# A A Guide to Structure Elicitation and Learning

This section starts with a discussion 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.

This calls for a clarification of what it means to claim

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

as the $\sigma_i$ are not random variables. For what follows, it suffices to interpret this statement as $X_i$ not changing in distribution across different values of $\sigma_1$ when $\sigma_2$ and $X_j$ are fixed/observed at any particular value.

The most basic statement of structure is expressed by a graph $X \rightarrow Y$. 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 $X \rightarrow Y$ 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 $\sigma_x$ and $\sigma_y$ 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 $\sigma_x$) [22]. This is more commonly described as "lack of unmeasured confounding between $X$ and $Y$" or *ignorability*. Moreover, *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 $X \rightarrow Y$, as in Fig. 6**(a)**, as well as the directions from $(\sigma_x, \sigma_y)$ 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's 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 $\sigma_y$, 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 so 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].

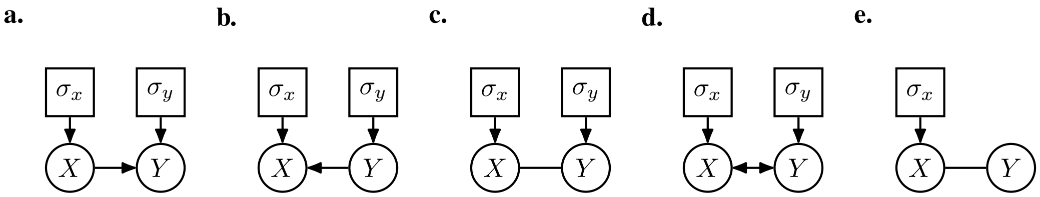

Figure 6: The meaning of $X \to Y$. **(a)** Expressing it when interventions on both $X$ and $Y$ are defined via variables $\sigma_x$ and $\sigma_y$, and we wish to express that $Y \perp\!\!\!\perp \sigma_x \mid X$ and $X \perp\!\!\!\perp \sigma_y$. **(b)-(d)** None of these graphs respect the two independence constraints on interest. **(e)** If $\sigma_y$ 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

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

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's 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 $\sigma_x$ 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 $\sigma_x$ "not causing" $Z$):

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

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], that there is some conditioning set $C$ so 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 any reasonable model 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, symbolic algorithms based on graph-theoretical concepts provide a way to

---

[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.

derive such implications more easily and transparently when compared to relying on algebra alone. 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 independent of its non-parental 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 4-cycle: the independencies encoded in the undirected graph $X_1 - X_2 - X_3 - X_4 - X_1$ cannot be represented by any DAG, even one with the same adjacencies e.g. $X_1 \rightarrow X_2 \rightarrow X_3 \leftarrow X_4 \leftarrow X_1$ (the converse is also true, the independencies encoded by that directed acyclic "diamond structure" 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]. 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 the following:

1. *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 by Figure 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;
2. *"explaining away"*: this well-known phenomenon is illustrated by independencies that get destroyed by conditioning upon further evidence. For instance, 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 $\sigma_x \rightarrow X \leftarrow Z$, 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 $\sigma_x$ 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 $\sigma_x$ 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 of Figure 1. Given the snapshot, some explaining away can happen between $\sigma$ and pre-treatment "Past" variables (Figure 1(b)) that happen to be recorded, but this can be simply accounted for by including the 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 where 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 (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 in models motivated by equilibrium equations. For instance, consider this example which can be found in [11], Section 3.1.1, where $f$ denotes differential equations at equilibrium, $X$ denotes observed random variables, $U$ denotes (mutually independent) latent variables and $I$ denotes an intervention indicator:

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

After marginalizing the $U$ variables, what we get is an IFM (this is not the whole story, though, 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, and interventions denote change to particular constraints. A stochastic differential equation (SDE) model may have *other* assumptions on top of the factorization, and parameters which carry particular meaningful interpretations, but 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. [47] has further elaborations on some of these ideas.

Notice 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. Intervention generalization here should categorically *not* be interpreted as extrapolating to different, unsampled, time points in a non-stationary process.

Finally, 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 $\sigma_i$ not depending on $\sigma_j$[12].

In our case, we will interpret statements

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

where $\sigma_{\backslash ij}$ are all intervention indicators other than $\sigma_i$ and $\sigma_j$, by merely linking it to the factorization

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

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 $p(x; \sigma_i, \sigma_j, \sigma_{\backslash ij})$ as a function of $\sigma$, as the normalizing constant will in general depend on all regime indicators (but not the data). It however denotes the difference between Figure 2(b) and Figure 2(c): the latter is a chain graph with an undirected component that does not suggest factorization over $\sigma_1$, $\sigma_2$ and $\sigma_3$ for any given $x$, while the factor graph explicitly encodes that.

### A.3 Structure Elicitation and Learning

Having agreed that a undirected/factor graph model structure is the natural choice under the scenario described above, we are left to describe how to extract structural knowledge from an expert or algorithm.

Simply put, removing edges from $\sigma$ 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 $Y \perp\!\!\!\perp \sigma_x \mid X, Z$, or plain 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 $\sigma$ to $X$ (that is, forbidding any factor to

---

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

include particular combinations of intervention and random variables) should follow from knowledge about the domain, the impossibility of some direct connections in all sorts of systems, from physical ones (including spatial systems [5]) to social ones (e.g., [59]).

Independencies of the type $\sigma_i \perp\!\!\!\perp \sigma_j \mid X, \sigma_{\backslash ij}$ 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].

None of that is to say that the work of structure elicitation is straightforward. We will conclude with broad ideas about structure learning.

Structure learning can aid the process, and 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 random variables against intervention variables, by querying an independence oracle under a particular regime $\sigma^0$ which we assume all other regimes should be faithful to. For instance, this takes place when $\sigma^0$ 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 $\sigma$ and assessing (conditional) equality in distribution for particular random variables will remove undirected edges between $\sigma$ and $X$ by assuming they will also be unnecessary under any other configuration of the unchanged variables.

Finally, edges "within" $\sigma$ vertices, 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.

A fully detailed account of structure learning for IFMs will be provided in future work. An early method for structure learning of probabilistic factor graph models is described by [1].

## 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, concluding with proofs for the purely algebraic case. 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 directly sampled from and added to $\Sigma_{\text{train}}$, in order to reduce the estimation error coming from particular product/ratios which 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 complete (a clique) and separates $A$ from $B$. The decomposition is *proper* if both $A$ and $B$ are non-empty. Moreover, an undirected graph is *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 $V_i \cap V_j$. 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 $\sigma_{F_k}$ in the original factor graph.

Let $\sigma^{[Z(w)]}$ be a particular instantiation of $\sigma$ where $Z \subseteq [d]$ and $\sigma_Z = w$ (possibly a vector), with the remaining entries of $\sigma$ being zero. For instance, if $d = 3$, $Z = \{2, 3\}$, and $w = (2, 1)$, then $\sigma^{[Z(w)]} = (0, 2, 1)$. Vector $w$ is allowed to include zero values. 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 $\sigma$ in the $\sigma$-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 $\sigma_i$ as being constants in $\sigma_i$, with scope $S_k$ being redefined as $S_{k'} := S_k \backslash \{\sigma_i\}$. $\Sigma_{\text{train}}$ 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

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

with the product over $ch(k)$ defined to be 1 if $ch(k) = \emptyset$. We will show how (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 show as well 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 $\Sigma_{\text{train}}$, showing that Eq. (7) holds for the leaf vertices of $\mathcal{T}$. Likewise, message $m_k^x$ 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 $h_k := f_k(x; \sigma_{F_k \cap B_k} = 1, \sigma_{F_k \backslash B_k} = 0)$ 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

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

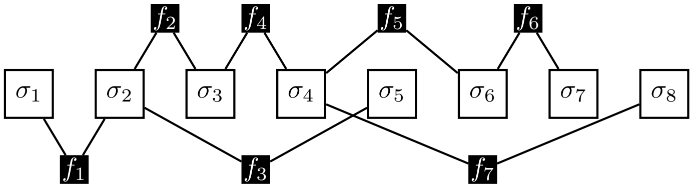

Figure 7: The factor graph used as an illustration of the technique in the proof of Theorem 3.1.

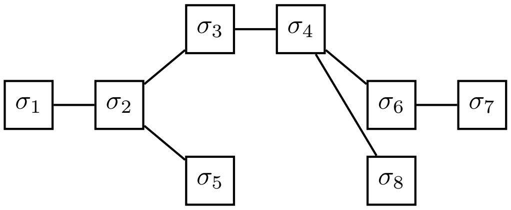

Figure 8: The $\sigma$-graph corresponding to the factor graph in Figure 7.

Moreover, the product

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

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, notice that the numerator sets the $\sigma_{F_j}$ 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 $\Sigma_{\text{train}}$ 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.** Let's 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 will generate the regime at $\sigma_1 = \sigma_2 = \cdots = \sigma_8 = 1$. For reference, this means considering the following sets implied by the factorization above:

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

To illustrate how message passing will work, let's introduce some symbols so that the steps are easier to follow. Let $g^{11}_{ij}$ represent a factor with $\sigma_i = \sigma_j = 1$. For instance, $g^{11}_{12} = f_1(x; \sigma_1 = 1, \sigma_2 = 1)$.

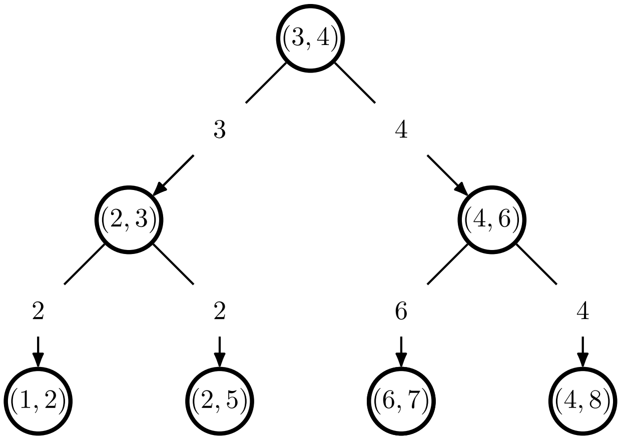

Figure 9: A (directed) junction tree corresponding to the undirected graph in Figure 8.

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 $h^{01}_{ij}$ and $h^{10}_{ij}$ denote assignments $(\sigma_i, \sigma_j) = (0, 1)$ and $(\sigma_i, \sigma_j) = (1, 0)$, respectively. Finally, let $z^{00}_{ij}$ 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

$$\begin{array}{rcll} m^x_1 & = & p(x; [12])/p(x; [2]) & (D_1 = \{1, 2\}, B_1 = \{2\}) \\ m^x_3 & = & p(x; [25])/p(x; [2]) & (D_3 = \{2, 5\}, B_3 = \{2\}) \\ m^x_6 & = & p(x; [67])/p(x; [6]) & (D_6 = \{6, 7\}, B_6 = \{6\}) \\ m^x_7 & = & p(x; [48])/p(x; [4]) & (D_7 = \{4, 8\}, B_7 = \{4\}) \end{array}$$

It can be readily verified that all of these are identifiable from $\Sigma_{\text{train}}$, 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

$$\begin{aligned} m^x_1 \times m^x_3 &= \frac{p(x; [12])}{p(x; [2])} \frac{p(x; [25])}{p(x; [2])} \\ &\propto \frac{g^{11}_{12} h^{10}_{23} h^{10}_{25} z^{00}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}}{h^{01}_{12} h^{10}_{23} h^{10}_{25} z^{00}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}} \times \frac{h^{01}_{12} h^{10}_{23} g^{11}_{25} z^{00}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}}{h^{01}_{12} h^{10}_{23} h^{10}_{25} z^{00}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}} \end{aligned}$$

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:

$$\begin{aligned} m^x_1 \times m^x_3 &= \frac{p(x; [12])}{p(x; [2])} \frac{p(x; [25])}{p(x; [2])} \\ &\propto \frac{g^{11}_{12} \cancel{h^{10}_{23}} \cancel{h^{10}_{25}} \cancel{z^{00}_{34}} \cancel{z^{00}_{46}} \cancel{z^{00}_{67}} \cancel{z^{00}_{48}}}{h^{01}_{12} \cancel{h^{10}_{23}} \cancel{h^{10}_{25}} \cancel{z^{00}_{34}} \cancel{z^{00}_{46}} \cancel{z^{00}_{67}} \cancel{z^{00}_{48}}} \times \frac{\cancel{h^{01}_{12}} \cancel{h^{10}_{23}} g^{11}_{25} \cancel{z^{00}_{34}} z^{00}_{46} z^{00}_{67} z^{00}_{48}}{\cancel{h^{01}_{12}} \cancel{h^{10}_{23}} h^{10}_{25} \cancel{z^{00}_{34}} z^{00}_{46} z^{00}_{67} z^{00}_{48}} \times \frac{g^{11}_{23}}{g^{11}_{23}} \times \frac{h^{10}_{34}}{h^{10}_{34}} \\ &= \frac{g^{11}_{12} g^{11}_{23} g^{11}_{25} h^{10}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}}{h^{01}_{12} g^{11}_{23} h^{10}_{25} h^{10}_{34} z^{00}_{46} z^{00}_{67} z^{00}_{48}} \\ &\propto \frac{p(x; [1235]}{p(x; [23])}. \end{aligned}$$

As $p(x; [23])$, $m^x_1$ and $m^x_3$ 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 $m_2^x$ that 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 $\Sigma_{\text{train}}$ 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. □

**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

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

for all $x$.

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

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

which implies

$$\sum_{k=1}^{l} z_{k^\star} \left( \sum_{i=1:\sigma_{F_k}^i = \sigma_{F_k}^\star}^{t} q_i \right) + \sum_{k=1}^{l} \sum_{\sigma_{F_k}^v \in \mathbb{D}_k \backslash \{\star\}} z_{kv} \left( \sum_{i=1:\sigma_{F_k}^i = \sigma_{F_k}^v}^{t} q_i \right) = 0.$$

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 $\sigma^i$ for every possible combination of $\sigma_{F_k}$, 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 $\sum_{i=1:\sigma_{F_1}^i=(1,1)}^t q_i$ evaluates to $0$, implying $0 = 1$.

**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] propose a more computationally tractable version that they call the "split conformal" method, and derive 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 $\hat{f}(x)$ to estimate $\mathbb{E}[Y \mid x]$ using samples from $\mathcal{I}_1$ and consider the score function $s^{(i)} = |y^{(i)} - \hat{f}(x^{(i)})|$ for $i \in \mathcal{I}_2$. Let $\hat{\tau}$ be the $q^{\text{th}}$ smallest value in $S$, with $q = \lceil (n/2+1)(1-\alpha) \rceil$. Define $\hat{C}(x) = \hat{f}(x) \pm \hat{\tau}$. (We assume symmetric errors for convenience; the result can easily be modified by invoking the appropriate quantiles of the residual distribution.)

**Theorem B.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:*

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

*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 B.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 $\tilde{w}^{(i)}(x)$ 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 $\sum_i \tilde{w}^{(i)}(x) \cdot \delta(s)^{(i)}$, where $\delta$ denotes the Dirac delta function. Let $\hat{\tau}(x)$ be the $q^{\text{th}}$ smallest value in $\sum_i \tilde{w}^{(i)}(x) \cdot \delta(s)^{(i)}$, with $q$ defined as above. Then we construct the weighted conformal band $\hat{C}_w(x) = \hat{f}(x) \pm \hat{\tau}(x)$ for all $x \notin \mathcal{I}_1$.

**Theorem B.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 $n + 1$:*

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

*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 $\sigma$, not $X$, and our result is simply a special case of the split weighted conformal inference theorem. □

# 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 $\sigma_i$ individually, but it 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 e.g. [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 the split of $\sigma$ 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 $\Sigma_{train}$ 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 instance, with binary treatments, a $\sigma$-graph without any edges (no interaction of intervention variables in a same factor), and with 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 $\sigma$-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 $\Sigma_{train}$ 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 $\sigma_i$ is very high? Without smoothness assumptions, getting a reasonable dose-response pattern with few evaluations of $\sigma_i$ 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 $\sigma$ 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 $\sigma$ or via partial identification procedures, see e.g. [35]). There is no free lunch.

# D 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 $\mathcal{D}^1, \dots, \mathcal{D}^t$, 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 describe for the direct method.

When generating an estimate $\hat{\mu}_{\sigma^\star}$, 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 $f_y(x;\hat{\theta}_y)$ 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 $\Sigma_{\text{test}}$ (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.

# E Experimental Details

In this section, we present further experimental details for Section 5, including setup for the datasets (Sachs and DREAM), oracular simulators (Causal-DAG and Causal-IFM), generating ground truth $X$ and $Y$, model implementation details, and complete training process for our experiments.

## E.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

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 E.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 $\sigma$ |
|---|---|---|---|
| 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] |

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 $\Sigma_{\text{train}}$. Hence, we need a way of establishing a (synthetic) ground truth for the $16 - 5 = 11$ possible test conditions, which we explain in Section E.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 $\sigma_i$. 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 $\sigma_i$, 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.

### E.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

---

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

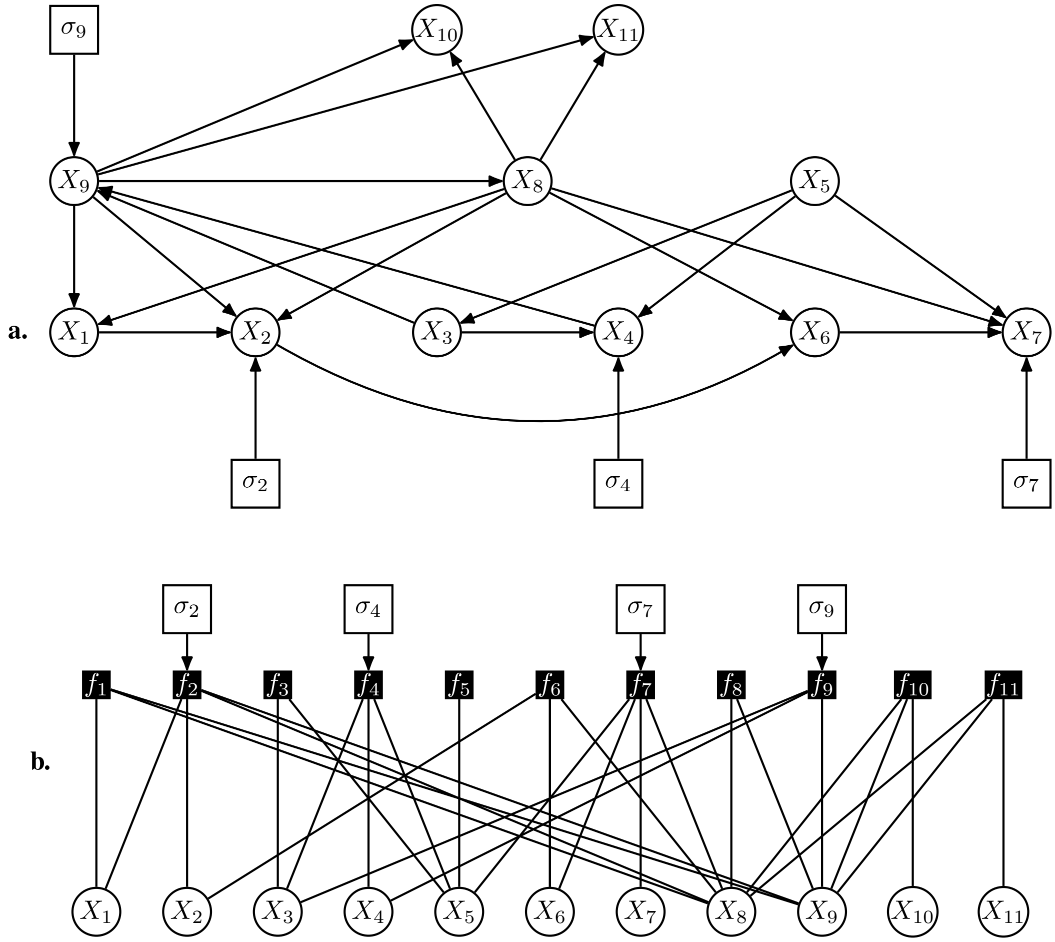

Figure 10: Causal structures used for building the synthetic ground-truth models for the Sachs et al. [66] data. The name of the random variables are $CD3CD28off$, $ICAM$-2, $Akt$-$inhibitor$, $G0076$, $Psitectorigenin$, $U0126$, $LY294002$, $PMA$, $B2camp$, with more details given in the companion Julia 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).

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

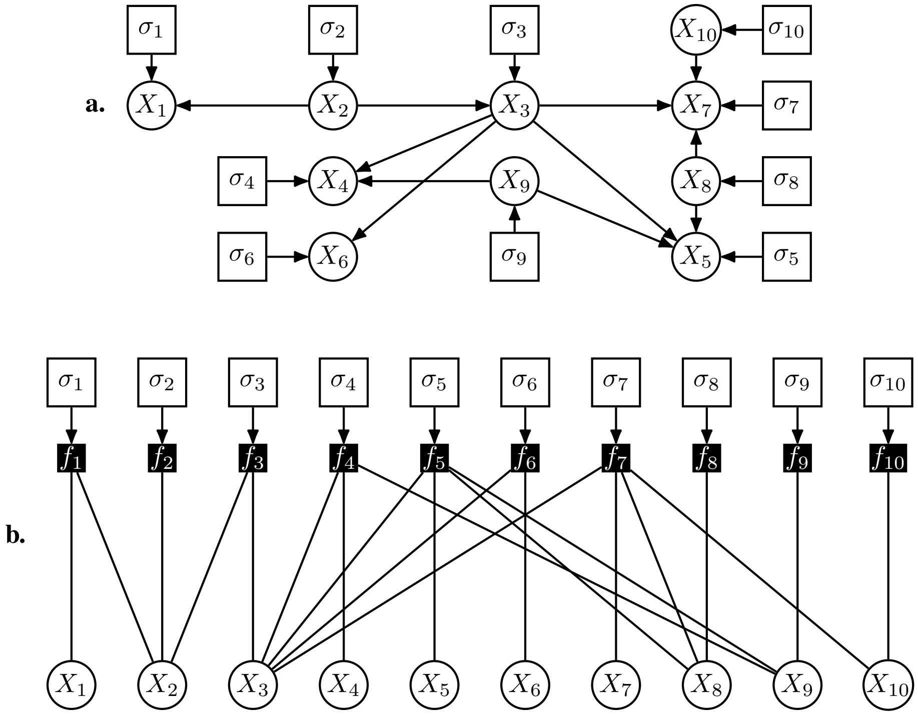
a.
$\sigma_1$
$\sigma_2$
$\sigma_3$
$X_{10}$
$\sigma_{10}$
$X_1$
$X_2$
$X_3$
$X_7$
$\sigma_7$
$\sigma_4$
$X_4$
$X_9$
$X_8$
$\sigma_8$
$\sigma_6$
$X_6$
$\sigma_9$
$X_5$
$\sigma_5$
b.
$\sigma_1$ $\sigma_2$ $\sigma_3$ $\sigma_4$ $\sigma_5$ $\sigma_6$ $\sigma_7$ $\sigma_8$ $\sigma_9$ $\sigma_{10}$
$f_1$ $f_2$ $f_3$ $f_4$ $f_5$ $f_6$ $f_7$ $f_8$ $f_9$ $f_{10}$
$X_1$ $X_2$ $X_3$ $X_4$ $X_5$ $X_6$ $X_7$ $X_8$ $X_9$ $X_{10}$

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

with disparate scales. For the Sachs data, we also take the logarithm of each random variable prior to standardization.

### E.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 $x_k$, $\sigma_{F_k}$ 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 $\sigma_k$ is just a switch: for each value of $\sigma_k$, 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.

### E.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 **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 $\sigma_{F_k}$ 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 $\mu_\sigma$, we use Gibbs sampling.

### E.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 $\Sigma$ 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, 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 $\lambda$ and $\epsilon_y \sim \mathcal{N}(0, v_y)$. $\lambda$ 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 $\lambda$. 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.

### E.4 Implementation Details

We now describe the implementation details, which are also detailed in the companion source code.

---

[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 $p(y \mid x)$.

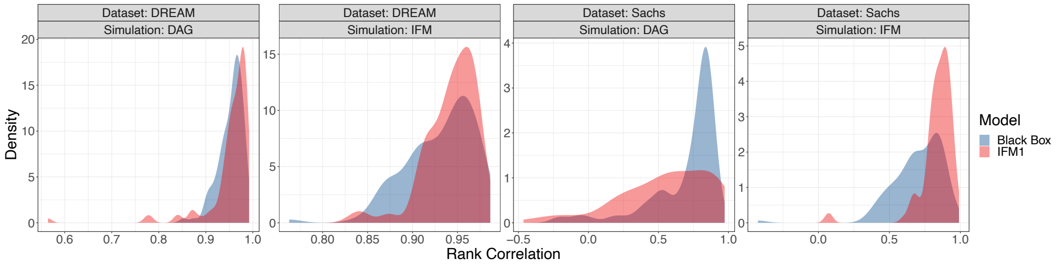

Figure 12: Overlapping density plots showing average rank correlation between true treatment effects and those predicted by the black box model and IFM1, respectively. Ideal performance is a point mass on 1.

Table 2: Experimental results for Sachs and InSilicoSize10-Ecoli1 datasets for our interventional generalization experiments. The values are 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 | – | – |

- **Blackbox Model**: We use the Julia wrapper of XGBoost[15]. In practice, given how sparse $\Sigma_{\text{train}}$ is in the space $\Sigma$ of possible combinations, this is hardly more effective than linear regression (results not shown);
- **Causal DAG**: We set each heteroscedastic MLP model with a hidden dimension of 10 and this is the same setting we used for the Causal DAG simulator. This gives this competitor much advantage in the benchmarks generated by DAGs, as following a parametric Gaussian with additive error structure is already substantive information to be exploited;
- **IFM**: We implemented the IFM model with a combination of neural factors, where each factor is determined by the DAG structure and each MLP has 25 hidden units. Note that the number of hidden units does not match the one used to generate the data.

# F Further Experimental Results

## F.1 Further Experimental Metrics

We present a series of further experimental results in numerical form based on the following metrics: (1) *proportional root mean squared error* (pRMSE): the average of the squared difference between the ground truth $Y$ and estimated $\hat{Y}$, where each entry is further divided by the ground truth variance of the corresponding $Y$; and (2) *rank correlation* (rCOR): the Spearman's $\rho$ between the ground truth vector $\mu_{\sigma^\star}$ for all entries in $\Sigma_{\text{test}}$, and the corresponding estimated vector (see Table 2[16]).

---

[15] https://juliapackages.com/p/xgboost.

[16] 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 | $< 0.001^{**}$ | $< 0.001^{**}$ | $< 0.001^{**}$ |

### F.2 Binomial Tests

We present results from a series of one-sided binomial tests to determine whether models significantly outperform the black box baseline (see Table 3).

## G Pseudolikelihood Details

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

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

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 $x_{\backslash r}^{i(j)}$ for the same data point is composed of all other random variables but the $r$-th variable.

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

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

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, ${x'_{S_k}}^{i(j)}$ 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 $\theta$ 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.