Intervention Generalization: A View from Factor Graph Models

Gecia Bravo-Hermsdorff* **David S. Watson†** **Jialin Yu*** **Jakob Zeitler‡** **Ricardo Silva***

Abstract

One of the goals of causal inference is to generalize from past experiments and observational data to novel conditions. While it is in principle possible to eventually learn a mapping from a novel experimental condition to an outcome of interest, provided a sufficient variety of experiments is available in the training data, coping with a large combinatorial space of possible interventions is hard. Under a typical sparse experimental design, this mapping is ill-posed without relying on heavy regularization or prior distributions. Such assumptions may or may not be reliable, and can be hard to defend or test. In this paper, we take a close look at how to warrant a leap from past experiments to novel conditions based on minimal assumptions about the factorization of the distribution of the manipulated system, communicated in the well-understood language of factor graph models. A postulated *interventional factor model* (IFM) may not always be informative, but it conveniently abstracts away a need for explicitly modeling unmeasured confounding and feedback mechanisms, leading to directly testable claims. Given an IFM and datasets from a collection of experimental regimes, we derive conditions for identifiability of the expected outcomes of new regimes never observed in these training data. We implement our framework using several efficient algorithms, and apply them on a range of semi-synthetic experiments.

1 Introduction

Causal inference is a fundamental problem in many sciences, such as clinical medicine [8, 73, 69] and molecular biology [66, 41, 29]. For example, causal inference can be used to identify the effects of chemical compounds on cell types [73] or determine the underlying mechanisms of disease [52].

One particular challenge in causal inference is *generalization* — the ability to extrapolate knowledge gained from past experiments and observational data to previously unseen scenarios. Consider a laboratory that has performed several gene knockouts and recorded subsequent outcomes. Do they have sufficient information to predict how the system will behave under some new combination(s) of knockouts? Conducting all possible experiments in this setting would be prohibitively expensive and time consuming. A supervised learning method could, in principle, map a vector representation of the design to outcome variables of interest. However, past experimental conditions may be too sparsely distributed in the set of all possible assignments, and such a direct supervised mapping would require leaps of faith about how assignment decisions interact with the outcome, even if under the guise of formal assumptions such as linearity.

In this paper, we propose a novel approach to the task of *intervention generalization*, i.e., predicting the effect of unseen treatment regimes. We rely on little more than a postulated factorization of the

*Department of Statistical Science, University College London.
†Department of Informatics, King's College London.
‡Department of Computer Science, University College London.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

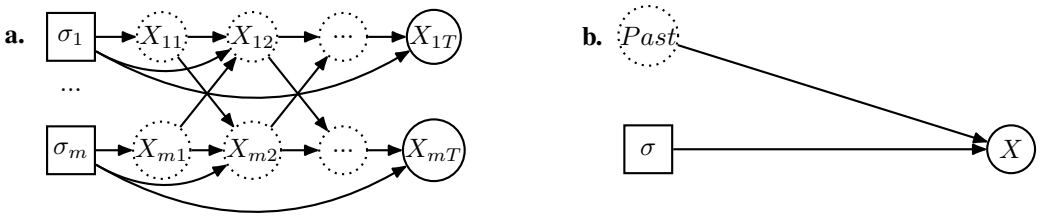


Figure 1: **(a)** A dynamic process where only the final step is (simultaneously) observed. Intervention variables are represented as white squares, random variables as circles, and dashed lines indicate hidden variables. **(b)** A bird's eye unstructured view of the sampling process, where an intervention vector dictates σ the distribution obtained at an agreed-upon time frame T.

distribution describing how intervention variables σ interact with a random process X, and a possible target outcome Y.

Related setups appear in the causal modeling literature on soft interventions [18], including its use in methods such as causal bandits [45] and causal Bayesian optimization [4] (see Section 6). However, these methods rely on directed acyclic graphs (DAGs), which may be hard to justify in many applications. Moreover, in the realistic situation where the target of a real-world intervention is a *set* of variables [26] and not only the "child" variable within a DAG factor, the parent-child distinction is blurred and previous identification results do not apply. Without claiming that our proposal is appropriate for every application, we suggest the following take on intervention generalization: *model a causal structure as a set of soft constraints, together with their putative (arbitrary but local) modifications by external actions*. Our *interventional factor model* (IFM) leads to provable intervention generalization via a factor graph decomposition which, when informative, can be tested without further assumptions beyond basic relations of conditional independence. IFMs are fully agnostic with regards to cycles or hidden variables, in the spirit of Dawid [23]'s decision-theoretic approach to causal inference, where the key ingredient boils down to statements of conditional independence among random and intervention variables.

Our primary contributions are as follows. (1) We introduce the *interventional factor model* (IFM), which aims to solve intervention generalization problems using only claims about which interventions interact with which observable random variables. (2) We establish necessary and sufficient conditions for the identifiability of treatment effects within the IFM framework. (3) We adapt existing results from conformal inference to our setting, providing distribution-free predictive intervals for identifiable causal effects with guaranteed finite sample coverage. (4) We implement our model using efficient algorithms, and apply them to a range of semi-synthetic experiments.

2 Problem Statement

Data assumptions. Fig. 1**(a)** illustrates a generative process common to many applications and particularly suitable to our framework: a perturbation, here represented by a set of interventional variables σ, is applied and a dynamic feedback loop takes place. Often in these applications (e.g, experiments in cell biology [66] and social science [59]), the sampling of this process is highly restrictive, and the time-resolution may boil down to a single *snapshot*, as illustrated by Fig. 1**(b)**. We assume that data is given as samples of a random vector X collected cross-sectionally at a well-defined time-point (not necessarily at equilibrium) under a well-defined intervention vector σ. The importance of establishing a clear sampling time in the context of graphical causal models is discussed by [21]. The process illustrated in Fig. 1**(a)** may suggest no particular Markovian structure at the time the snapshot is taken. However, in practice, it is possible to model the black-box sampling process represented in Fig. 1**(b)** in terms of an *energy function* representing soft constraints [48]. These could be the result of particular equilibrium processes [47], including deterministic differential equations [40], or empirically-verifiable approximations [59].

The role of a causal model is to describe how intervention σ locally changes the energy function. In the context of causal DAG models, sometimes this takes the guise of "soft interventions" and other variations that can be called "interventions on structure" [43, 55] or edge interventions [70]. Changes of structural coefficients in possibly cyclic models have also been considered [37]. The model family we propose takes this to the most abstract level, modeling energy functions via recombinations of

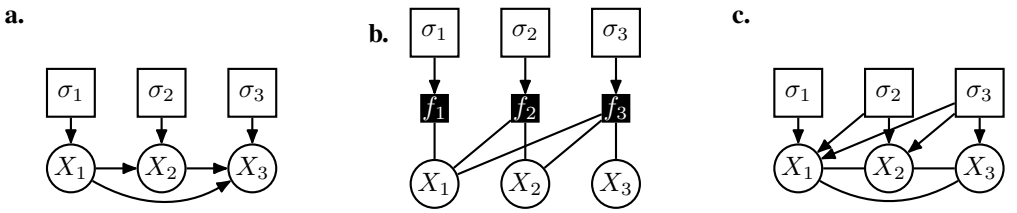


Figure 2: Examples of distinct causal graphical models, expressing different factorization assumptions. Random variables are represented as circles, intervention variables as white squares, and factors as black squares. **(a)** A directed acyclic graph (DAG) with explicit intervention variables. **(b)** The corresponding interventional factor model (IFM) for **(a)**. **(c)** A Markov random field (MRF) with interventional variables, thus, forming a chain graph.

local interventional effects without acyclicity constraints, as such constraints should not be taken for granted [22] and are at best a crude approximation for some problems at hand (e.g., [66]).

Background and notation. The notion of an *intervention variable* in causal inference encodes an action that modifies the distribution of a system of random variables. This notion is sometimes brought up explicitly in graphical formulations of causal models [60, 72, 23]. To formalize it, let σ denote a vector of intervention variables (also known as *regime indicators*), with each σ_i taking values in a finite set $\{0, 1, 2, \dots, \aleph_i - 1\}$. A fully specified vector of intervention variables characterizes a *regime* or an *environment* (we use these terms interchangeably). Graphically, we represent intervention variables using white squares and random variables using circles (Fig. 2). We use subscripts to index individual (random or interventional) variables and superscripts to index regimes. For instance, X_i^j denotes the random variable X_i under regime σ^j, while σ_i^j denotes the ith intervention variable of the jth environment. (Note that the order of the variables is arbitrary.) We occasionally require parenthetical superscripts to index samples, so $x_i^{j(k)}$ denotes the kth sample of X_i under regime σ^j.

For causal DAGs, Pearl's do operator [60] denotes whether a particular random variable X_i is manipulated to have a given value (regardless of the values of its parents). As an example, consider a binary variable $X_i \in \{\text{TRUE}, \text{FALSE}\}$. We can use a (categorical) intervention variable σ_i to index the two "do-interventional" regimes: $\sigma_i = 1$ denotes "$do(X_i = \text{TRUE})$", and $\sigma_i = 2$ denotes "$do(X_i = \text{FALSE})$." We reserve $\sigma_i = 0$ to denote the choice of "no manipulation", also known as the "observational" regime. However, one need not think of this $\sigma_i = 0$ setting as fundamentally different than the others; indeed, it is convenient to treat all regimes as choices of data generating process.[4] Moreover, there is no need for intervention variables to correspond to deterministic settings of random variables; they may in principle describe any well-defined change in distribution, such as stochastic or conditional interventions [23, 17]. As a stochastic example, $\sigma_i = 3$ could correspond to randomly choosing $do(X_i = \text{TRUE})$ 30% of the time and $do(X_i = \text{FALSE})$ 70% of the time. As a conditional example, $\sigma_i = 4$ could mean "if parents of X_i satisfy a given condition, $do(X_i = \text{TRUE})$, otherwise do nothing". When X_i can take more values, the options for interventional regimes become even more varied. The main point to remember is that $\sigma_i = 1, \sigma_i = 2, \dots$ should be treated as distinct categorical options, and we reserve $\sigma_i = 0$ to denote the "observational" case of "no manipulation".

As discussed in the previous section, the probability density/mass function $p(x; \sigma)$ can be the result of a feedback process that does not naturally fit a DAG representation. Indeed, a growing literature in causal inference carefully considers how DAGs may give rise to equilibrium distributions (e.g., [47, 12, 11]), or marginals of continuous-time processes (e.g., [57]). However, they come with considerable added complexity of assumptions to ensure identifiability. In this work, we abstract away all low-level details about how an equilibrium distribution comes to be, and instead require solely a model for how a distribution $p(x)$ factorizes as a function of σ. These assumptions are naturally formulated as a factor graph model [44] augmented with intervention variables, which we call an *interventional factor model (IFM)*.

[4] In general, there can be infinitely many choices for interventional settings. However, for the identifiability results in the next section, we cover the finite case only, as it removes the need for smoothness assumptions on the effect of intervention levels.

Problem statement. We are given a space Σ of possible values for an intervention vector σ of dimension d, making $|\Sigma| \leq \prod_{i=1}^{d} \aleph_i$. For a range of training regimes $\Sigma_{\text{train}} \subset \Sigma$, we are given collection of datasets $\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^t$, with dataset $\mathcal{D}^j$ collected under environment/regime $\sigma^j \in \Sigma_{\text{train}}$. The goal is to learn $p(x; \sigma^\star)$, and $\mu(\sigma^\star) := \mathbb{E}[Y; \sigma^\star]$, for all test regimes $\sigma^\star \in \Sigma_{\text{test}} = \Sigma \backslash \Sigma_{\text{train}}$.

In each training dataset, we measure a sample of post-treatment i.i.d. draws of some m-dimensional random vector X (possibly with $m \neq d$, as there is no reason to always assume a one-to-one mapping between intervention and random variables), and optionally an extra outcome variable Y. The data generating process $p(x; \sigma^j)$ is unknown. We assume $Y \perp\!\!\!\perp \sigma \mid X$ for simplicity[5], which holds automatically in cases where Y is a known deterministic summary of X. We are also given a factorization of $p(x; \sigma)$,

$$p(x; \sigma) \propto \prod_{k=1}^{l} f_k(x_{S_k}; \sigma_{F_k}), \ \forall \sigma \in \Sigma, \tag{1}$$

where $S_k \subseteq [m]$ and $F_k \subseteq [d]$ define the causal structure. The (positive) functions $f_k(\cdot; \cdot)$ are unknown. The model $p(y \mid x)$ can also be unknown, depending on the problem.

As illustrated in Figs. 2**(a-b)**, such an intervention factor model (IFM) can also encode (a relaxation of) the structural constraints implied by DAG assumptions. In particular, note the one-to-one correspondence between the DAG factorization in Fig. 2**(a)** and the factors in the IFM in Fig. 2**(b)**: $f_1(x_1; \sigma_1) := p(x_1; \sigma_1)$, $f_2(x_1, x_2; \sigma_2) := p(x_2 \mid x_1; \sigma_2)$, and $f_3(x_1, x_2, x_3; \sigma_3) := p(x_3 \mid x_1, x_2; \sigma_3)$. This explicitly represents that the conditional distribution of x_1 given all other variables fully factorizes in σ: i.e., $p(x_1 \mid x_2, x_3; \sigma) \propto f_1(x_1; \sigma_1) f_2(x_1, x_2; \sigma_2) f_3(x_1, x_2, x_3; \sigma_3)$. Such factorization is not enforced by usual parameterizations of Markov random fields (i.e., graphical models with only undirected edges) or chain graphs (graphical models with acyclic directed edges and undirected edges) [25] such as the example shown in Fig. 2**(c)**.

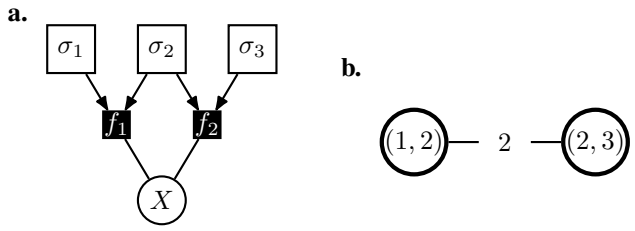
Scope and limitations. The factorization in Eq. (1) may come from different sources, e.g., from knowledge about physical connections (it is typically the case that one is able to postulate which variables are directly or only indirectly affected by an intervention), or as the result of structure learning methods (e.g., [1]). For structure learning, faithfulness-like assumptions [72] are required, as conditional independencies discovered under configurations Σ_{train} can only be extrapolated to Σ_{test} by assuming that independencies observed over particular values of σ can be generalized across all regimes. We do not commit to any particular structure learning technique, and refer to the literature on eliciting and learning graphical structure for a variety of methods [42]. In Appendix A, we provide a general guide on structure elicitation and learning, and a primer on causal modeling and reasoning based solely on abstract conditional independence statements, without a priori commitment to a particular family of graphical models [23].

Even then, the structural knowledge expressed by the factorization in Eq. (1) may be uninformative. Depending on the nature of Σ_{train} and Σ_{test}, it may be the case that we cannot generalize from training to test environments, and $p(x; \sigma^\star)$ is unidentifiable for all $\sigma^\star \in \Sigma_{\text{test}}$. However, all methods for causal inference rely on a trade-off between assumptions and informativeness. For example, unmeasured confounding may imply no independence constraints, but modeling unmeasured confounding is challenging, even more so for equilibrium data without observable dynamics. If we can get away with pure factorization constraints implied by an array of experimental conditions and domain knowledge, we should embrace this opportunity. This is what is done, for instance, in the literature on causal bandits and causal Bayesian optimization [45, 49, 4, 75], which leverage similar assumptions to decide what to do next. However, we are *not* proposing a method for bandits, Bayesian optimization, or active learning. The task of estimating $p(x; \sigma^\star)$ and $\mu(\sigma^\star)$ for a novel regime is relevant in itself.

3 Interventional Factor Model: Identification

We define our *identification problem* as follows: given the *population* distributions $\mathbb{P}(\Sigma_{\text{train}}) := \{p(x; \sigma^1), p(x; \sigma^2), \ldots, p(x; \sigma^t)\}$ for the training regimes $\sigma^j \in \Sigma_{\text{train}}$, and knowledge of the factorization assumptions as given by Eq. (1), can we identify a given $p(x; \sigma^\star)$ corresponding to some unobserved test regime $\sigma^\star \in \Sigma_{\text{test}}$?

[5] Otherwise, we can just define $Y := X_m$, and use the identification results in Section 3 to check whether Y can be predicted from σ.



c.

σ_1	σ_2	σ_3
0	0	0
0	1	0
1	0	0
1	1	0
0	0	1
0	1	1

Figure 3: **(a)** An interventional factor model (IFM) with three (binary) intervention variables. **(b)** The junction tree of the *σ-graph* associated with the IFM in **(a)**: intervention variables are arranged as a hypergraph, where the hypervertices represent the sets of intervention variables that share a factor and the edge represents the overlap between the two sets of intervention variables. **(c)** A table displaying the assignments for regimes in Σ_{train}. From this Σ_{train} and the assumptions in **(a)**, it is possible to generalize to the two missing regimes, i.e., $\sigma_1 = \sigma_2 = \sigma_3 = 1$, and $\sigma_1 = \sigma_3 = 1$, $\sigma_2 = 0$.

This identification problem is not analyzed in setups such as causal Bayesian optimization [4, 75], which build on DAGs with interventions targeting single variables. Without this analysis, it is unclear to what extent the learning might be attributed to artifacts due to the choice of prior distribution or regularization, particularly for a sparsely populated Σ_{train}. As an extreme case that is not uncommon in practice, if we have binary intervention variables and never observe more than one σ_i set to 1 within the same experiment, it is unclear why we should expect a non-linear model to provide information about pairs of assignments $\sigma_i = \sigma_j = 1$ using an off-the-shelf prior or regularizer. Although eventually a dense enough process of exploration will enrich the database of experiments, this process may be slow and less suitable to situations where the goal is not just to maximize some expected reward but to provide a more extensive picture of the dose-response relationships in a system. While the smoothness of $p(x; \sigma)$ as a function of σ is a relevant domain-specific information, it complicates identifiability criteria without further assumptions. In what follows, we assume no smoothness conditions, meaning that for some pair $p(x; \sigma^a), p(x; \sigma^b)$, with $a \neq b$, these probability density/mass functions are allowed to be arbitrarily different. Such setting is particularly suitable for situations where interventions do take categorical levels, with limited to no magnitude information.

Identification: preliminaries. Before introducing the main result of this section, let us start by considering the toy example displayed in Fig. 3: Fig. 3**(a)** shows the graphical model (how X could be factorized is not relevant here), and Fig. 3**(c)** shows the assignments for the training regimes Σ_{train} (all intervention variables are binary). This training set lacks the experimental assignment $\sigma_1 = \sigma_2 = \sigma_3 = 1$.[6] However, this regime is implied by the model and Σ_{train}. To see this, consider the factorization $p(x; \sigma) \propto f_1(x; \sigma_1, \sigma_2) f_2(x; \sigma_2, \sigma_3)$, it implies:

$$\frac{p\big(x; (1,1,0)\big)}{p\big(x; (0,1,0)\big)} \propto \frac{f_1\big(x; (1,1)\big)}{f_1\big(x; (0,1)\big)},$$

where we used $(1,1,0)$ etc. to represent the σ assignments. Because both $(1,1,0)$ and $(0,1,0)$ are in Σ_{train}, the ratio is identifiable up to a multiplicative constant. Moreover, multiplying and dividing the result by $f_2\big(x; (1,1)\big)$, we get:

$$\frac{p\big(x; (1,1,0)\big)}{p\big(x; (0,1,0)\big)} \propto \frac{f_1\big(x; (1,1)\big)}{f_1\big(x; (0,1)\big)} \frac{f_2\big(x; (1,1)\big)}{f_2\big(x; (1,1)\big)} \propto \frac{p\big(x; (1,1,1)\big)}{p\big(x; (0,1,1)\big)},$$

from which, given that $(0,1,1)$ is also in Σ_{train}, we can derive $p\big(x; (1,1,1)\big)$.

3.1 Message Passing Formulation

The steps in this reasoning can be visualized in Fig. 3**(b)**. The leftmost *hypervertex* represents (σ_1, σ_2) and suggests σ_1 can be isolated from σ_3. The first ratio $p\big(x; (1,1,0)\big)/p\big(x; (0,1,0)\big)$ considers three roles: σ_1 can be isolated (it is set to a "baseline" of 0 in the denominator) and σ_3 is yet to be considered (it is set to the baseline value of 0 in the numerator). This ratio sets the stage for the next step where (σ_2, σ_3) is "absorbed" in the construction of the model evaluated at $(1,1,1)$. The entries in Σ_{train} were chosen so that we see all four combinations of (σ_1, σ_2) and all four combinations of

[6] As well as the assignment $\sigma_1 = \sigma_3 = 1, \sigma_2 = 0$, which can be recovered by an analogous argument.

(σ_2, σ_3), while avoiding the requirement of seeing all eight combinations of $(\sigma_1, \sigma_2, \sigma_3)$. How to generalize this idea is the challenge. Next, we present our first proposed solution.

Definitions. Before we proceed, we need a few definitions. First, recall that we represent the *baseline* (or "observational") regime as $\sigma_1 = \sigma_2 = \cdots = \sigma_d = 0$. This describes data captured under a default protocol, e.g., a transcriptomic study in which no genes are knocked down. Let $\Sigma^0_{[Z]}$ denote the set of all environments $\sigma^j \in \Sigma$ such that $\sigma^j_i = 0$ if $i \notin Z$. For instance, for the example in Fig. 3, a set $Z := \{2\}$ implies that $\Sigma^0_{[Z]} = \big\{(0,0,0), (0,1,0)\big\}$. Finally, let $\sigma^{[Z(\star)]}$ be the intervention vector given by $\sigma_i = \sigma^\star_i$, if $i \in Z$, and 0 otherwise. In what follows, we also use of the concepts of graph *decompositions*, *decomposable graphs* and *junction trees*, as commonly applied to graphical models [46, 20]. In Appendix B, we review these concepts.[7] An IFM $\mathcal{I}$ with intervention vertices $\sigma_1, \ldots, \sigma_d$ has an associated *σ-graph* denoted by $\mathcal{G}_{\sigma(\mathcal{I})}$, defined as an undirected graph with vertices $\sigma_1, \ldots, \sigma_d$, and where edge $\sigma_i - \sigma_j$ is present if and only if σ_i and σ_j are simultaneously present in at least one common factor f_k in $\mathcal{I}$. For example, the σ-graph of the IFM represented in Fig. 3**(a)** is $\sigma_1 - \sigma_2 - \sigma_3$. This is a decomposable graph with vertex partition $A = \{\sigma_1\}$, $B = \{\sigma_2\}$ and $C = \{\sigma_3\}$. As this σ-graph is an undirected decomposable graph, it has a junction tree, which we depict in Fig. 3**(b)**.

Identification: message passing formulation. Let $\mathcal{I}$ be an IFM representing a model for the m-dimensional random vector X under the d-dimensional intervention vector $\sigma \in \Sigma$. Model $\mathcal{I}$ has unknown factor parameters but a known factorization $p(x; \sigma) \propto \prod_k f_k(x_{S_k}; \sigma_{F_k})$.

Theorem 3.1 *Assume that the σ-graph $\mathcal{G}_{\sigma(\mathcal{I})}$ is decomposable. Given a set $\Sigma_{\text{train}} = \{\sigma^1, \ldots, \sigma^t\} \subseteq \Sigma$ and known distributions $p(x; \sigma^1), \ldots, p(x; \sigma^t)$, the following conditions are sufficient to identify any $p(x; \sigma^\star)$, $\sigma^\star \in \Sigma$:*

i) all distributions indexed by $\sigma^1, \ldots, \sigma^t, \sigma^\star$ have the same support; and

ii) $\Sigma^0_{[F_k]} \in \Sigma_{\text{train}}$ for all F_k in the factorization of $\mathcal{I}$.

The algorithm for computing $p(x; \sigma^\star)$ works as follows. Construct a junction tree $\mathcal{T}$ for $\mathcal{I}$, choose an arbitrary vertex in $\mathcal{T}$ to be the root, and direct $\mathcal{T}$ accordingly. If V_k is a hypervertex in $\mathcal{T}$, let D_k be the union of all intervention variables contained in the descendants of V_k in $\mathcal{T}$. Let B_k be the intersection of the invervention variables contained in V_k with the intervention variables contained in the parent $V_{\pi(k)}$ of V_k in $\mathcal{T}$.

We define a message *from a non-root vertex V_k to its parent $V_{\pi(k)}$ as*

$$m^x_k := \frac{p(x; \sigma^{[D_k(\star)]})}{p(x; \sigma^{[B_k(\star)]})}, \tag{2}$$

with the update equation

$$p(x; \sigma^{[D_k(\star)]}) \propto p(x; \sigma^{[F_k(\star)]}) \prod_{V_{k'} \in ch(k)} m^x_{k'}, \tag{3}$$

where the product over $ch(k)$, the children of V_k in $\mathcal{T}$, is defined to be 1 if $ch(k) = \emptyset$.

(Proofs are provided in Appendix B.) In particular, if no factor contains more than one intervention variable, the corresponding σ-graph will be fully disconnected. This happens, e.g, for an IFM derived from a DAG without hidden variables and where each intervention has one child, as in Fig. 2.

3.2 Algebraic Formulation

If a σ-graph is not decomposable, the usual trick of triangulation prior to clique extraction can be used [46, 20], at the cost of creating cliques which are larger than the original factors. Alternatively, and a generalization of Eqs. (2) and (3), we consider transformations of the distributions in $\mathbb{P}(\Sigma_{\text{train}})$

[7] **Quick summary**: a junction tree is formed by turning the cliques of an undirected graph into the (hyper)vertices of the tree. Essential to the definition, a junction tree $\mathcal{T}$ must have a *running intersection property*: given an intersection $S := H_i \cap H_j$ of any two hypervertices H_i, H_j, all hypervertices in the unique path in $\mathcal{T}$ between H_i and H_j must contain S. This property captures the notion that satisfying local agreements ("equality of properties" of subsets of elements between two adjacent hypervertices in the tree) should imply global agreements. A decomposable graph is simply a graph whose cliques can be arranged as a junction tree.

	σ_1	σ_2	σ_3	f_1^{00}	f_1^{01}	f_1^{10}	f_1^{11}	f_2^{00}	f_2^{01}	f_2^{10}	f_2^{11}	f_3^{00}	f_3^{01}	f_3^{10}	f_3^{11}
q_1	0	0	0	✓				✓				✓			
q_2	0	1	0		✓					✓		✓			
q_3	1	0	0			✓		✓						✓	
q_4	1	1	0				✓			✓				✓	
q_5	0	0	1	✓					✓				✓		
q_6	0	1	1		✓						✓		✓		
q_7	1	0	1			✓			✓						✓
$p^\star$	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1

Figure 4: Example of how to infer the target distribution $p\big(x;(1,1,1)\big)$ from the other seven possible regimes given the postulated factorization $p(x;\sigma) \propto f_1(x;\sigma_1,\sigma_2)f_2(x;\sigma_2,\sigma_3)f_3(x;\sigma_1,\sigma_3)$. We use f_k^{ab} as a shorthand notation for $f_k(x;\sigma_{k_1}=a,\sigma_{k_2}=b)$, e.g, $p\big(x;(1,1,1)\big) \propto f_1^{11}f_2^{11}f_3^{11}$. For a PR-transformation $(f_1^{00}f_2^{00}f_3^{00})^{q_1} \times (f_1^{01}f_2^{10}f_3^{00})^{q_2} \times \cdots \times (f_1^{10}f_2^{01}f_3^{11})^{q_7}\ f_1^{11}f_2^{11}f_3^{11}$ up to a multiplicative constant independent of x, we need to find a solution satisfying: $q_1+q_5=0$, $q_2+q_6=0$, $\ldots$, $q_7=1$. (In the table, this can be seen by going through each f_k^{ab} column and adding up the corresponding exponents q_j when there is a tick in row j; the sum must agree with the corresponding entry in the final row). A solution is $q_1=q_4=q_6=q_7=1$, $q_2=q_3=q_5=-1$, corresponding to $p(x;\sigma^\star) \propto \big(p(x;\sigma^1)p(x;\sigma^4)p(x;\sigma^6)p(x;\sigma^7)\big)\,/\,\big(p(x;\sigma^2)p(x;\sigma^3)p(x;\sigma^5)\big)$.

by products and ratios. We define a *PR-transformation* of a density set $\{p(x;\sigma^1),\ldots,p(x;\sigma^t)\}$ as any formula $\prod_{i=1}^t p(x;\sigma^i)^{q_i}$, for a collection $q_1,\ldots,q_t$ of real numbers.

Theorem 3.2 *Let $\sigma_{F_k}^v$ denote a particular value of σ_{F_k}, and let $\mathbb{D}_k$ be the domain of σ_{F_k}. Given a collection $\mathbb{P}(\Sigma_{\text{train}}) := \{p(x;\sigma^1),\ldots,p(x;\sigma^t)\}$ and a postulated model factorization $p(x;\sigma) \propto \prod_{k=1}^l f_k(x_{S_k};\sigma_{F_k})$, a sufficient and almost-everywhere necessary condition for a given $p(x;\sigma^\star)$ to be identifiable by PR-transformations of $\mathbb{P}(\Sigma_{\text{train}})$ is that there exists some solution to the system*

$$\forall k \in \{1,2,\ldots,l\}, \forall \sigma_{F_k}^v \in \mathbb{D}_k, \left(\sum_{i=1:\sigma_{F_k}^i=\sigma_{F_k}^v}^{t} q_i\right) = \mathbb{1}(\sigma_{F_k}^\star = \sigma_{F_k}^v), \tag{4}$$

where $\mathbb{1}(\cdot)$ is the indicator function returning 1 or 0 if its argument is true or false, respectively.

The solution to the system gives the PR-transformation. An example is shown in Fig. 4. The main idea is that, for a fixed x, any factor $f_k(x_{S_k};\sigma_{F_k})$ can algebraically be interpreted as an arbitrary symbol indexed by σ_{F_k}, say "$f_k^{\sigma_{F_k}}$". This means that any density $p(x;\sigma^i)$ is (proportional to) a "squarefree" monomial m^i in those symbols (i.e, products with exponents 0 or 1). We need to manipulate a set of monomials (the unnormalized densities in $\mathbb{P}(\Sigma_{\text{train}})$) into another monomial $m^\star$ (the unnormalized $p(x;\sigma^\star)$). More generally, the possible set functions $g(\cdot), h(\cdot)$ such that $g(m^\star,\{m^1,\ldots,m^t\}) = h(\{m^1,\ldots,m^t\})$, with $g(\cdot)$ invertible in $m^\star$, suggest that $g(\cdot)$ and $h(\cdot)$ must be the product function (monomials are closed under products, but not other analytical manipulations), although we stop short of stating formally the conditions required for PR-transformations to be complete in the space of all possible functions of $\mathbb{P}(\Sigma_{\text{train}})$. This would be a stronger claim than Theorem 3.2, which shows that Eq. (4) is almost-everywhere complete in the space of PR transformations.

The message passing scheme and the algebraic method provide complementary views, with the former giving a divide-and-conquer perspective that identifies subsystems that can be estimated without requiring changes from the baseline treatment everywhere else. The algebraic method is more general, but suggests no hierarchy of simpler problems. These results show that the factorization over X is unimportant for identifiability, which may be surprising. Appendix C discusses the consequences of this finding, along with a discussion about requirements on the size of Σ_{train}.

4 Learning Algorithms

The identification results give a license to choose any estimation method we want if identification is established — while we must make the assumption of the model factorizing according to a postulated IFM, we are not required to explicitly use a likelihood function. Theorems 3.1 and 3.2 provide ways of constructing a target distribution $p(x;\sigma^\star)$ from products of ratios of densities from $\mathbb{P}(\Sigma_{\text{train}})$. This

suggests a plug-in approach for estimation: estimate products of ratios using density ratio estimation methods and multiply results. However, in practice, we found that fitting a likelihood function directly often works better than estimating the product of density ratios, even for intractable likelihoods. We now discuss three strategies for estimating some $\mathbb{E}[Y; \sigma^\star]$ of interest.

Deep energy-based models and direct regression. The most direct learning algorithm is to first maximize the sum of log-likelihoods $\mathcal{L}(\theta; \mathcal{D}^1, \ldots, \mathcal{D}^t) := \sum_{i=1}^{t} \sum_{j=1}^{n_i} \log p_\theta(x^{i(j)}; \sigma^i)$, where n_i is the number of samples in the ith regime. The parameterization of the model is indexed by a vector θ, which defines $p(\cdot)$ as $\log p_\theta(x; \sigma) := \sum_{k=1}^{l} \phi_{\theta_{k,\sigma_{F_k}}}(x_{S_k}) + \text{constant}$. Here, $\phi_\theta(\cdot)$ is a differentiable black-box function, which in our experiments is a MLP (aka a multilayer perceptron, or feedforward neural network). Parameter vector $\theta_{k,\sigma_{F_k}}$ is the collection of weights and biases of the MLP, a different instance for each factor k *and* combination of values in σ_{F_k}. In principle, making the parameters smooth functions of σ_{F_k} is possible, but in the interest of simplifying the presentation, we instead use a look-up table for completely independent parameters as indexed by the possible values of σ_{F_k}. Parameter set θ is the union of all $\sum_{k=1}^{l} \prod_{j \in F_k} |\aleph_j|$ MLP parameter sets. As maximizing log-likelihood is generally intractable, in our implementation we apply pseudo-likelihood with discretization of each variable X in a grid. The level of discretization does not affect the number of parameters, as we take their numerical value as is, renormalizing over the pre-defined grid. Score matching [38], noise contrastive estimation [32] or other variants (e.g., [71]) could be used; our pipeline is agnostic to this choice, with further details in Appendix F. We then estimate $f_y(x) := \mathbb{E}[Y \mid x]$ using an off-the-shelf method, which in our case is another MLP. For a given $\sigma^\star$, we sample from the corresponding $p(x; \sigma^\star)$ with Gibbs sampling and average the results of the regression estimate $\widehat{f}_y(x)$ to obtain an estimate $\widehat{\mu}(\sigma^\star)$.

Inverse probability weighting (IPW). A more direct method is to reweight each training sample by the target distribution $p(x; \sigma^\star)$ to generate $\widehat{\mu}(\sigma^{i\star}) := \sum_{j=1}^{n_i} y^{i(j)} w^{i(j)^\star}$, where $w^{i(j)^\star}$ is the density ratio $p(x^{i(j)}; \sigma^\star)/p(x^{i(j)}; \sigma^i)$, rescaled such that $\sum_{j=1}^{n_i} w^{i(j)^\star} = 1$. There are several direct methods for density ratio estimation [53] that could be combined using the messages/product-ratios of the previous section, but we found that it was stabler to just take the density ratio of the fitted models using deep energy-based learning, just like in the previous algorithm. Once estimators $\widehat{\mu}(\sigma^{\star(1)}), \ldots, \widehat{\mu}(\sigma^{\star(t)})$ are obtained, we combine them by the usual inverse variance weighting rule, $\widehat{\mu}(\sigma^{i\star}) := \sum_{i=1}^{t} r^i \mu_{\sigma^{\star(i)}} / \sum_{i=1}^{t} r^i$, where $r^i := 1/\widehat{v}^i$, and $\widehat{v}^i := \sum_{j=1}^{n_i} (y^{i(j)})^2 (w^{i(j)^\star})^2$. This method requires neither a model for $f_y(x)$ nor Markov chain Monte Carlo. However, it may behave more erratically than the direct method described above, particularly under strong shifts in distribution.

Covariate shift regression. Finally, a third approach for estimating $\mu(\sigma^\star)$ is to combine models for $f_y(x)$ with density ratios, learning a customized $\widehat{f}_y(x)$ for each test regime separately. A s this is very slow and did not appear to be advantageous compared to the direct method, we defer a more complete description to Appendix E.

Predictive Coverage. Even when treatment effects are identifiable within the IFM framework, the uncertainty of resulting estimates can vary widely depending on the training data and learning algorithm. Building on recent work in conformal inference [78, 51, 76], we derive the following finite sample coverage guarantee for potential outcomes, which requires no extra assumptions beyond those stated above.

Theorem 4.1 (Predictive Coverage.) *Assume the identifiability conditions of Theorems 3.1 or 3.2 hold. Fix a target level $\alpha \in (0, 1)$, and let $\mathcal{I}_1, \mathcal{I}_2$ be a random partition of observed regimes intro training and test sets of size $n/2$. Fit a model $\widehat{\mu}$ using data from $\mathcal{I}_1$ and compute conformity scores $s^{(i)} = |y^{k(i)} - \widehat{\mu}(\sigma^k)|$ using data from $\mathcal{I}_2$. For some new test environment $\sigma^\star$, compute the normalized likelihood ratio $w^{(i)}(\sigma^\star) \propto p(x^{k(i)}; \sigma^\star)/p(x^{k(i)}; \sigma^k)$, rescaled to sum to $n/2$. Let $\widehat{\tau}(\sigma^\star)$ be the q^{th} smallest value in the reweighted empirical distribution $\sum_i w^{(i)}(\sigma^\star) \cdot \delta(s^{(i)})$, where δ denotes the Dirac delta function and $q = \lceil (n/2 + 1)(1 - \alpha) \rceil$.*
Then for any new sample $n + 1$, we have:

$$\mathbb{P}\big(Y^{\star(n+1)} \in \widehat{\mu}(\sigma^\star) \pm \widehat{\tau}(\sigma^*)\big) \geq 1 - \alpha.$$

Moreover, if weighted conformity scores have a continuous joint distribution, then the upper bound on this probability is $1 - \alpha + 1/(n/2 + 1)$.

5 Experiments

We run a number of semi-synthetic experiments to evaluate the performance of the IFM approach on a range of intervention generalization tasks. In this section, we summarize our experimental set-up and main results. The code for reproducing all results and figures is available online[8]; in Appendix G, we provide a detailed description of the datasets and models; and in Appendix H we present further analysis and results.

Datasets. Our experiments are based on the following two biomolecular datasets: *i)* **Sachs** [66]: a cellular signaling network with 11 nodes representing phosphorylated proteins and phospholipids, several of which were perturbed with targeted reagents to stimulate or inhibit expression. There are 4 binary intervention variables. Σ_{train} has 5 regimes: a *baseline*, with all $\sigma_i = 0$, and 4 "single-intervention" regimes, each with a different single $\sigma_i = 1$. Σ_{test} consist of the remaining 11 $(= 2^4 - 5)$ unseen regimes. *ii)* **DREAM** [30]: Simulated data based on a known *E. coli* regulatory sub-network with 10 nodes. There are 10 binary intervention variables, and (similarly) Σ_{train} has a baseline regime, with all $\sigma_i = 0$, and 10 single-intervention regimes, with single choices of $\sigma_i = 1$. Σ_{test} consists of 45 $(10 \times 9/2 = 45)$ unseen regimes defined by all pairs $\sigma_i = \sigma_j = 1$.

Oracular simulators. The first step to test intervention generalization is to build a set of proper test beds (i.e., simulators that serve as causal effect oracles for any $\sigma \in \Sigma$), motivated by expert knowledge about the underlying system dynamics. Neither the original Sachs data nor the DREAM simulator we used provide joint intervention data required in our evaluation. Thus, for each domain, we trained two ground truth simulators: *i)* **Causal-DAG**: a DAG model following the DAG structure and data provided by the original Sachs et al. and DREAM sources (DAGs shown in Fig. 10(**a**)) and Fig. 11(**a**), respectively). Given the DAG, we fit a model where each conditional distribution is a heteroskedastic Gaussian with mean and variance parameterized by MLPs (with 10 hidden units) of the respective parents. *ii)* **Causal-IFMs**: the corresponding IFM is obtain by a direct projection of the postulated DAG factors (as done in, e.g., Fig. 2(**b**)). The likelihood is a neural energy model (Section 4) with MLPs with 15 hidden units defining potential functions. After fitting these models, we compute by Monte Carlo simulation their implied ground truths for every choice of σ^*. The outcomes Y are then generated under 100 different structural equations of the form $\tanh(\lambda^\top X) + \epsilon$, with random independent normal weights λ and $\epsilon \sim \mathcal{N}(0, v_y)$. λ and v_y are scaled such that the ground truth variance of $\lambda^\mathsf{T} X$, $\text{Var}(\lambda^\mathsf{T} X)$, is sampled uniformly at random from the interval $[0.6, 0.8]$, and set $v_y := 1 - \text{Var}(\lambda^\mathsf{T} X)$.

Compared models. We implement three variants of our proposed IFM model, corresponding to the three learning algorithms described in Section 4: *i)* **IFM1** uses deep energy-based models and direct regression; *ii)* **IFM2** uses an IPW estimator; and *iii)* **IFM3** relies on covariate shift regression. We compare these models to the following benchmarks: *i)* **Black-box**: We apply an off-the-shelf algorithm (XGBoost [14]) to learn a direct mapping from σ to Y without using X. This model does not exploit any structural assumptions. *ii)* **DAG**: We estimate the structural equations in an acyclic topological ordering that is consistent with the data generating process. The likelihood is defined by conditional Gaussian models with mean and variances parameterized as MLPs, matching exactly one of the simulators described below. This gives this competitor much advantage in the benchmarks generated by DAGs, as following a parametric Gaussian with additive error structure is substantive information to be exploited.

Results. We evaluate model performance based on the proportional root mean squared error (pRMSE), defined as the average of the squared difference between the ground truth Y and estimated $\hat{Y}$, with each entry further divided by the ground truth variance of the corresponding Y. Results are visualized in Fig. 5. We additionally run a series of one-sided binomial tests to determine whether models significantly outperform the black box baseline, and compare the Spearman's rank correlation [80] between expected and observed outcomes for all test regimes. Unsurprisingly, DAGs do best when the ground truth is a Causal-DAG, while IFM methods do better when the data generator is a Causal-IFM. Still, some IFMs are robust to both ground truth models, with IFM1 (the deep energy model) doing especially well on the DREAM dataset, and IFM2 (the IPW estimator) excelling on the Sachs data.

[8] https://github.com/rbas-ucl/intgen

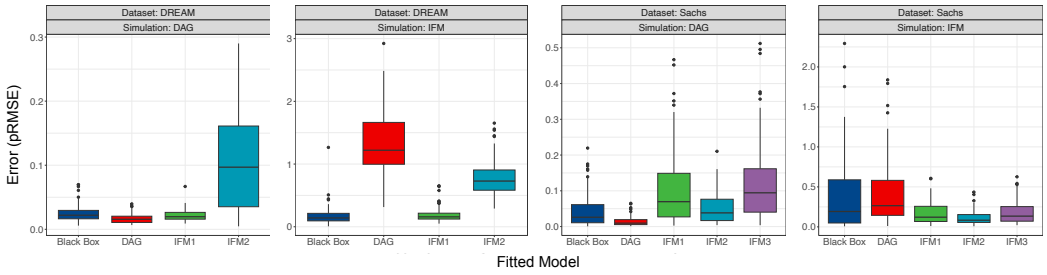


Figure 5: Experimental results on a range of intervention generalization tasks, see text for details.

In particular, for the DREAM datasets, IFM1 (scaled) errors remain stable, while the DAG has a major loss of performance when a non-DAG ground truth is presented. However, IFM2 underperform on DREAM datasets, which might be because the knockdowns regimes in those datasets induce dramatic shifts in distribution overlap among regimes. (We omit results for IFM3 (covariate shift regression) on these datasets, as the method does not converge in a reasonable time.) Though the black-box method (XGBoost algorithm and no structural assumption) sometimes struggles to extrapolate, displaying a long-tailed distribution of errors, it actually does surprisingly well in some experiments, especially on the DREAM dataset. In fact, the black-box method is essentially indistinguishable from linear regression in this benchmark (results not shown), which is not surprising given the sparsity of the training data. However, the non-additivity of the σ effect on Y is more prominent in the Sachs datasets, making it difficult to generalize without structural assumptions.

6 Related Work

A factor graph interpretation of Pearl's *do*-calculus framework is described in [79], but without addressing the problem of identifiability. More generally, several authors have exploit structural assumptions to predict the effects of unseen treatments. Much of this research falls under the framework of *transportability* [61, 7], where the goal is to identify causal estimands from a combination of observational and experimental data collected under different regimes. If only atomic interventions are considered, the *do*-calculus is sound and complete for this task [50]. The σ-calculus introduced by [28, 17] extends these transportability results to soft interventions [18, 19]. However, these methods are less clearly defined when an intervention affects several variables simultaneously, and while DAGs with additive errors have some identifiability [67] and estimation [75] advantages, acyclic error additivity may not be an appropriate assumption in some domains. Under further parametric assumptions, causal effects can be imputed with matrix completion techniques [73, 3] or more generic supervised learning approaches [31], but these methods often require some data to be collected for all regimes in Σ_{test}. Finally, [2] is the closest related work in terms of goals, factorizing the function space of each $\mathbb{E}[X_i; \sigma]$ directly as a function of σ.

Another strand of related research pertains to online learning settings. For instance, several works have shown that causal information can boost convergence rates in multi-armed bandit problems when dependencies are present between arms [45, 49, 24], even when these structures must themselves be adaptively learned [9]. This suggests a promising direction for future work, where identification strategies based on the IFM framework are used to prioritize the search for optimal treatments. Indeed, combining samples across multiple regimes can be an effective strategy for causal discovery, as illustrated by recent advances in invariant causal prediction [63, 33, 64, 81], and it can also help with domain adaptation and covariate shift [54, 6, 13, 15].

7 Conclusion

We introduced the IFM framework for solving intervention generalization tasks. Results from simulations calibrated by real-world data show that our method successfully predicts outcomes for novel treatments, providing practitioners with new methods for conducting synthetic experiments. Future work includes: *i)* integrating the approach with experimental design, Bayesian optimization and bandit learning; *ii)* variations that include pre-treatment variables and generalizations across heterogeneous subpopulations, inspired by complementary matrix factorization methods such as [2]; *iii)* tackling sequential treatments; and *iv)* diagnostics of cross-regime overlap issues [36, 58].

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and Mathias Drton for useful discussions. GBH was supported by the ONR grant 62909-19-1-2096, and JY was supported by the EPSRC grant EP/W024330/1. RS was partially supported by both grants. JZ was supported by UKRI grant EP/S021566/1.

References

[1] Abbeel, P., Koller, D., and Ng, A. Y. (2006). Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788.

[2] Agarwal, A., Agarwal, A., and Vijaykumar, S. (2023). Synthetic combinations: A causal inference framework for combinatorial interventions. *arXiv* preprint, 2303.14226.

[3] Agarwal, A., Shah, D., and Shen, D. (2020). Synthetic A/B testing using synthetic interventions. *arXiv* preprint, 2006.07691.

[4] Aglietti, V., Lu, X., Paleyes, A., and González, J. (2020). Causal Bayesian optimization. 23^{rd} *International Conference on Artificial Intelligence and Statistics*.

[5] Akbari, K., Winter, S., and Tomko, M. (2023). Spatial causality: A systematic review on spatial causal inference. *Geographical Analysis*, 55:56–89.

[6] Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2020). Invariant risk minimization. *arXiv* preprint, 1907.02893.

[7] Bareinboim, E. and Pearl, J. (2014). Transportability from multiple environments with limited experiments: Completeness results. In *Advances in Neural Information Processing Systems*, volume 27.

[8] Bica, I., Alaa, A. M., Lambert, C., and van der Schaar, M. (2021). From real-world patient data to individualized treatment effects using machine learning: Current and future methods to address underlying challenges. *Clinical Pharmacology & Therapeutics*, 109(1):87–100.

[9] Bilodeau, B., Wang, L., and Roy, D. M. (2022). Adaptively exploiting d-separators with causal bandits. In *Advances in Neural Information Processing Systems*.

[10] Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1974). *Discrete Multivariate Analysis: Theory and Practice*. MIT Press.

[11] Blom, T. and Mooij, J. M. (2023). Causality and independence in perfectly adapted dynamical systems. *Journal of Causal Inference*, 11(1):2885–2915.

[12] Bongers, S., Forré, P., Peters, J., and Mooij, J. M. (2021). Foundations of structural causal models with cycles and latent variables. *Annals of Statistics*, 49(5):2885–2915.

[13] Bühlmann, P. (2020). Invariance, Causality and Robustness. *Statistical Science*, 35(3):404 – 426.

[14] Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794.

[15] Chen, Y. and Bühlmann, P. (2021). Domain adaptation under structural causal models. *J. Mach. Learn. Res.*, 22(1).

[16] Constantinou, P. and Dawid, A. P. (2017). Extended conditional independence and applications in causal inference. *The Annals of Statistics*, 45(6):2618–2653.

[17] Correa, J. and Bareinboim, E. (2020a). A calculus for stochastic interventions:causal effect identification and surrogate experiments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10093–10100.

[18] Correa, J. and Bareinboim, E. (2020b). General transportability of soft interventions: Completeness results. In *Advances in Neural Information Processing Systems*, volume 33, pages 10902–10912.

[19] Correa, J., Lee, S., and Bareinboim, E. (2022). Counterfactual transportability: A formal approach. In *International Conference on Machine Learning*.

[20] Cowell, R., Dawid, A., Lauritzen, S., and Spiegelhalter, D. (1999). *Probabilistic Networks and Expert Systems*. Springer-Verlag.

[21] Dash, D. (2005). Restructuring dynamic causal systems in equilibrium. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 81–88.

[22] Dawid, A. P. (2010). Beware of the DAG! *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008, PMLR*, 6:59–86.

[23] Dawid, P. (2021). Decision-theoretic foundations for statistical causality. *Journal of Causal Inference*, 9(56):39–77.

[24] de Kroon, A. A. W. M., Belgrave, D., and Mooij, J. M. (2020). Causal discovery for causal bandits utilizing separating sets. *arXiv*:2009.07916.

[25] Drton, M. (2009). Discrete chain graph models. *Bernoulli*, 15:736–753.

[26] Eaton, D. and Murphy, K. (2007). Exact Bayesian structure learning from uncertain interventions. *Artificial Intelligence & Statistics*.

[27] Eberhardt, F. and Scheines, R. (2007). Interventions and causal inference. *Philosophy of Science*, 74:981–989.

[28] Forré, P. and Mooij, J. M. (2018). Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders. In *Proceedings of the 34th Annual Conference on Uncertainty in Artificial Intelligence*, pages 269–278.

[29] Gentzel, A. M., Pruthi, P., and Jensen, D. (2021). How and why to use experimental data to evaluate methods for observational causal inference. In *International Conference on Machine Learning*, pages 3660–3671. PMLR.

[30] Greenfield, A., Madar, A., Ostrer, H., and Bonneau, R. (2010). Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397.

[31] Gultchin, L., Watson, D., Kusner, M., and Silva, R. (2021). Operationalizing complex causes: A pragmatic view of mediation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3875–3885. PMLR.

[32] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 297–304.

[33] Heinze-Deml, C., Peters, J., and Meinshausen, N. (2018). Invariant Causal Prediction for Nonlinear Models. *J. Causal Inference*, 6(2).

[34] Hernán, M. and J, R. (2020). *Causal Inference: What If*. Chapman & Hall/CRC.

[35] Higbee, S. (2023). Policy learning with new treatments. *arXiv:2210.04703*.

[36] Hill, J. and Su, Y. (2011). Assessing lack of common support in causal inference using bayesian nonparametrics: Implications for evaluating the effect of breastfeeding on children's cognitive outcomes. *The Annals of Applied Statistics*, 7:1386–1420.

[37] Hoover, K. (2001). *Causality in Macroeconomics*. Cambridge University Press.

[38] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709.

[39] Hyvärinen, A., Shimizu, S., and Hoyer, P. O. (2008). Causal modelling combining instantaneous and lagged effects: an identifiable model based on non-Gaussianity. *Proceedings of the 25th international conference on machine learning (ICML 2008)*, pages 424–431.

[40] Iwasaki, Y. and Simon, H. A. (1994). Artificial intelligence. *Causality and model abstraction*, 67:143–194.

[41] Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780.

[42] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[43] Korb, K. B., Hope, L. R., Nicholson, A. E., and Axnick, K. (2004). Varieties of causal intervention. *Pacific Rim International Conference on Artificial Intelligence (PRICAI 2004)*, pages 322–331.

[44] Kschischang, F., Frey, B., Brendan, J., and Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498—519.

[45] Lattimore, F., Lattimore, T., and Reid, M. D. (2016). Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems*, pages 1181–1189.

[46] Lauritzen, S. (1996). *Graphical Models*. Oxford University Press.

[47] Lauritzen, S. L. and Richardson, T. S. (2002). Chain graph models and their causal interpretation. *Journal of the Royal Statistical Society Series B*, 64:321–361.

[48] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. J. (2006). A tutorial on energy-based learning. In BakIr, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S., editors, *Predicting Structured Data*. MIT Press.

[49] Lee, S. and Bareinboim, E. (2018). Structural causal bandits: Where to intervene? In *Advances in Neural Information Processing Systems*, pages 2568–2578.

[50] Lee, S., Correa, J., and Bareinboim, E. (2020). General transportability – synthesizing observations and experiments from heterogeneous domains. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10210–10217.

[51] Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

[52] Leist, A. K., Klee, M., Kim, J. H., Rehkopf, D. H., Bordas, S. P., Muniz-Terrera, G., and Wade, S. (2022). Mapping of machine learning approaches for description, prediction, and causal inference in the social and health sciences. *Science Advances*, 8(42):eabk1942.

[53] Lu, N., Zhang, T., Fang, T., Teshima, T., and Sugiyama, M. (2023). Rethinking importance weighting for transfer learning. In *Federated and Transfer Learning*, pages 185–231. Springer.

[54] Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. (2018). Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, volume 31.

[55] Malinsky, D. (2018). Intervening on structure. *Synthese*, 195:2295—-2312.

[56] McKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

[57] Mogensen, S. W., Malinsky, D., and Hansen, N. R. (2018). Causal learning for partially observed stochastic dynamical systems. *Proceedings of the 34th conference on Uncertainty in Artificial Intelligence (UAI 2018)*, pages 350–360.

[58] Oberst, M., Johansson, F., Wei, D., Gao, T., Brat, G., Sontaga, D., and Varshney, K. (2020). Characterization of overlap in observational studies. 23^{rd} *International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, pages 788–798.

[59] Ogburn, E. L., Shpitser, I., and Lee, Y. (2020). Causal inference, social networks and chain graphs. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 183:1659—-1676.

[60] Pearl, J. (2009). *Causality: Models, Reasoning and Inference, 2nd edition*. Cambridge University Press.

[61] Pearl, J. and Bareinboim, E. (2011). Transportability of causal and statistical relations: A formal approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):247–254.

[62] Pearl, J. and McKenzie, D. (2018). *The Book of Why*. Allen Lane.

[63] Peters, J., Bühlmann, P., and Meinshausen, N. (2016). Causal inference by using invariant prediction: identification and confidence intervals. *J. Royal Stat. Soc. Ser. B Methodol.*, 78(5):947–1012.

[64] Pfister, N., Bühlmann, P., and Peters, J. (2019). Invariant causal prediction for sequential data. *Journal of the American Statistical Association*, 114(527):1264–1276.

[65] Richardson, T. (2003). Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30:145–157.

[66] Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

[67] Saengkyongam, S. and Silva, R. (2020). Learning joint nonlinear effects from single-variable interventions in the presence of hidden confounders. 36^{th} *Conference on Uncertainty in Artificial Intelligence (UAI 2020)*.

[68] Sejdinovic, D., Gretton, A., and Bergsma, W. (2013). A kernel test for three-variable interactions. *Neural Information Processing Systems (NeurIPS)*, 26:1124—-1132.

[69] Shi, C., Sridhar, D., Misra, V., and Blei, D. (2022). On the assumptions of synthetic control methods. In *International Conference on Artificial Intelligence and Statistics*, pages 7163–7175. PMLR.

[70] Shpitser, I. and TchetgenTchetgen, E. (2016). Causal inference with a graphical hierarchy of interventions. *Annals of Statistics*, 44:2433–2466.

[71] Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*, 32.

[72] Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction and Search*. Cambridge University Press.

[73] Squires, C., Shen, D., Agarwal, A., Shah, D., and Uhler, C. (2022). Causal imputation via synthetic interventions. In *Conference on Causal Learning and Reasoning*, pages 688–711. PMLR.

[74] Studený, M. (2051). *Probabilistic conditional independence structures*. Springer.

[75] Sussex, S., Makarova, A., and Krause, A. (2023). Model-based causal Bayesian optimization. 11^{th} *International Conference on Learning Representations*.

[76] Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. (2019). Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32.

[77] Tigas, P., Annadani, Y., Jesson, A., Schölkopf, B., Gal, Y., and Bauer, S. (2022). Interventions, where and how? Experimental design for causal models at scale. In *Neural Information Processing Systems (NeurIPS 2022)*.

[78] Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer, New York.

[79] Winn, J. (2012). Causality with gates. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, PMLR*, 22:1314–1322.

[80] Zar, J. H. (2014). Spearman rank correlation: overview. *Wiley StatsRef: Statistics Reference Online*.

[81] Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., and Precup, D. (2020). Invariant causal prediction for block MDPs. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 11214–11224. PMLR.

A A Guide to Structure Elicitation and Learning

This section starts with a discussion particularly aimed at methodologists and practitioners who may not feel entirely at ease using a factor graph approach to model causality. We then conclude with a brief discussion on how to build IFMs by knowledge elicitation and structure learning. For further discussion on the use of alternative independence models in causality, we recommend Section 11 of [22]. Section 12 of the same paper provides further discussion of the meaning of linking intervention variables to random variables in statistical modeling and its DAG interpretations.

A.1 Background: Extended Conditional Independence

Our starting point is Dawid's framework for causal inference [23]. Although dubbed "decision-theoretical", no utility optimization is actually necessary to justify its approach for encoding causal assumptions and deriving their consequences. As such, we will drop the "decision-theoretical" label and refer to it as the *extended conditional independence* (ECI) approach, formalized in [16]. The main point is that causal assumptions, either given in the form of independencies in a mutilated DAG or in distributions of potential outcomes, can directly be framed as "mere" statements of conditional independence among random variables and intervention variables.

As the intervention variables σ_i are not random variables, this calls for a clarification of what it means to claim

$$X_i \perp\!\!\!\perp \sigma_1 \mid X_j, \sigma_2$$

For what follows, it suffices to interpret this statement as X_i not changing in distribution across different values of σ_1 when σ_2 and X_j are fixed/observed at any particular value.

The most basic statement of structure is expressed by a graph $X \to Y$. This graph is intended to communicate that, if we intervene on X, the distribution of Y may change; but if we intervene on Y, the distribution of X does not change. This may feel unsatisfactory, as the graph $X \to Y$ by itself does not communicate any independence constraint.[9]

The ECI approach is explicit: we take as primitive the notion of "intervention on X" and "intervention on Y" operationalized by a pair of intervention variables σ_x and σ_y such that

$$\begin{aligned} Y &\perp\!\!\!\perp \sigma_x \mid X \\ X &\perp\!\!\!\perp \sigma_y \end{aligned} \tag{5}$$

The first independence establishes that once I know *which* value X took, it does not matter *how* X came to be (i.e., the value of σ_x) [22]. This is more commonly described as "lack of unmeasured confounding between X and Y" or *ignorability*.

The second independence establish that *how* Y comes to be does not change the distribution of X. This is more commonly described as "Y does not cause X". Note that the model does not explicit state that "X causes Y", just that it's *allowed* to. This mirrors the typical interpretation of a graphical model, by which a graph does not imply *dependencies*. Instead, a graph is defined by its *independencies* [46].

Now, why do we feel compelled to write the edge $X \to Y$, as in Fig. 6**(a)**, as well as the directions from (σ_x, σ_y) to (X, Y)? This is because, among all "canonical graphical models"[10], *that's the only option that we have*. To understand that, consider the three variations in Fig. 6**(b)-(d)**: each one of them violates one or both of the relations encoded in Eq. (5). That is the sense in which the DAG is justified here: syntactic sugar for Eq. (5). This is particularly emphasized by Fig. 6**(e)**: if we do not define σ_y, the model is defined by the sole constraint $Y \perp\!\!\!\perp \sigma_x \mid X$. In that case, a chain graph with undirected component $X - Y$ suffices (and so does the purely undirected structured). Any arrows here are for cosmetic purposes.

The same idea applies to *conditional ignorability*: if we have some set of covariates Z such that $Y \perp\!\!\!\perp \sigma_x \mid X, Z$, this allows us, for instance, to learn causal effects from observational data. That is, if

[9] The graph may suggest a factorization, and the causal ordering implied by the factorization does matter for models such as the additive error model [63]. However, since these graphs are primarily models of independence, something is still amiss here.

[10] This means the usual machinery of directed, undirected (Markov), mixed and chain graph models [46, 65], a relatively small but highly interpretable corner in the universe of possible independence models [74].

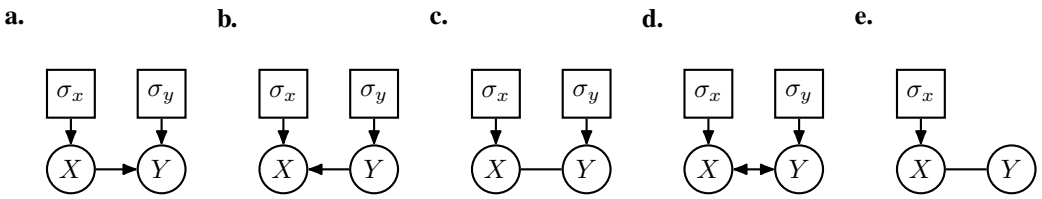


Figure 6: The meaning of $X \to Y$. **(a)** Expressing it when interventions on both X and Y are defined via variables σ_x and σ_y, and we wish to express that $Y \perp\!\!\!\perp \sigma_x \mid X$ and $X \perp\!\!\!\perp \sigma_y$. **(b)-(d)** None of these graphs respect the two independence constraints on interest. **(e)** If σ_y is not defined, a graph like this one suffices to encode the remaining constraint $Y \perp\!\!\!\perp \sigma_x \mid X$.

$\sigma_x \in \{$"do nothing", "$do(X = 0)$", "$do(X = 1)$"$\}$, we can obtain $p(y \mid z; \sigma_x = do(X = x))$ from observational data as

$$\begin{array}{lcll} p(y \mid z; \sigma_x = do(X = x)) & = & p(y \mid x, z; \sigma_x = do(X = x)) & \text{(since } do(X = x) \Rightarrow X = x\text{)} \\ & = & p(y \mid x, z; \sigma_x = \text{do nothing}) & \text{(since } Y \perp\!\!\!\perp \sigma_x \mid X, Z\text{)} \\ & = & p(y \mid x, z) & \text{(conventional notation)} \end{array}$$

The first line exploits a feature of atomic interventions ($do(X = x) \Rightarrow X = x$) which is not generally available for other types of intervention. That is why non-atomic interventions cannot directly be handled by Pearl's *do*-calculus [17].

As a further example, average treatment effects can be obtained from the backdoor formula/g-formula [60, 34], which further requires "Z not to be caused by X". In the ECI framework[11], this is just the further requirement that $\sigma_x \perp\!\!\!\perp Z$ (under the interpretation that σ_x has "no causes" because it comes from a hypothetical agent "outside the system that generates the data", this marginal independence can only be explained by σ_x "not causing" Z):

$$\begin{array}{lcl} p(y; \sigma_x = do(X = x)) & = & \sum_z p(y \mid z; \sigma_x = do(X = x)) p(z; \sigma_x = do(X = x)) \\ & & \text{(standard marginalization)} \\ & = & \sum_z p(y \mid x, z; \sigma_x = do(X = x)) p(z) \\ & & \text{(since } Z \perp\!\!\!\perp \sigma_x \text{ and } do(X = x) \Rightarrow X = x\text{)} \\ & = & \sum_z p(y \mid x, z; \sigma_x = \text{do nothing}) p(z) \\ & & \text{(since } Y \perp\!\!\!\perp \sigma_x \mid X, Z\text{)} \\ & = & \sum_z p(y \mid x, z) p(z) \\ & & \text{(conventional notation)} \end{array}$$

As a matter of fact, the classical axiom of *consistency* that underpins causal reasoning from potential outcomes [34] (basically, that the potential outcome of Y under intervention $do(X = x)$ should match the observed outcome Y if $X = x$ i.e. $Y_x = Y$ if $X = x$) can be interpreted as the lack of "fat-hand" interventions [27], i.e., that there is some conditioning set C such that $\sigma_x \perp\!\!\!\perp Y \mid C$ for any random variable Y other than X.

A.2 Graphical Models and the IFM

The above discussion indicates that conditional independencies among intervention variables and random variables are the building blocks of a (counterfactual-free, or "Rung 2"[62]) causal modeling language. However, we need more than that for any practical way of encoding assumptions, as many reasonable models will involve an extremely large number of conditional independencies. For instance, even a simple directed Markov chain $X_1 \to X_2 \to \cdots \to X_p$ involves a super-exponential number of conditional independencies (e.g., X_p is conditionally independence of X_1 given any non-empty subset of $\{X_2, \ldots, X_{p-1}\}$).

Graphical models are extremely useful families of independence models that allow for the use of a relatively small number of *local* Markov conditions to describe *global* Markov conditions. Moreover,

[11] To be clear, the idea of using explicit intervention variables to describe the backdoor adjustment dates back at least to the original graphical formulation of [72]. The proof in [60] also relies on explicit regime variables. ECI formalizes explicitly the role of conditional independence statements that involve non-random variables.

when compared to relying on algebra alone, symbolic algorithms based on graph-theoretical concepts provide ways to derive such implications in a easier and more transparent manner. This explains the popularity of graphical models in causal modeling, regardless of the cosmetic appeal of drawing edges.

In what follows, we will start by contrasting undirected ("Markov") networks to DAGs, as factor graph models encode the very same families of independencies as undirected graphs (with the extra facility of representing low-order interactions on top of independence constraints). For instance, in a DAG, the local Markov condition is a variable being conditional independent of its non-descendants given its parents; the global Markov condition is anything entailed by d-separation [60, 72]. See [46] for many classical results.

Creating an independence (graphical) model requires trade-offs, as not every family of conditional independencies can be easily cast in graph-theoretical terms [74]. One common example is the (chordless) simple cycle of length 4: the independencies encoded in this undirected graph $X_1 - X_2 - X_3 - X_4 - X_1$ cannot be represented by any DAG, even one with the same adjacencies, such as, $X_1 \to X_2 \to X_3 \leftarrow X_4 \leftarrow X_1$. (The converse is also true, the independencies encoded by this "diamond structure" DAG have no correspondence to any undirected graph.) It is out of our scope to get into any of the fine details of such differences, see [46] for details. Instead, we will focus on some broad aspects relevant to causal inference.

DAGs (with or without hidden variables) are by far the most common type of graphical model for expressing causality. There are different ways of explaining this appeal, of which we highlight *"marginal independencies"* and *"explaining away"*.

Marginal Independencies. A connected undirected graph implies no marginal independendencies. Yet, marginal independencies lie behind the claim that "the future does not cause the past". This is illustrated in Fig. 6**(a)** by interpreting Y as encoding events that happen after the events encoded by X, and hence an assumption of $X \perp\!\!\!\perp \sigma_y$ is desirable.

Explaining Away. This well-known phenomenon is illustrated by independencies that get destroyed by conditioning upon further evidence. For example, if $X_3 = f(X_1, X_2)$, then even if $X_1 \perp\!\!\!\perp X_2$, it is clear that knowing also the value of X_3 will change the support of X_1 given X_2. More generally, $p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 \mid x_1, x_2)$ means that $p(x_1, x_2) = g(x_1)h(x_2)$ for some $g(\cdot), h(\cdot)$, but $p(x_1, x_2 \mid x_3) \propto g(x_1)h(x_2)k_{x_3}(x_1, x_2)$ for some $k_{x_3}(\cdot, \cdot)$ which does not factorize in general. This also applies to combinations of random variables and intervention variables. If "X does not cause Z" and we operationalize that by $\sigma_x \perp\!\!\!\perp Z$ in the DAG $\sigma_x \to X \leftarrow Z$, it is possible to have this independence destroyed by conditioning on X, since $p(z \mid x; \sigma_x)$ will in general be different if σ_x is "do nothing" (where Z is allowed to vary with different values of X) compared against $\sigma_x = do(X = x)$ (where Z is independent of X). That is, in this example we have $Z \not\perp\!\!\!\perp \sigma_x \mid X$, in the sense that $p(z \mid x; \sigma_x)$ is a non-trivial function of σ_x even if $\sigma_x \perp\!\!\!\perp Z$.

We claim that neither of the two properties above are particularly well-motivated in the snapshot sampling process illustrated in Figure 1. Given the snapshot, some explaining away could indeed happen between an intervention variable σ and pre-treatment/"Past" variables (Fig. 1**(b)**) that happen to be recorded. However, this can be simply accounted for by including these pre-treatment variables as conditioning variables within the IFM factors.

For longitudinal studies, where marginal independencies do matter ("the future doesn't cause the past") *and* we do happen to make multiple snapshots (as opposed to "contemporaneous DAG structures", as in [39]), it is not a technical challenge to extent the IFM to a chain graph structure [47] with factor graph undirected components. We leave this development for future work.

To summarize, if marginal independencies and explaining away are not of particular relevance to the problem at hand, then we recommend the IFM as a family of independence models, particularly in light of its very simple local Markov condition (i.e., in the corresponding undirected graph implied by the factor structure, "a variable is independent of its non-neighbors given its neighbors"). This comes to life especially in models motivated by equilibrium equations. For instance, consider the following example from Section 3.1.1 of [11]:

$$\begin{aligned} f_I: &\quad X_I - U_I = 0 \\ f_D: &\quad U_1(X_I - X_O) = 0 \\ f_P: &\quad U_2(gU_3X_D - X_P) = 0 \\ f_O: &\quad U_4(U_5I_KX_P - X_O) = 0 \end{aligned}$$

where f denotes differential equations at equilibrium, X denotes observed random variables, U denotes (mutually independent) latent variables, and I denotes an intervention indicator.

After marginalizing the U variables, we "get" an IFM (although, this is not the whole story, as other non-graphical constraints may take place given the particular equations. [11] discusses U_I being marginally independent of I_K on top of the above). In general, energy-based models are to be interpreted as conjunctions of soft-constraints, the factor graph is one implication of a system of stochastic differential equations (SDE), and interventions denote change to particular constraints.

A SDE model can indeed have *other* assumptions on top of the factorization, and parameters which carry particular meaningful interpretations. Nonetheless, this fits well with our claims that the IFM is a "minimalist" family of models in terms of structural assumptions — a reasonably conservative direction to follow; particularly, when the dynamics of many natural phenomena cannot be (currently) measured at individual level, as it's the case of much of cell biology data, and hence writing down a full SDE model may be more inspirational than scientifically grounded. See [47] for further elaborations on some of these ideas.

It is worth highlighting that, as long as the timing of the measurements is well-defined and consistently respected by the real-world sampling procedure, there is no need to wait for a system to get into equilibrium in order for an IFM to be applicable. [21] discusses how "causal structure" changes depending on which equations have reached equilibrium at any particular point in time — after all, the process as a function of time is non-stationary until (and if) it reaches equilibrium. Importantly, intervention generalization here should *not* be interpreted as extrapolating to different, unsampled, time points in a non-stationary process.

Finally, we note that the ECI framework as described by [16] is relatively restricted in the definition of statements such as

$$\sigma_i \perp\!\!\!\perp \sigma_j \mid X_r, \sigma_s,$$

that is, claims of independence between sets of intervention variables. The focus there is on *variation independence*, which roughly speaking can be interpreted as the range of possible values for σ_i not depending on σ_j.[12]

In contrast, we interpret statements

$$\sigma_i \perp\!\!\!\perp \sigma_j \mid X, \sigma_{\setminus ij},$$

where $\sigma_{\setminus ij}$ are all intervention indicators other than σ_i and σ_j, by merely linking it to the factorization

$$p(x; \sigma_i, \sigma_j, \sigma_{\setminus ij}) \propto g(x, \sigma_i, \sigma_{\setminus ij}) h(x, \sigma_j, \sigma_{\setminus ij}),$$

for some functions $g(\cdot)$ and $h(\cdot)$. Going from pairwise independence to setwise independence is defined here by the usual graphical criterion of pairwise independence for the product space of the sets implying setwise independence.

The above does not mean a genuine factorization of $p(x; \sigma_i, \sigma_j, \sigma_{\setminus ij})$ as a function of σ, as the normalizing constant will in general depend on all regime indicators (but not on the data). It however denotes the difference between Fig. 2**(b)** and Fig. 2**(c)**: the latter is a chain graph with an undirected component that does not suggest factorization over σ_1, σ_2 and σ_3 for any given x, while the factor graph explicitly encodes that.

A.3 Structure Elicitation and Learning

Having agreed that an intervention factor model is the natural choice under the scenarios described above, the remaining consideration is: how to extract structural knowledge from an expert or algorithm.

Simply put, removing edges from σ to X (or among X) *should be business as usual* once we understand that structure follows from conditional independencies among random variables and regime indicators, with the global Markov condition being that of a factor graph model. An expert who is ready to answer conditional ignorability questions of the type $Y \perp\!\!\!\perp \sigma_x \mid X, Z$, and/or plain

[12] As it would not be the case, for instance, if a choice of σ_j corresponds to removing a condition or resource required for carrying out an action encoded by some values of σ_i. This happens, for example, in resource allocation problems, where σ_i and σ_j correspond to resource allocation decisions limited by some budget constraint.

consistency-like questions to judge whether $\sigma_x \perp\!\!\!\perp Y \mid C$ for some C and $Y \neq X$, should have no qualms about answering general questions for interventions that don't have a particular single-variable target. In particular, the notion of "removing edges" from σ to X (that is, forbidding any factor to include particular combinations of intervention and random variables) should follow from knowledge about the domain. Indeed, impossibility of some direct connections is presented in all sorts of systems, from physical ones (including spatial systems [5]) to social ones (e.g., [59]).

Independencies of the type $\sigma_i \perp\!\!\!\perp \sigma_j \mid X, \sigma_{\setminus ij}$ are better understood as the lack of particular interactions. Non-linear multivariate models such as log-linear models have for long been understood as defining hierarchies of interactions within the allowed probabilistic dependencies [10]. Likewise, analysis of variance (ANOVA) is predicated on the idea that lower-order interactions can suffice to model a variety of empirical phenomena. Judging whether a set of random variables ("soft constraint") should be regulated by interactions of particular intervention variables, conditional on all other variables, is knowledge akin to judging interactions in ANOVA or log-linear models, and it has a long tradition in multivariate analysis, dating back at least to the work of Ising on statistical mechanics [56].

However, none of that is to say that the work of structure elicitation is straightforward; it is definitely not. With this in mind, we now conclude this section with broad ideas on structure learning for IFMs. Indeed, structure learning can aid the process of structure elicitation. An early method for structure learning of probabilistic factor graph models is described by [1], and it is our intent to provide a fully detailed account of structure learning for IFMs in future work.

There is a close link between classical DAG learning algorithms and algorithms for undirected models, using variants of the faithfulness assumption [72]. In particular, akin to the initial stage of the PC algorithm [72], we can start with a fully connected undirected graph and remove edges, creating a factor graph model out of the cliques remaining after a step that removes edges.

We can remove edges between random variables and edges between random variables and intervention variables, by querying an independence oracle under a particular regime σ^0, which we assume all other regimes should be faithful to. For instance, this takes place when σ^0 is an "observational regime" as defined by an unperturbed system, and any independence among random variables is assumed be carried over to other regimes. Likewise, varying one entry in σ and assessing (conditional) equality in distribution for particular random variables will remove undirected edges between σ and X by assuming that they are also be unnecessary under any other configuration of the unchanged variables.

Finally, we note that edges "within" σ variables, and interactions in general, are less straightforward to deal with. For any clique remaining in the current undirected graph, one possibility is to test whether the distribution of this clique given all other variables provides the same goodness-of-fit (by statistical significance) with or without particular interactions. Statistical power may be an issue, see [68] for a discussion on nonparametric testing of three-way interactions. An alternative is to adopt a blanket assumption to remove higher-order interactions if there is no evidence against lower-order interactions across models fit separately in each of the $\mathcal{D}^i$ datasets collected.

B Proofs of Identifiability and Further Examples

This section presents results concerning Theorems 3.1 and 3.2. We start with some background with textbook definitions, followed by proofs and examples for the decomposable graph, and concluding with proofs for the purely algebraic case. In particular, the decomposable case sheds light on how to hierarchically structure products and ratios of $\mathbb{P}(\Sigma_{\text{train}})$. Among other uses, this theoretically suggests which regimes could be added to Σ_{train} in order to reduce the estimation error coming from particular product/ratios that are required to identify larger marginal distributions.

Background. A *decomposition* of an undirected graph $\mathcal{G}$ is formed from a partition of its vertices into a triplet (A, B, C) where C is a complete subgraph (i.e., a clique) and separates A from B. The decomposition is called *proper* if both A and B are non-empty. Moreover, an undirected graph $\mathcal{G}$ is called *decomposable* if it is complete or, failing that, it has a proper decomposition into a triplet (A, B, C), where the subgraph of $\mathcal{G}$ with vertices $A \cup B$ and the subgraph with vertices $B \cup C$ are both decomposable.

A *junction tree* $\mathcal{T}$ of a decomposable graph $\mathcal{G}$ is a tree where each vertex V_i is labeled with the elements of a unique maximal clique from $\mathcal{G}$ (hence, this type of vertex is sometimes called a hypervertex), so that $V_i \cap V_j$ denotes the corresponding intersection among sets of vertices in $\mathcal{G}$. Edges $V_i - V_j$ of $\mathcal{T}$ are graphically represented with labels denoting the intersection $V_i \cap V_j$ (hence, these edges are also sometimes called hyperedges). A junction tree must have a *running intersection* property: any intersection $V_i \cap V_j$ must be contained in all vertices in the (unique) path between V_i and V_j in $\mathcal{T}$.

If $\mathcal{G}_{\sigma(\mathcal{I})}$ is decomposable, there exists at least one junction tree compatible with it. Let $\mathcal{T}$ be one of them. Without loss of generality, pick an arbitrary vertex of $\mathcal{T}$ to be the root and direct edges away from it to create a directed tree out of the junction tree, so that we can assume $\mathcal{T}$ to be directed. We will prove identifiability by an induction argument that starts at the leaves of the directed junction tree, moving towards the (unique) parent of any particular vertex child in the induction step.

Definitions and notation. In what follows, we use V_k to denote a vertex in $\mathcal{T}$. By abuse of notation, depending on context, V_k is also used to denote the corresponding intervention variables σ_{F_k} in the original factor graph.

Let $\sigma^{[Z(w)]}$ be a particular instantiation of σ, where $Z \subseteq [d]$ and $\sigma_Z = w$ (possibly a vector), with the remaining entries of σ being zero. For instance, if $d = 3$, $Z = \{2, 3\}$, and $w = (2, 1)$, then $\sigma^{[Z(w)]} = (0, 2, 1)$. To avoid subsequently heavy notation, from this point on we will use $\sigma^{[Z(\star)]}$ to denote $\sigma^{[Z(\sigma^\star_Z)]}$, that is, the vector of assignments that we obtain by setting to zero all entries of $\sigma^\star$ which are not in Z.

We use $ch(k)$ to denote the set of children of vertex V_k in $\mathcal{T}$, and $V_{\pi(k)}$ to denote its (unique) parent, if V_k is not the root vertex. Also, let D_k denote the union of the intervention variables contained in at least one descendant of V_k in $\mathcal{T}$, remembering that by convention V_k is also a descendant of itself. Finally, let $B_k := D_k \cap V_{\pi(k)}$, the set of intervention variables common to both D_k and $V_{\pi(k)}$. This means that, by the running intersection property of junction trees, B_k separates $A_k := D_k \backslash B_k$ from the rest of σ in the σ-graph $\mathcal{G}_{\sigma(\mathcal{I})}$.

Proof of Theorem 3.1. To simplify the proof, assume without loss of generality that no entry in $\sigma^\star$ is zero. To see this, if $\sigma^\star_i = 0$, consider the factors k containing σ_i as being constants in σ_i, with scope S_k being redefined as $S_{k'} := S_k \backslash \{\sigma_i\}$. Σ_{train} in this redefined space still satisfies the assumption of having entries spanning all possible values for $\sigma_{S_{k'}}$ while holding the remaining intervention variables at the background level of 0. Likewise, as identifiability will be shown pointwise for a given $\sigma^\star$ (where the categorical labels for the intervention values are arbitrary symbols), we can assume all entries $\sigma^\star_i$ as being equal, and equal to 1.

We define a *message* from vertex V_k to its parent $V_{\pi(k)}$ as

$$m_k^x := \frac{p(x; \sigma^{[D_k(\star)]})}{p(x; \sigma^{[B_k(\star)]})}, \tag{6}$$

and state that

$$p(x; \sigma^{[D_k(\star)]}) \propto p(x; \sigma^{[F_k(\star)]}) \prod_{V_{k'} \in ch(k)} m^x_{k'}, \tag{7}$$

with the product over $ch(k)$ defined to be 1 if $ch(k) = \emptyset$.

We will show how Eq. (6) can be recursively identified from a message scheduling that starts from the leaves and propagates messages towards the root of $\mathcal{T}$. We will also show how Eq. (7) holds.

Let V_k be a leaf of $\mathcal{T}$. Then $D_k = F_k$ and $p(x; \sigma^{[F_k(\star)]})$ is identified as it is part of Σ_{train}, showing that Eq. (7) holds for the leaf vertices of $\mathcal{T}$. Likewise, message m^x_k is identifiable for leaf vertices, as both D_k and B_k are contained in F_k.

Let V_k now be an internal vertex of $\mathcal{T}$, and assume that Equations (6) and (7) are identified for all of its proper descendants. As all entries of $\sigma^\star$ are assumed to be 1, it will be useful to define $g_k := f_k(x; \sigma_{F_k} = 1)$. Also define $h_k := f_k(x; \sigma_{F_k \cap B_k} = 1, \sigma_{F_k \setminus B_k} = 0)$ and $z_k := f_k(x; \sigma_{F_k} = 0)$.

Since $\mathcal{T}$ is a junction tree, D_k can be partitioned into sets $D_{k'}$, where $V_{k'} \in ch(k)$, or otherwise there would be a violation of the running intersection property. Let $Q(k')$ be the set of all indices of factors q where $F_q \subseteq D_{k'}$. Let

$$\mathcal{Q}_{k'} := \prod_{q \in Q(k')} \frac{g_q}{h_q}.$$

We can multiply and divide $\mathcal{Q}_{k'}$ by the product of all factors z_q, where $F_q \cap D_{k'} = \emptyset$. This implies

$$\mathcal{Q}_{k'} \propto \frac{p(x; \sigma^{[D_{k'}(\star)]})}{p(x; \sigma^{[B_{k'}(\star)]})} = m^x_{k'}.$$

Moreover, the product

$$\frac{\prod_{q: F_q \cap D_k = \emptyset} z_q}{\prod_{q: F_q \cap D_k = \emptyset} z_q} \times \frac{f_k(x; \sigma_{F_k} = 1)}{f_k(x; \sigma_{F_k} = 1)} \times \frac{f_{\pi(k)}(x; \sigma_{B_k} = 1, \sigma_{F_{\pi(k)} \setminus B_k} = 0)}{f_{\pi(k)}(x; \sigma_{B_k} = 1, \sigma_{F_{\pi(k)} \setminus B_k} = 0)} \times \prod_{k': V_{k'} \in ch(k)} \mathcal{Q}_{k'}$$

is such that the numerator is proportional to $p(x; \sigma^{[D_k(\star)]})$ and the denominator is proportional to $p(x; \sigma^{[F_k(\star)]})$. To see this, note that the numerator sets the σ_{F_j} variables for all factors F_j in a coherent way such that entries in D_k are set to 1 while everything else is set to zero (entries in D_k may still appear in $V_{\pi(k)}$, as $D_k \cap F_{\pi(k)} = B_k$, is possibly non-empty. Hence, we set to 1 those entries in $F_{\pi(k)}$ which are in B_k, explaining the appearance of the $f_{\pi(k)}$ factors in the expression above).

This implies

$$\frac{p(x; \sigma^{[D_k(\star)]})}{p(x; \sigma^{[F_k(\star)]})} \propto \prod_{V_{k'} \in ch(k)} m^x_{k'},$$

from which Eq. (7) follows from quantities previously identified, and as such it identifies $p(x; \sigma^{[D_k(\star)]})$.

To build message m^x_k, all that remains is $p(x; \sigma^{[B_k(\star)]})$. However, $B_k \subseteq F_k$, and since Σ_{train} contains the distribution $p(x; \sigma^{[H_k(\star)]})$ for all $H_k \subseteq F_k$, this is also identified. The required identifiability of $p(x; \sigma^\star)$ follows from propagating these messages all the way up to the root of $\mathcal{T}$. □

Example. We now solve the example shown in Figures 7–9. The IFM itself is given by

$$p(x; \sigma) \propto f_1(x; \sigma_1, \sigma_2) f_2(x; \sigma_2, \sigma_3) f_3(x; \sigma_2, \sigma_5) f_4(x; \sigma_3, \sigma_4) f_5(x; \sigma_4, \sigma_6) f_6(x; \sigma_6, \sigma_7) f_7(x; \sigma_4, \sigma_8),$$

where all intervention variables are binary, and we our goal is to generate the test regime where all intervention variables are set to 1, i.e., $\sigma_1 = \sigma_2 = \cdots = \sigma_8 = 1$. For reference, this means considering the following sets implied by the factorization above:

$$\begin{array}{lll}
F_1 = \{1, 2\} & D_1 = \{1, 2\} & B_1 = \{2\} \\
F_2 = \{2, 3\} & D_2 = \{1, 2, 3, 5\} & B_2 = \{3\} \\
F_3 = \{2, 5\} & D_3 = \{2, 5\} & B_3 = \{2\} \\
F_4 = \{3, 4\} & D_4 = \{1, 2, 3, 4, 5, 6, 7, 8\} & B_4 = \emptyset \\
F_5 = \{4, 6\} & D_5 = \{4, 6, 7, 8\} & B_5 = \{4\} \\
F_6 = \{6, 7\} & D_6 = \{6, 7\} & B_6 = \{6\} \\
F_7 = \{4, 8\} & D_7 = \{4, 8\} & B_7 = \{4\}
\end{array}$$

To illustrate how message passing will work in this example, let's introduce some symbols so that the steps are easier to follow. Let g_{ij}^{11} represent a factor with $\sigma_i = \sigma_j = 1$. For instance, $g_{12}^{11} = f_1(x; \sigma_1 = 1, \sigma_2 = 1)$. This is slightly redundant compared to the notion in the proof (which uses "g_1" to denote $f_1(x; \sigma_1 = 1, \sigma_2 = 1)$), but the redundancy of the superscripts will hopefully make it easier to visualize the logic in the steps that follow.

Likewise, let h_{ij}^{01} and h_{ij}^{10} denote assignments $(\sigma_i, \sigma_j) = (0, 1)$ and $(\sigma_i, \sigma_j) = (1, 0)$, respectively. Finally, let z_{ij}^{00} denote the respective factor with assignment $\sigma_i = \sigma_j = 0$.

We will use expressions such as $p(x; [ij])$ to denote $p(x; \sigma_i = 1, \sigma_j = 1, \sigma_{1:8\setminus\{i,j\}} = 0)$ to make the notation simpler.

The messages at the leaves are

$$\begin{array}{rcll} m_1^x & = & p(x;[12])/p(x;[2]) & (D_1 = \{1,2\}, B_1 = \{2\}) \\ m_3^x & = & p(x;[25])/p(x;[2]) & (D_3 = \{2,5\}, B_3 = \{2\}) \\ m_6^x & = & p(x;[67])/p(x;[6]) & (D_6 = \{6,7\}, B_6 = \{6\}) \\ m_7^x & = & p(x;[48])/p(x;[4]) & (D_7 = \{4,8\}, B_7 = \{4\}) \end{array}$$

It can be readily verified that all of these are identifiable from Σ_{train}, as all non-zero assignments are contained within some factor.

Now, let's pass messages to $(2, 3)$ using formula (7). To see how it is applicable, start from

$$\begin{aligned} m_1^x \times m_3^x &= \frac{p(x;[12])}{p(x;[2])} \frac{p(x;[25])}{p(x;[2])} \\ &\propto \frac{g_{12}^{11} h_{23}^{10} h_{25}^{10} z_{34}^{00} z_{46}^{00} z_{67}^{00} z_{48}^{00}}{h_{12}^{01} h_{23}^{10} h_{25}^{10} z_{34}^{00} z_{46}^{00} z_{67}^{00} z_{48}^{00}} \times \frac{h_{12}^{01} h_{23}^{10} g_{25}^{11} z_{34}^{00} z_{46}^{00} z_{67}^{00} z_{48}^{00}}{h_{12}^{01} h_{23}^{10} h_{25}^{10} z_{34}^{00} z_{46}^{00} z_{67}^{00} z_{48}^{00}} \end{aligned}$$

Now, we multiply and divide it by the factor of $(2, 3)$ and its parent $(3, 4)$ evaluated at $(\sigma_2, \sigma_3, \sigma_4) = (1, 1, 0)$, and reorganize the numerator and denominator:

$$\begin{aligned} m_1^x \times m_3^x &= \frac{p(x;[12])}{p(x;[2])} \frac{p(x;[25])}{p(x;[2])} \\ &\propto \frac{g_{12}^{11} \cancel{h_{23}^{10}} \cancel{h_{25}^{10}} \cancel{z_{34}^{00}} \cancel{z_{46}^{00}} \cancel{z_{67}^{00}} \cancel{z_{48}^{00}}}{h_{12}^{01} \cancel{h_{23}^{10}} \cancel{h_{25}^{10}} \cancel{z_{34}^{00}} \cancel{z_{46}^{00}} \cancel{z_{67}^{00}} \cancel{z_{48}^{00}}} \times \frac{\cancel{h_{12}^{01}} \cancel{h_{23}^{10}} g_{25}^{11} \cancel{z_{34}^{00}} z_{46}^{00} z_{67}^{00} z_{48}^{00}}{\cancel{h_{12}^{01}} \cancel{h_{23}^{10}} h_{25}^{10} \cancel{z_{34}^{00}} z_{46}^{00} z_{67}^{00} z_{48}^{00}} \times \frac{g_{23}^{11}}{g_{23}^{11}} \times \frac{h_{34}^{10}}{h_{34}^{10}} \\ &= \frac{g_{12}^{11} g_{23}^{11} g_{25}^{11} h_{34}^{10} z_{46}^{00} z_{67}^{00} z_{48}^{00}}{h_{12}^{01} g_{23}^{11} h_{25}^{10} h_{34}^{10} z_{46}^{00} z_{67}^{00} z_{48}^{00}} \\ &\propto \frac{p(x;[1235]}{p(x;[23])}. \end{aligned}$$

As $p(x; [23])$, m_1^x and m_3^x have been previously identified, from the above we get the update for $p(x; [1235])$ per Eq. (7), pointing out that indeed $D_2 = \{1, 2, 3, 5\}$ and $F_2 = \{2, 3\}$.

To construct the message m_2^x the factor $(2, 3)$ needs to pass to its own parent $(3, 4)$, we also need the corresponding $p(x; \sigma^{[B_2(\star)]})$, which in the example notation is $p(x; [3])$. But as $B_2 = \{3\}$ is contained in $F_2 = \{2, 3\}$, and this will be the case for all (B_k, F_k) pairs, by assumption Σ_{train} will contain $p(x; [3])$. Therefore, we identified m_2^x.

The steps for $(4, 6)$ and $(3, 4)$ follow identical, if somewhat tedious, reasoning.

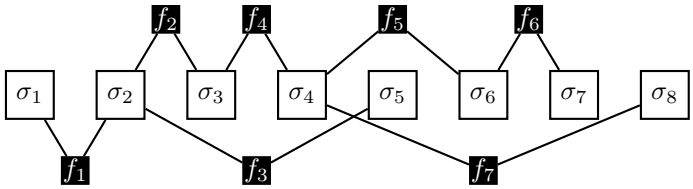


Figure 7: The factor graph used as an illustration of the technique in the proof of Theorem 3.1.

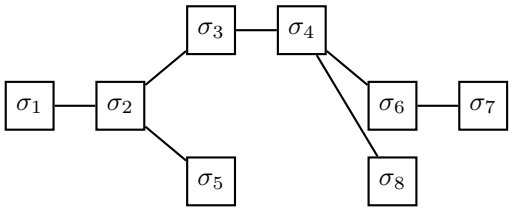


Figure 8: The σ-graph corresponding to the factor graph in Figure 7.

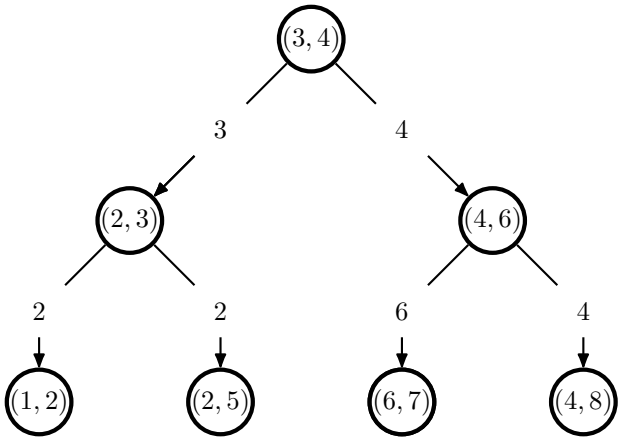


Figure 9: A (directed) junction tree corresponding to the undirected graph in Figure 8.

Proof of Theorem 3.2. Sufficiency follows immediately from the fact that, under Eq (4) being satisfied, the PR-transformation $\prod_{i=1}^{t} p(x;\sigma^i)^{q_i}$ is equivalent to

$$\prod_{k=1}^{l} \prod_{\sigma^v_{F_k} \in \mathbb{D}_k} f_k(x_{S_k};\sigma^v_{F_k})^{\sum_{i=1:\sigma^i_{F_k}=\sigma^v_{F_k}}^{t} q_i} = \prod_{k=1}^{l} f_k(x_{S_k};\sigma^\star_{F_k}) \propto p(x;\sigma^\star), \tag{8}$$

for all x.

For almost-everywhere necessity, let $z_{kv} := \log f_k(x_{S_k};\sigma^v_{F_k})$. Taking the logarithm on both sides of the equality in Eq. (8), we have

$$\sum_{k=1}^{l} \sum_{\sigma^v_{F_k} \in \mathbb{D}_k} z_{kv} \left(\sum_{i=1:\sigma^i_{F_k}=\sigma^v_{F_k}}^{t} q_i \right) = \sum_{k=1}^{l} z_{k^\star},$$

which implies

$$\sum_{k=1}^{l} z_{k^\star} \left(\sum_{i=1:\sigma^i_{F_k}=\sigma^\star_{F_k}}^{t} q_i \right) + \sum_{k=1}^{l} \sum_{\sigma^v_{F_k} \in \mathbb{D}_k \setminus \{\star\}} z_{kv} \left(\sum_{i=1:\sigma^i_{F_k}=\sigma^v_{F_k}}^{t} q_i \right) = 0.$$

As no $z_{k^\star}$ appears in the second term of the expression above, the only way for this equality to hold without $\{q_1, \ldots, q_t\}$ satisfying Eq. (4) is if constrains tie together the different $z_{k^\star}$. For any reasonable continuous measure by which the parameters of such functions are free to be chosen from (say, as draws of a multivariate Gaussian), this will be a set of measure zero. □

Discussion. As a corollary it is implied that, similar to the conditions of Theorem 3.1, we need to have at least one train condition σ^i for every possible combination of σ_{F_k}, for each F_k. To see why, imagine if the example of Figure 4 we did not have condition 4, that is, $(\sigma_1, \sigma_2, \sigma_3) = (1, 1, 0)$ is left out. This means that there is nothing to be added in the column f_1^{11}, and the sum $\sum_{i=1:\sigma^i_{F_1}=(1,1)}^{t} q_i$ evaluates to 0, implying $0 = 1$.

C More on Elicitation, Testability and Experimental Design

As mentioned before, the main result shows that the factorization over X is unimportant for identifiability, which may be surprising. However, it is important to remember that identifiability and testability are two different concepts. While Figure 2**(b)** has testable implications of conditional independence, testing factorizations may require more intervention levels than the minimal set implied by Theorem 3.1. In particular, if we have a model $p(x; \sigma) \propto \prod_{k=1}^{l} f_k(x_k; \sigma_k)$, we may be able to identify the model by singleton experiments spanning the range of each σ_i individually, however, this does not mean we can falsify this factorization with just this data.

In general, our advice for graph construction is akin to any causal modeling exercise: apply independence constraint tests and interaction tests where applicable (see [68] for an example of nonparametric three-way interaction test), but untestable conditions (under the available data) can be used if there is a sensible theoretical justification for it. This means expert assessment of the lack of direct dependency between an intervention variable and particular random variables, and of the split of σ into sets F_k from postulated lack of interactions among intervention variables when causing particular random variables. Although not necessarily always the case, we anticipate that in general this exercise will imply a factorization over the random variables too.

Also of interest is understanding which minimal size Σ_{train} should have in order to identify a particular test regime. This is straightforward to answer in the decomposable case: simply ensure that the regimes used in the messages of the message passing scheme are available in the training set. A simple iterative algorithm can list the required messages for a target regime $\sigma^\star$. For example, with binary treatments, a σ-graph without any edges (no interaction of intervention variables in a same factor), and the goal of identifying *all* combinations of interventions, this is simply $d + 1$, where d is the number of intervention variables (this follows from having the baseline regime plus one regime where a single intervention variable is set to 1).

For non-decomposable graphs, we can triangulate the corresponding σ-graph and run the same procedure defined for decomposable graphs to provide an upper bound on number of training conditions and a superset of conditions. We can run a greedy procedure to iteratively remove redundant entries in Σ_{train} by proposing candidate training regimes to be removed and testing whether the PR condition for the regime $\sigma^\star$ of interest is still satisfied.

What if the cardinality $\aleph_i$ of some σ_i is very high? Without smoothness assumptions, getting a reasonable dose-response pattern with few evaluations of σ_i is clearly impossible regardless of any method — this is true even for a single intervention variable in the $[0, 1]$ interval where (say) $p(x; \sigma)$ jumps arbitrarily as we sweep the values of σ in $[0, 1]$. *With* smoothness assumptions, we can simply elicit a grid of values for the intervention variables, ask conditions for the identifiability of those, and fill up the remaining potential functions/expected outcome values of interest via whatever smoothing procedure we deem appropriate (from potential functions which are smooth functions of σ or via partial identification procedures, see e.g. [35]). There is no free lunch.

D Proof of Predictive Coverage Result

Proof of Theorem 4.1. Vovk et al. [78] introduced a distribution-free procedure for computing prediction intervals with guaranteed finite sample coverage, under the assumption that training and test data are exchangeable. Lei et al. [51] proposed a more computationally tractable version that they called the "split conformal" method, and derived a novel upper bound on conformal coverage. We review some fundamental results.

Consider the regression setting with $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and $Y \in \mathcal{Y} \subseteq \mathbb{R}$. We partition the data into equal-sized subsets $\mathcal{I}_1, \mathcal{I}_2$, using the former for model training and the latter for computing conformity scores. For instance, we may fit a model $\widehat{f}(x)$ to estimate $\mathbb{E}[Y \mid x]$ using samples from $\mathcal{I}_1$ and consider the score function $s^{(i)} = |y^{(i)} - \widehat{f}(x^{(i)})|$ for $i \in \mathcal{I}_2$. Let $\widehat{\tau}$ be the q^{th} smallest value in S, with $q = \lceil (n/2+1)(1-\alpha) \rceil$. Define $\widehat{C}(x) = \widehat{f}(x) \pm \widehat{\tau}$. (We assume symmetric errors for convenience; the result can easily be modified by invoking the appropriate quantiles of the residual distribution.)

Theorem D.1 (Split conformal inference [51].) *Fix a target level $\alpha \in (0,1)$. If $(x^{(i)}, y^{(i)}), i \in [n]$, are exchangeable, then for any new $n+1$ from the same distribution:*

$$\mathbb{P}\big(Y^{(n+1)} \in \widehat{C}(X^{(n+1)})\big) \geq 1 - \alpha.$$

Moreover, if scores have a continuous joint distribution, then the upper bound on this probability is $1 - \alpha + 2/(n+2)$.

Tibshirani et al. [76] extend this result beyond exchangeable data by introducing the notion of *weighted exchangeability*. We call random variables $V_1, \ldots, V_n$ *weighted exchangeable*, with weight functions $w_1, \ldots, w_n$, if their joint density can be factorized as:

$$f(v_1, \ldots, v_n) = \prod_{i=1}^{n} w_i(v_i) \cdot g(v_1, \ldots, v_n),$$

where g does not depend on the ordering of its inputs, i.e., g is permutation invariant. This entails the following lemma.

Lemma D.2 (Weighted exchangeability [76].) *Let $Z_i \sim P_i, i \in [n]$, be independent draws, where each P_i is absolutely continuous with respect to P_1, for $i \geq 2$. Then $Z_1, \ldots, Z_n$ are weighted exchangeable, with weight functions $w_1 = 1$ and $w_i = dP_i/dP_1, i \geq 2$.*

This allows us to generalize the conformal guarantee to weighted exchangeable distributions. Let $\widetilde{w}^{(i)}(x)$ denote a rescaled version of the weight function, such that weights sum to n. The original paper does not use the split conformal approach, but we adapt the result below. First, we reweight the empirical scores to create the new distribution $\sum_i \widetilde{w}^{(i)}(x) \cdot \delta(s)^{(i)}$, where δ denotes the Dirac delta function. Now, let $\widehat{\tau}(x)$ be the q^{th} smallest value in $\sum_i \widetilde{w}^{(i)}(x) \cdot \delta(s)^{(i)}$, with q defined as above. Then, we construct the weighted conformal band $\widehat{C}_w(x) = \widehat{f}(x) \pm \widehat{\tau}(x)$ for all $x \notin \mathcal{I}_1$.

Theorem D.3 (Split weighted conformal inference [76].) *Fix a target level $\alpha \in (0,1)$.*
If $(x^{(i)}, y^{(i)})$, are weighted exchangeable with weight functions $w^{(i)}, i \in [n]$, then for any new $n+1$:

$$\mathbb{P}\big(Y^{(n+1)} \in \widehat{C}_w(X^{(n+1)})\big) \geq 1 - \alpha.$$

Moreover, if scores have a continuous joint distribution, then the upper bound on this probability is $1 - \alpha + 2/(n+2)$.

Our case is somewhat trickier, as we do not have access to data X from the unobserved environment $\sigma^\star$ and our regime variables are not random, so ratios such as $p(\sigma^a)/p(\sigma^b)$ are undefined. However, we can use a similar reweighting strategy based on likelihood ratios of the form $p(x^{k(i)}; \sigma^\star)/p(x^{k(i)}; \sigma^k)$ to ensure that conformity scores satisfy weighted exchangeability with respect to any target regime. This works because we observe the mediators x for each conformity score s, and assume identifiability of the relevant likelihood ratios via previous Theorems 3.1 and/or 3.2. Thus our conformal bands are functions of σ, not X, and our result is simply a special case of the split weighted conformal inference theorem. □

E Covariate Shift Method

As IPW may have large variance, one alternative is to use covariate shift regression [53]. In particular, for each test regime $\sigma^\star$, we provide a customize estimate of $f_y(x) := \mathbb{E}[Y|X]$.

As before, we combine data from all training regimes $\mathcal{D}^1, \ldots, \mathcal{D}^t$, but reweighting then according to (the estimated) $p(x; \sigma^\star)$. We propose minimizing the following objective function,

$$\mathcal{L}_y(\theta_y) := \sum_{i=1}^{t} \sum_{j=1}^{n_i} (y^{ij} - f_y(x^{ij}; \theta_y))^2 w^{ij^\star},$$

where $w^{ij^\star}$ is an estimate of $p(x^{ij}; \sigma^\star)/p(x^{ij}; \sigma^i)$, and all the training data regimes are weighted equally given that $\sum_{j=1}^{n} w^{ij^\star} = 1$ for all i. As done with the IPW method, this ratio is taken directly from the likelihood function of the deep energy-based model we described for the direct method.

When generating an estimate $\widehat{\mu}_{\sigma^\star}$, we just apply the same idea as in the direct method, where samples from the estimated $p(x; \sigma^\star)$ are generated by Gibbs sampling, so that we average $f_y(x; \widehat{\theta}_y)$ over these samples.

As we are averaging over $f_y(X)$ instead of considering predictions at each realization of X, the main motivation for covariate shift here is to improve on IPW by substituting the use of Y^{ij} as the empirical plug-in estimate of $\mathbb{E}[Y^{ij} \mid x^{ij}]$ with a smoothed version of it given by a shared learned $f_y(X^{ij}; \theta_y)$.

However, in our experiments, this covariate shift method was far too slow when considering the cost over the entire Σ_{test} (as expected, given that the output model is fitted again for every test regime) and did not show concrete advantages compared to the direct method.

F Pseudolikelihood Method

The pseudo-loglikelihood function $p\mathcal{L}(\theta; \mathcal{D}^1, \ldots, \mathcal{D}^t)$ is given by

$$p\mathcal{L}(\theta; \mathcal{D}^1, \ldots, \mathcal{D}^t) := \sum_{i=1}^{t} \sum_{j=1}^{n_i} \sum_{r=1}^{m} \log p_\theta(x_r^{i(j)} \mid x_{\setminus r}^{i(j)}; \sigma^i),$$

where $x_r^{i(j)}$ is the r-th variable of the j-th data point in dataset $\mathcal{D}^i$, with n_i being the number of data points in $\mathcal{D}^i$. Vector $x_{\setminus r}^{i(j)}$ for the same data point is composed of all other random variables but the r-th variable.

The log-conditional distribution $\log p_\theta(x_r^{i(j)} \mid x_{\setminus r}^{i(j)}; \sigma^i)$ is given by

$$\log p_\theta(x_k^{i(j)} \mid x_{\setminus k}^{i(j)}; \sigma^i) = \sum_{k=1}^{l} \phi_{k,\sigma_{F_k}}(x_{S_k}^{i(j)}) - \log \left\{ \sum_{x'_r} \exp \left(\sum_{k=1}^{k} \phi_{k,\sigma_{F_k}}(x'^{i(j)}_{S_k}) \right) \right\},$$

where the second term on the right-hand side is the log-normalizing constant summing all possible values x'_r of the r-th random variable. Here, $x'^{i(j)}_{S_k}$ is the vector obtained by substituting the value of x_k within data point j of dataset i with x'_k, prior to selecting the subvector corresponding to S_k.

The above assumes that all variables are discrete. As described in the main text, we discretize our variables in an uniform grid, preserving the magnitude information. The parameterization θ is the same regardless of the number of discretization levels, so that this number is chosen basically by the computational considerations of performing the sum over x'_k in the log-normalizing constant. Finer discretizations preserve more information but increase this cost.

G Experimental Details

In this section, we present further experimental details for Section 5, including the full setup for the datasets (Sachs and DREAM), the oracular simulators (Causal-DAG and Causal-IFM) that generate the ground-truth X and Y, and the training procedures we used in the experiments.

G.1 Datasets

Sachs (et al.) dataset. The original Sachs et al. study [66] consisted of 14 different datasets collected under different compound perturbations in single-cell systems measured by 11 protein/lipid concentrations. Perturbations can be described in terms of binary intervention variables, labeled by the associated compound. For instance, condition pma describes the introduction or not of phorbol 12-myristate 13-acetate. Among all perturbations, pma and $b2camp$ are entangled with $cd3cd28$ (this means, for instance, that $pma = 1$ or $b2camp = 1$ imply $cd3cd28off = 1$). Hence, we ignore these two experimental setups, and all remaining datasets are collected under $cd3cd28 = 1$ so that it can be considered as an implicit condition not modeled explicitly with a separate intervention variable.

Other conditions implying unresolved entanglements were not considered, in particular the uses of $icam$-2 and $ly294002$. The remaining datasets are listed in Table 1. Assumptions about each intervention targeting a single protein in the network are taken from [66]. In summary, the original Sachs et al. data used to train the simulator contains samples from 5 (1 "baseline" plus 4 "perturbed") different regimes, and each data sample has 11 variables.

Since each intervention is considered as a binary value (0 for no perturbation and 1 for perturbation), this gives us a total of $2^4 = 16$ combinatorial possibilities, with 5 in Σ_{train}. Hence, we need a way of establishing a (synthetic) ground truth for the $16 - 5 = 11$ possible test conditions, which we explain in Section G.2.

DREAM dataset. The DREAM challenges include a series of problems for causal inference in protein networks [30]. We generate data based on a known *E. coli* sub-network with 10 nodes, and consider that each random variable X_i has a corresponding interventional variable σ_i. We use the GeneNetWeaver simulator[13] to generate this data, under "InSilicoSize10-Ecoli1" from the "DREAM3_In-Silico_Size_10" task and there is no further data selection process as in the Sachs case. The simulation is based on a series of predefined ODEs and SDEs. For each data regime, a single data sample is collected with a random seed initialization with an otherwise exact similar simulation setting for that particular regime. Following [77], we gather the data sample once it reaches its equilibrium state and repeat this process as many times as the sample size is required. In summary, this provides us with a dataset consisting of 11 (1 baseline plus 10 perturbation) regimes. As we are interested in combinations of 10 binary indicator variables σ_i, not directly provided in the original DREAM simulator, we had to create our own ground-truth synthetic model based on samples from the 11 regimes we can obtain from DREAM.

Table 1: Details for the Sachs et al. datasets used for our first batch of intervention generalization experiments. Data files can be downloaded from the website of the original reference [66], with the name described below. Column *Target X node* describes the theoretical direct connection (as given by [66]) between the perturbation and 11-dimensional system described by a vector of 11 random variables X, with condition $cd3cd28$ always present and affecting all variables, and hence interpreted as a targeting none. As described in Section G.2, we encode each regime as a 11-dimensional binary vector, and display them in the last column. A Julia notebook exemplifying the pre-processing of this data and a Julia script outlining a complete pipeline of batch simulated experiments comparing methods is provided in the supplementary material.

File name	Target X Node	Data Regime	Corresponding σ
cd3cd28.xls	None (background condition)	Regime 0	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
cd3cd28+aktinhib.xls	Variable 7	Regime 1	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
cd3cd28+g0076.xls	Variable 9	Regime 2	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
cd3cd28+psitect.xls	Variable 4	Regime 3	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
cd3cd28+u0126.xls	Variable 2	Regime 4	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[13] https://gnw.sourceforge.net

G.2 Oracular Simulators

Both Sachs and DREAM come with a ground truth DAG (either defined by expert domain knowledge or motivated by physical systems dynamics). We used each of the DAGs to construct the associated IFMs. To further explain: in a DAG, the joint probability distribution can be factorized based on the local Markov condition [46], where a single factor is defined by a vertex and its parents; this suggests are least one IFM, with the factorization following from interpreting each child-parents factor as a black-box (i.e., not normalized by the child) positive function of these variables. Graphically, this is known as the *moralization* step of "marrying the parents" followed by the dropping of directions in order to create an undirected Markov network [46]. We use this to define a factor graph model without stating that this would be the best representation for the corresponding data. It relaxes the DAG assumption (i.e., it removes some of the independence constraints encoded in the DAG) and could be refined by adding other constraints (such as breaking the factors into products of reduced sets of variables), which we do not attempt. See Figure 10 for an example with the Sachs et al. model. Approaches such as [1] could be used to refine this structure, if so desired.

Given two theoretical constructions and the respective parameterizations they use, this suggests *two* ways of building ground-truth simulator models to generate ground truth data X, which we now explain.

Common to both ground-truth simulators is the fitting of a postulated causal structure to real data (either Sachs or DREAM). Prior to fitting, we scale the data of each study so that the respecive "merged empirical distributions", defined by taking the union of all respective datasets collected under all available training regimes, have empirical mean of zero and empirical variance of 1 for each measured random variable. This does not mean any given variable in any given training regime will have zero empirical mean and unit variance, but pragmatically it helps to control having variables with disparate scales. For the Sachs data, we also take the logarithm of each random variable prior to standardization.

G.2.1 Causal DAG Ground-Truth

The first ground-truth simulator is implied by the respective causal DAG model. The DAG for each study are shown in Figures 10**(a)** and 11**(a)**. The factorization comes from the structure of the DAG and can be rewritten as follows:

$$p(x;\sigma) = \prod_{k=1}^{l} p(x_k \mid \mathbf{pa}(x_k); \sigma_{F_k}), \tag{9}$$

where, l is the total number of random variables, $p(x_k \mid \mathbf{pa}(x_k); \sigma_{F_k})$ is the conditional density function for x_k, σ_{F_k} is the regime indicator subvector for the intervention variables which are parents of x_k in the DAG, and $\mathbf{Pa}(x_k)$ refers to the random variables which are parents of x_k.

For parameterizing the causal DAG model family, we assume a heteroscedastic conditionally Gaussian formulation. This can be represented by the equation

$$X_k = f_k(\mathbf{pa}(X_k), \sigma_k) + g_k(\mathbf{pa}(X_k), \sigma_k) \times \epsilon_k, \epsilon_k \sim \mathcal{N}(0, 1).$$

Here, each f_k and g_k are multilayer perceptrons (MLPs) with 10 hidden units and the role of σ_k is just a switch: for each value of σ_k, we pick one independent set of parameters for the MLP mapping $\mathbf{pa}(X_k)$ to the real line. To learn the parameters in functions f_k and g_k, maximum likelihood is used. Further details are provided in the companion Julia code.

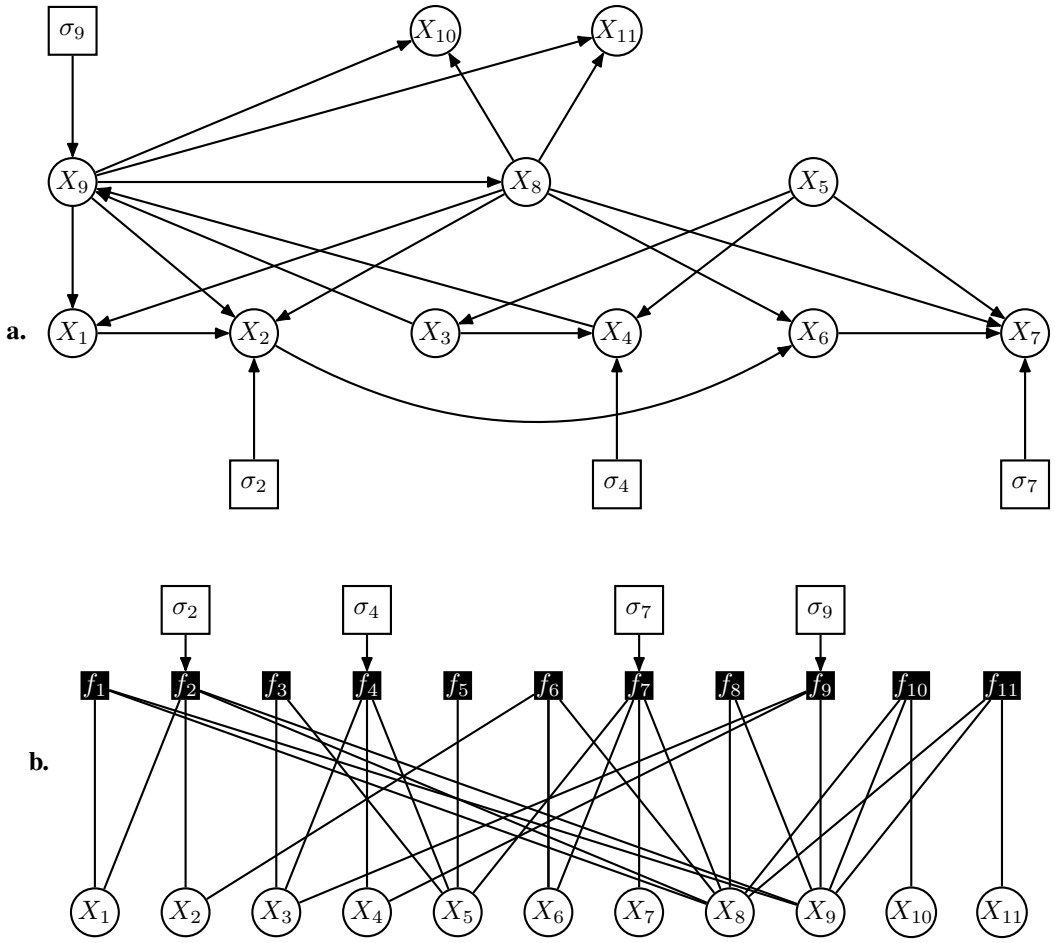


Figure 10: Causal structures used for building the synthetic ground-truth models for the Sachs et al. [66] data. The name of the random variables are $CD3CD28off$, $ICAM$-2, Akt-$inhibitor$, $G0076$, $Psitectorigenin$, $U0126$, $LY294002$, PMA, $B2camp$, with more details given in the companion Julia notebook. **(a)** A directed acyclic graph (DAG) for the Sachs et al. process, with intervention vertices representing intervention variables. **(b)** The interventional factor graph, inspired by the DAG, which we use in our synthetic ground-truth simulator. This is done by creating a factor for each child and parent set from the postulated DAG. These independence models are not equivalent. The point is *not* to provide an exact model, but to build a synthetic ground truth with parameters calibrated by real data instead of arbitrarily sampled, and with independence constraints and factorizations that do not contradict a given expert assessment (as the factor graph contains *fewer* independence assumptions than the DAG, not more).

G.2.2 IFM Ground-Truth

The second simulator is the causal IFM. The factorization comes from the structure of the DAG, using the moralization criterion described in the previous section. Figures 10**(b)** and 11**(b)** show the respective IFM graph structure. This results in the following factorization form:

$$p(x;\sigma) \propto \prod_{k=1}^{l} f_k(x_{\{k\}} \cup \mathbf{pa}(x_k); \sigma_{F_k}), \tag{10}$$

where $\mathbf{pa}(x_k)$ comes from the respective theoretical causal DAG case, with F_k given accordingly by k.

To learn the causal IFM simulator, we use pseudo-likelihood and assign each factor again to a black-box MLP of 15 hidden units where the corresponding σ_{F_k} is a switch between independent sets of parameters within each factor. We additionally perform a discretization step for variable X_i

by collecting all data and doing uniformly binning it in 20 bins, so that it is faster to compute the conditional normalizing constants[14] for each term in the pseudo-likelihood objective function.

To sample from the learned IFM, so that we can numerically compute quantities such as μ_σ, we use Gibbs sampling.

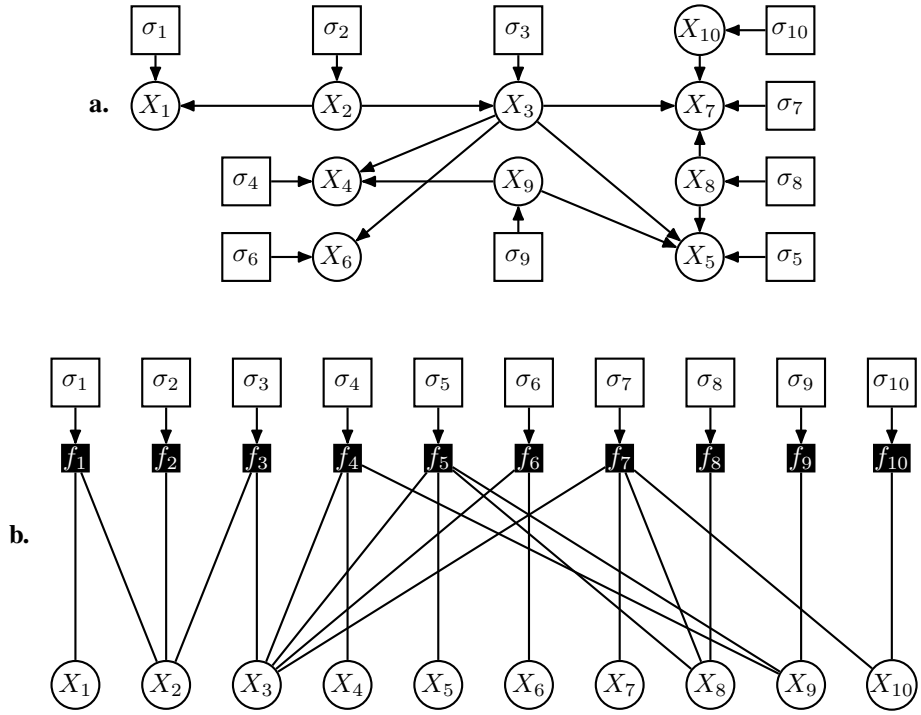


Figure 11: The DREAM structural assumptions, following a process analogous to the Sachs et al. case described in Figure 10.

G.3 Generating Ground-Truth Population Models and Data

Generating ground truth data, either for numerically computing population quantities by Monte Carlo or as a generator of training data, includes the following steps: (1) learning simulators, (2) generating ground truth X and (3) generating ground truth Y.

Learning simulators. The first step involves learning the simulators: with the Sachs et al. data, we use 5 data regimes as the training data for the simulator (1 baseline regime and 4 interventional); and with DREAM data, we use 11 data regimes as the training data for the simulator (1 baseline regime and 10 interventional). As described above, two simulators are built for each of the two studies.

Generating ground-truth system X. Since each intervention is considered as a binary value (0 for no intervention and 1 for with intervention), with the training dataset of 5 data regimes in Sachs, this gives us a total of $2^4 = 16$ combinatorial possibility regimes; as for the DREAM case, we have in total of 11 regimes, which means that the complete space Σ has $2^{10} = 1024$ combinations. To simplify the computation of the benchmark, we are interested in the "one-to-double knockdown" scenario and hence generate a total of 56 regimes ($= \frac{10\times 9}{2} + 11$).

The original training datasets for both simulators are discarded and we now consider the simulator as the oracle for any required training set and population functionals. In particular, for each regime,

[14]While it is theoretically possible to use continuous variables and automatic differentiation through a quadrature method that computes each univariate integral for each term in the pseudo-likelihood, this is still far too slow in practice. The discretization level chosen for these examples are fine enough so that it does not appear to affect the predictive performance of the $p(y \mid x)$.

we generate 25,000 samples to obtain a Monte Carlo representation of the ground-truth respective population function $p(x; \sigma)$.

Generating ground-truth outcome processes Y. For outcome variables Y from which we want to obtain $\mu_{\sigma^\star} := \mathbb{E}[Y; \sigma^\star]$ for given test regimes $\sigma^\star$, we consider models of the type $Y = \tanh(\lambda^\top X) + \epsilon_y$, with random independent normal weights λ and $\epsilon_y \sim \mathcal{N}(0, v_y)$. λ and v_y are scaled such that the ground truth variance of $\lambda^\mathsf{T} X$ is a number v_x sampled uniformly at random from the interval $[0.6, 0.8]$, and set $v_y := 1 - v_x$.

For each of the four benchmarks (i.e., based on either the Sachs et al. data or DREAM data, with either a DAG model-based ground-truth or an IFM-based ground truth), we generate 100 random vectors λ. The point of these 100 problems is just to illustrate the ability to learn (noisy) summaries of X, or general downward triggers or markers predictable from X under different conditions. When generating Y from X, *we keep a single sample for* X. We then generate an unique sample for each of the 100 Y variations given the same X data.

Generating training data. For training our models, we additionally generate 5000 samples for the observational regime (baseline) and 500 samples for each of the remaining 4 experimental conditions (Sachs) and 10 experimental conditions (DREAM). To map from X to Y, we use the model described above.

H Further Experimental Results

In Table 2 (see also Fig. 12) we present a series of further experimental results in numerical form based on the following metrics: *i) proportional root mean squared error* (pRMSE): the average of the squared difference between the ground truth Y and estimated $\widehat{Y}$, where each entry is further divided by the ground truth variance of the corresponding Y; and *ii) rank correlation* (rCOR): the Spearman's ρ between the ground truth vector $\mu_{\sigma^\star}$ for all entries in Σ_{test}, and the corresponding estimated vector.[15]

In Table 3, we present results from a series of one-sided binomial tests to determine whether models significantly outperform the black box baseline.

Table 2: Results of our interventional generalization experiments for the Sachs and DREAM datasets. The values correspond to the average of 100 Y problems.

	Sachs		**DREAM**	
	Causal-DAG	*Causal-IFM*	*Causal-DAG*	*Causal-IFM*
	pRMSE	**pRMSE**	**pRMSE**	**pRMSE**
Blackbox	0.043	0.414	0.025	0.174
Causal-DAG	0.014	0.408	0.017	1.337
IFM-1	0.105	0.168	0.022	0.185
IFM-2	0.051	0.111	0.107	0.769
IFM-3	0.123	0.175	–	–
	rCOR	**rCOR**	**rCOR**	**rCOR**
Blackbox	0.696	0.701	0.953	0.930
Causal-DAG	0.873	0.405	0.972	0.502
IFM-1	0.546	0.835	0.952	0.942
IFM-2	0.673	0.821	0.865	0.737
IFM-3	0.503	0.811	–	–

[15]The IFM-3 results are omitted from Tables 2 and 3, as the method does not convergence in a reasonable time.

Table 3: P-values from a series of one-sided binomial tests against the null hypothesis that models perform no better on average than the black box model. Significance at $\alpha = 0.05$ is indicated with one asterisk, and $\alpha = 0.001$ with two.

Data	Simulation	DAG	IFM1	IFM2	IFM3
DREAM	Causal-DAG	$< 0.001^{**}$	0.044^{*}	1	NA
DREAM	Causal-IFM	1	0.972	1	NA
Sachs	Causal-DAG	$< 0.001^{**}$	1	0.998	1
Sachs	Causal-IFM	0.956	$< 0.001^{**}$	$< 0.001^{**}$	$< 0.001^{**}$

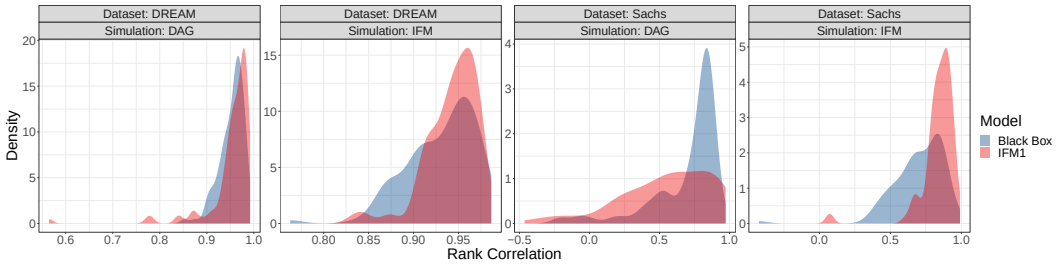


Figure 12: Overlapping density plots showing average rank correlation between true treatment effects and those predicted by the black box model and IFM1, respectively. Ideal performance is a point mass on 1.