an observed confounder between X and Y.

(b) If Z is unobserved, then Markovianity no longer holds because Z becomes an unobserved confounder between X and Y.

(c) In non-Markovian settings, unobserved confounding between two observed variables is denoted as a bidirected edge.

Figure 23: Graphs for Markovianity discussion.

2. When $E_{R=0} = 0$, $J_{R=1,E_{R=0}} = 1$ when $\widehat{U}_J < 0.25$, 0 otherwise;
3. When $E_{R=0} = 1$, $J_{R=1,E_{R=0}} = 1$ when $\widehat{U}_J < 0.75$, 0 otherwise.

Hence, we can compute

$$
\begin{aligned}
&P^{\widehat{M}}(J_{R=1,E_{R=0}} = 1) \\
&= P(0.375 \leq \widehat{U}_{RE} < 0.875)P(\widehat{U}_J < 0.25) + P(\widehat{U}_{RE} < 0.375 \vee \widehat{U}_{RE} \geq 0.875)P(\widehat{U}_J < 0.75) \\
&= (0.5)(0.25) + (0.5)(0.75) = 0.5,
\end{aligned}
$$

which matches the symbolic result. Additionally, we see that $P^{\widehat{M}}(J_{R=0} = 1) = P(\widehat{U}_J < 0.75) = 0.75$, so we see that the result induced by $\widehat{M}$ for Q is indeed $0.5 - 0.75 = -0.25$.

Computing the true value of the query from $\mathcal{M}^*$ using values from Table 4, we see that

$$
\begin{aligned}
&P^{\mathcal{M}^*}(J_{R=1,E_{R=0}} = 1) - P^{\mathcal{M}^*}(J_{R=0} = 1) \\
&= p_0 + p_3 + p_5 + p_6 - p_0 - p_2 - p_4 - p_6 = p_3 + p_5 - p_2 - p_4 \\
&= \frac{1}{32} + \frac{3}{32} - \frac{3}{32} - \frac{9}{32} = -\frac{8}{32} = -0.25.
\end{aligned}
$$

Remarkably, by simply fitting the $\mathcal{G}$-NCM on the data and querying it as if it were $\mathcal{M}^*$ itself, we achieve the correct value of Q matching both the true value from $\mathcal{M}^*$ and the result from the symbolic derivation. As long as the training succeeds in matching the data, it does not matter what functions are learned, and the evaluation requires no knowledge of the symbolic approach, demonstrating the strength of using the NCM for evaluating identifiable queries. ■

D.4 Discussion on Markovianity

One of the major contributions emphasized in this work is the generality of the approach. In particular, Alg. 1 solves the counterfactual identification and estimation problems for any counterfactual query with any given input in any causal graph setting. Notably, we do not make the *Markovianity* assumption about the generating SCM. Markovianity restricts the set of unobserved variables $\mathbf{U}$, requiring them to be independent and having each $U \in \mathbf{U}$ as an input to at most one function in $\mathcal{F}$. This translates to the lack of bidirected edges in any causal diagram generated by a Markovian SCM (Fig. 23a, for example). In other words, since no unobserved variable can be a common cause of two observed variables, Markovianity encodes the fact that there is no unobserved confounding.

Markovianity is frequently assumed in many works (including several related works in Appendix C) because it drastically simplifies the problem. When attempting to infer $\mathcal{L}_2$ queries from $\mathcal{L}_1$ data and the causal diagram $\mathcal{G}$, it turns out that any interventional query $P(\mathbf{y}_{\mathbf{x}})$ can trivially be identified as

$$
P(\mathbf{y}_{\mathbf{x}}) = \sum_{\mathbf{z}} P(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) P(\mathbf{z}), \tag{38}
$$

where $\mathbf{Z}$ is the set of non-descendants of $\mathbf{X}$; for details, see (Bareinboim et al., 2022, Corol. 2).

Several observations follow from this observation. First, this implies that works that attempt to infer $\mathcal{L}_2$ queries from $\mathcal{L}_1$ data under the Markovianity assumption do not even need to consider the identification problem, as they are all identifiable given the graph. Kocaoglu et al. (2018) is an example of a work that can directly estimate the query without having to be concerned about the identification problem due to the Markovianity assumption.

Second, when attempting to estimate the $\mathcal{L}_2$ query, the estimand in Eq. 38 is the same for every query, so only one form of the query needs to be considered when designing an estimator. For example, although not explicitly assuming Markovianity, the works in Footnote 1 assume conditional ignorability, which implies the lack of unobserved confounding between the treatment X and the outcome Y. This allows the causal effect $P(\mathbf{y_x})$ to be identified as stated in Eq. 38, so these works can focus directly on estimating the specific equation in Eq. 38. On the counterfactual level, the identification problem may still need to be considered as there are non-identifiable $\mathcal{L}_3$ queries even in the Markovian case. Still, the problem is drastically simplified as confounding is no longer an issue. In fact, works like Pawlowski et al. (2020) and Sanchez-Martin et al. (2021) rely on this simplification for counterfactual estimation in the Markovian case and do not consider the identification problem despite potential the lack of identifiability of various queries.

Third, another notable simplification introduced by Markovianity is that distributions from the SCM can often be factorized into a convenient form. For example, the observational distribution $P(\mathbf{v})$ can be factorized as the product:

$$P(\mathbf{v}) = \prod_{V \in \mathbf{V}} P(v \mid \mathbf{pa}_V),$$

where $\mathbf{Pa}_V$ refer to the set of variables with an edge into V in $\mathcal{G}$, and v and $\mathbf{pa}_V$ are consistent with $\mathbf{v}$. Similarly, an interventional distribution $P(\mathbf{v_x})$ can be factorized as

$$P(\mathbf{v_x}) = \prod_{V \in \mathbf{V} \setminus \mathbf{X}} P(v \mid \mathbf{pa}_V),$$

where v and $\mathbf{pa}_V$ are consistent with $\mathbf{v}$ and $\mathbf{x}$. Counterfactual distributions follow a similar pattern. For example, when considering the graph $\mathcal{G}$ in Fig. 23a, $P(y_x, x'_z)$ factorizes as

$$P(y_x, x'_z) = \sum_{z'} P(y \mid x, z') P(x' \mid z) P(z'),$$

where $x \neq x'$. The pattern is that in every case, each term in the estimand can be written as the probability of a variable given its parents. This allows for models to encode the mechanism for each variable V as simply the conditional distribution $P(V \mid \mathbf{Pa}_V)$, abstracting away the functions $\mathcal{F}$ and exogenous noise $P(\mathbf{U})$ from the SCM. Architectures that benefit from this include DeepSCM from Pawlowski et al. (2020), which utilizes normalizing flows to model each conditional probability $P(V \mid \mathbf{Pa}_V)$, and VACA from Sanchez-Martin et al. (2021), which fits a variational graph autoencoder to the factorized distributions. In contrast, in non-Markovian cases, these factorizations typically do not hold. For example, if the true SCM induces the graph in Fig. 23c, then $P(y_x) \neq P(y \mid x)$ despite X being the only observed parent of Y. Works that rely on Markovian factorizations often do not have a straightforward generalization to non-Markovian cases.

Fourth, the works cited in this section that leverage the Markovianity assumption have still provided important contributions to the understanding and practical approaches to causal and counterfactual inference, and they are widely praised for their successes. However, Markovianity is a very restricting assumption and is unrealistic in many practical settings. For example, consider a simple study on the effects of exercise X on cholesterol Y, with age Z as a confounder. The relationship between variables can be modeled as shown in Fig. 23a. If, in the study, data on age is simply not collected, then Z becomes an unobserved confounder (see Fig. 23b), and Markovianity is immediately violated. The corresponding graph would look like Fig. 23c with Z omitted. In many cases, collecting the data may not even be an option. For example, if X is smoking, Y is cancer, and Z is the expressiveness of a gene, perhaps collecting data on gene expressivity is too difficult or even impossible. In general, one cannot expect to be able to collect data on every relevant variable in the study, so unobserved confounding is to be expected in realistic settings.

The solution presented in this paper is therefore important to understand the more complex, realistic, and general setting of potentially having unobserved confounding. Since the lack of the Markovianity

assumption greatly complicates the problem of counterfactual inference, many of the theoretical results in this work (e.g. Thms. 2 and 3) required more involved proofs. The result is that using Alg. 1, $\mathcal{G}$-NCMs can solve any instance of counterfactual identification and estimation even under unobserved confounding. This is demonstrated in the experiments, which consider several settings in which Markovianity is violated (e.g. the second, third, and fourth rows in Fig. 4, where the unobserved confounding is shown visually in the causal diagrams as red dashed bidirected arrows).

E FREQUENTLY ASKED QUESTIONS

Q1. Why do you assume that the causal diagram exists instead of learning it? Is it reasonable to expect that the causal diagram is available?

Answer. For a detailed discussion on this topic, see Sec. D.1. In summary, the assumption of the causal diagram is made out of necessity. The Neural Causal Hierarchy Theorem (N-CHT) (Xia et al., 2021, Corol. 1) states that distributions from lower layers almost always underdetermine higher layers in a measure-theoretic sense (Bareinboim et al., 2022, Thm. 1). This means that despite its universal expressivity, an NCM trained only on lower layer data (e.g., observational) cannot be reliably used for inferences on higher layers (e.g., interventional or counterfactual). Therefore, assumptions are necessary to make inferences across the PCH's layers. The causal diagram is a flexible assumption that is used throughout the literature and encodes a qualitative description of the generating model, which is often much easier to obtain than the actual mechanisms of the generating SCM (Pearl, 2000; Spirtes et al., 2000; Peters et al., 2017). Given the N-CHT, any attempts to learn the causal diagram will be incorrect or not meaningful unless other assumptions are made in this process (e.g., faithfulness). While other types of assumptions can be made, all such assumptions have their own benefits and drawbacks. The goal of this paper is not to decide which set of assumptions is the best, but rather to provide the toolkit for data analysts to perform the inferences once the assumptions have already been made.

Q2. How could the graphical assumption be relaxed?

Answer. Relaxing the graphical assumption is out of the scope of this work since the goal of this paper is first to establish a principled approach of using neural models to perform counterfactual inferences under general well-understood assumptions. Still, finding a solution to the problem when only partial information about the graph is available is an interesting direction for future work. One such possible approach, discussed at the end of Sec. D.1, is to utilize equivalence classes of causal diagrams, learnable from data, to perform inferences (Jaber et al., 2018; 2019; 2022, forthcoming). However, this approach has its own drawbacks, notably, that equivalence classes are not as descriptive as the true causal diagram, and therefore fewer queries will be identifiable. Still, this is one approach that can be considered in cases where assuming the availability of the full causal diagram is too stringent of a condition.

Q3. Why do you assume non-Markovianity? Is this more general than solutions that assume Markovianity?

Answer. That is an interesting question. We do not assume "non-Markovianity", but rather we do not make any assumption on Markovianity at all. Typically, Markovianity is the assumption and not the other way around. Markovianity requires that there exists no unobserved confounding, but works that do not assume Markovianity do not enforce such a requirement and are therefore more general because they work in both Markovian and non-Markovian cases. As mentioned earlier in the manuscript, it is hard to ascertain Markovianity in many practical settings, so not making the assumption is more reasonable for natural settings. Unlike many related works like those in Appendix C, this paper considers cases with unobserved confounding, which is represented graphically through dashed bidirected edges. This approach is more general in the sense that Alg. 1 can be used even in the presence of unobserved confounders, which is one of the major contributions of this paper. Markovianity is often assumed to simplify the problem setting, but it is often unrealistic to expect that data on all potential confounding variables are collected in a study. For a detailed discussion on this topic, see Appendix D.4.

Q4. Why do you use the $\mathcal{G}$-NCM definition in Def. 2 with feedforward neural networks and uniform noise? Could the results in this paper also hold for other function classes and noise?

Answer. The architectural choices made in Def. 2 are standard choices used to ground the discussion to a more concrete model. The same architecture is also used in the experiments. Still, the specific choice of architecture is not important to the theoretical results as long as key properties persist (e.g. $\mathcal{G}^{(\mathcal{L}_3)}$-consistency, $\mathcal{L}_3$-$\mathcal{G}$ expressiveness), which will often

be the case, since the arguments to prove the theoretical results in this work are mostly independent of the architecture. In particular, Sec. D.1 of Xia et al. (2021), which introduces the $\mathcal{G}$-NCM, discusses generalizations of NCMs in terms of functions and noise which still guarantee the relevant theoretical results, so long as the functions are sufficiently expressive and the noise variables have well-defined probability density functions.

Q5. Given that neural networks are universal approximators, it seems unsurprising that $\mathcal{G}$-NCMs can perform counterfactual inference. What makes $\mathcal{G}$-NCMs so special compared to other model classes?

Answer. The structure of this paper follows the delicate balance between expressiveness and constraint. As emphasized by the Causal Hierarchy Theorem (Bareinboim et al., 2022, Thm. 1) and (Xia et al., 2021, Corol. 1) , expressivity alone is not enough to infer higher layer quantities from lower layer data. Without assumptions, any model, including NCMs, trained only on $\mathcal{L}_1$ or $\mathcal{L}_2$ data will not contain any meaningful information about $\mathcal{L}_3$ quantities. Hence, assumptions in the form of constraints are discussed in Sec. 2. One important question is how to properly incorporate such constraints into a model. Constraining a model too much may hinder its expressiveness, and it may no longer be able to express the true model. Therefore, a proper model class for counterfactual inference must be able to incorporate the constraints without losing expressivity within the constrained class.

The $\mathcal{G}$-NCM model class has both the critical properties of expressiveness and constraint (Thms. 1 and 2) that are not enjoyed by many model classes. Examples of these properties are shown in Appendix D.3. Example 4 shows an example of a model class that, although expressive on all three layers, fails to properly incorporate the constraints of the graph, $\mathcal{G}$. Examples 5 and 6 show examples of model classes that incorporate the constraints of $\mathcal{G}$ but fail to achieve expressiveness within the constrained class. In contrast to these examples, the $\mathcal{G}$-NCM model class has a natural and intuitive approach to incorporating the constraints of $\mathcal{G}$ without damaging its expressiveness. These properties empower the $\mathcal{G}$-NCM the capability of solving the problems of counterfactual identification and estimation (see Thm. 3, Corols. 1 and 2, and Alg. 1). Examples 8 and 9 concretely demonstrate this ability.

When compared to similar works, like the ones in Appendix C, these properties are not as clearly established. While many of the works do incorporate graphical constraints, these constraints are often taken for granted as they are not formally discussed. In contrast, the ideas of graphical constraints are developed formally in this work in Sec. 2 and Appendix A.1. Many models in the related works assume Markovianity, which fails in terms of expressiveness for the same reason as Example 6. Some of the works only focus on the interventional ($\mathcal{L}_2$) level rather than the counterfactual ($\mathcal{L}_3$) level.

Q6. Why would one prefer to identify counterfactual quantities using NCMs? Would it not be easier first to identify the query using symbolic approaches, then use the NCM for estimation?

Answer. For a detailed discussion on the difference between symbolic and optimization approaches, we encourage the reader to read Appendix C.4 of Xia et al. (2021). In summary, the approach in this paper utilizes $\mathcal{G}$-NCMs to serve as a proxy model for the true generating SCM. In contrast, existing symbolic approaches like Correa et al. (2021) work directly with the graph and data, which is on a level of abstraction without the generating SCM. These works typically derive an exact answer for the target query by directly using the constraints of the graph. Indeed, one could use a hybrid approach of using a symbolic algorithm to identify a counterfactual query followed by training a $\mathcal{G}$-NCM to obtain an estimate. This is certainly an option, and the goal of this paper is to provide another solution route, not to decide for the data analyst which approach is better.

The proxy model approach presented in this paper is insightful to the problem of counterfactual inference because it shows that $\mathcal{G}$-NCMs are capable of jointly solving both counterfactual identification and estimation, which typically appear together in practice. Solving both problems together indicates that the $\mathcal{G}$-NCM is capable of both reasoning whether the query has any counterfactual meaning through identification and providing the practical benefits of statistical modeling through estimation. There are many reasons why one may prefer a proxy-based approach. First, the generative properties of the learned

proxy model allow it to serve as an infinite source of samples for the user. Second, the trained model could provide quick estimation results for multiple queries without the need for retraining, provided the queries are identifiable. Third, the idea of minimizing and maximizing the query in Alg. 1 is central to identification problems. Hence, a similar procedure can be used in similar identification problem settings without an existing symbolic solution. Fourth, this idea can potentially be used to bound non-identifiable queries, which is an exciting avenue for future work.

Q7. Why would one want to learn an entire SCM to solve the estimation problem for a specific query instead of just directly estimating the query?

Answer. Indeed, if the user simply wants to estimate one query and is knowledgeable about the estimand, it may be more efficient, statistically or computationally, to focus efforts on just estimating that query. However, there are many reasons why the proxy model approach used in this work may be beneficial. First, by learning an entire SCM and estimating queries using Eq. 4, one can apply the same procedure to any potential query, reducing the complexity of the solution. This relieves the user of the need to derive an estimand. Even if the estimand is known, developing a specific approach to estimating this estimand can be difficult depending on the form of the estimand. Second, if the user wants to estimate multiple queries, they could potentially reuse the same $\mathcal{G}$-NCM without retraining, provided that the queries are identifiable. Third, as a generative model, the $\mathcal{G}$-NCM can be used as an infinite source of samples should the user desire this in addition to having an estimate for the query.

Q8. Why is the GAN architecture used for the experiments?

Answer. We found that using a GAN approach was the most straightforward way to implement the $\mathcal{G}$-NCM in practice. When using an implicit generative model approach, one can construct a $\mathcal{G}$-NCM exactly as described in Def. 2 and train it as specified in Alg. 3. Using a discriminator network as the divergence function ($\mathbb{D}_P$) provides a flexible and practical approach to generating a loss for optimization, as understood in GAN literature. Alternative approaches to modeling the divergence functions all have practical drawbacks. For example, KL divergence is typically intractable to compute directly. Appendix B.6.1 explains why an explicit generative model approach like the MLE-NCM (used in Xia et al. (2021)) is significantly more complicated. We also include experiments comparing against the MLE-NCM, which demonstrated some key flaws of the MLE approach, such as poor runtime scalability (see Fig. 5) or having a bias to output "ID" even in non-ID settings (see Fig. 4). For further results, see Appendix B.1. Still, the key theoretical results in this work do not depend on the specific architecture used in practice, and the choice of using GANs is unimportant in this discussion. Users can use any architecture they see fit while still applying the results in this work.

Q9. Given the training instability of GANs, is the GAN-NCM approach guaranteed to work?

Answer. Indeed, the theoretical results described in Sec. 3 assume that infinite data is provided and optimization is perfect, which is unrealistic in practice. Hence, a hypothesis testing procedure is required to decide identifiability, described in Appendix B.4. This means that the results from solving the identification and estimation problems will have some degree of error in practice. While deriving bounds on the error could be an interesting direction for future work, the error is shown to be quite manageable empirically using GANs, as shown in the experiments in Sec. 5. In the estimation experiments, the error is shown to decrease as samples increase, demonstrating a level of statistical consistency.

Q10. Does the work apply to cases with continuous variables?

Answer. The theoretical results in this work assume that the variable domains are finite, and no claims are made about continuous variables. If extending to the continuous case, on the theoretical side, counterfactual distributions and queries would have to be properly defined and new machinery based on measure theory evoked. It is non-trivial to prove some of the theoretical results of this paper in continuous settings. While the exogenous variables and structural functions of the true model can take any form, this paper relies on the discreteness

of the endogenous variables to ensure that there exists a $\mathcal{G}$-NCM that can exactly match the true model (see also Zhang et al. (2022)). It is currently an open problem to generalize these results to the continuous case and in generality.

On the experimental side, sampling or maximizing/minimizing the query from a $\mathcal{G}$-NCM may have complications depending on the definition of a counterfactual query/distribution in a continuous setting. Intuitively, however, if the query can be sampled via Alg. 2, then similar techniques applied in our paper can be extended without much change (although currently without theoretical guarantees). For example, if the query is $E[Y_{X=x}|X \geq 0]$ with both continuous X and Y, one could (1) sample several instances of X, (2) filter out negative samples, (3) sample Y using the values of $\mathbf{U}$ from the remaining instances, and (4) compute the average out of these values of Y. Then, this Monte Carlo estimate can be used in Alg. 3 for maximizing/minimizing the query and estimating the final result if identifiable. If the query is not so easily compatible with Alg. 2, then the approach may need to be adjusted to accommodate continuous definitions.

Q11. Why are there no empirical studies comparing other neural estimation methods (such as the ones in Appendix C)?

Answer. Despite their notable contributions, as discussed earlier, related works like the one in Appendix C are not as general as the approach described in this work. For example, some works assume Markovianity, some only solve queries from $\mathcal{L}_2$ or specific queries from $\mathcal{L}_3$, and some do not consider identifiability (see Appendix C for a detailed discussion on each of these works). The works in Footnote 1 assume conditional ignorability or backdoor admissibility, which only hold in a few graphs, such as graph (a) from Fig. 11 (it does not hold in any other graph in Fig. 11). The approach presented in this paper is the first work to use neural models to identify any $\mathcal{L}_3$ query given any combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ data and any graph setting (even non-Markovian cases). It will also perform the estimation for identifiable cases, but it will return "FAIL" instead of estimating non-identifiable cases. Hence, comparisons with these specific related works are not appropriate since they cannot handle such general settings.

Q12. How does the proposed approach scale to higher dimensional variables or with a greater number of variables?

Answer. The training of the GAN-NCM scales tractably with higher dimensionality or variables, as shown in Fig. 5. While having higher dimensions or more variables may require a larger architecture, and therefore more training time, typically this also comes with a higher data requirement. Hence, the architecture can be scaled with the complexity of the problem, and although complex problems will require more training time, the user may be willing to spend more computational resources on a more complex problem. We hope this work can be a baseline and act as a catalyst to the study of different neural architectures with different computational and statistical properties.