DeCaFlow: A deconfounding causal generative model

Alejandro Almodóvar[1,*] **Adrián Javaloy**[2,*]

Juan Parras[1] **Santiago Zazo**[1] **Isabel Valera**[3]

[1]Information Processing and Telecommunications Center, Universidad Politécnica de Madrid, ES
[2]School of Informatics, University of Edinburgh, UK
[3] Department of Computer science, Saarland University, DE

Abstract

We introduce DeCaFlow, a deconfounding causal generative model. Training once per dataset using just observational data and the underlying causal graph, DeCaFlow enables accurate causal inference on continuous variables under the presence of hidden confounders. Specifically, we extend previous results on causal estimation under hidden confounding to show that a single instance of DeCaFlow provides correct estimates for all causal queries identifiable with do-calculus, leveraging proxy variables to adjust for the causal effects when do-calculus alone is insufficient. Moreover, we show that counterfactual queries are identifiable as long as their interventional counterparts are identifiable, and thus are also correctly estimated by DeCaFlow. Our empirical results on diverse settings—including the Ecoli70 dataset, with 3 independent hidden confounders, tens of observed variables and hundreds of causal queries—show that DeCaFlow outperforms existing approaches, while demonstrating its out-of-the-box applicability to any given causal graph.

1 Causal generative models and hidden confounding

Causal inference seeks to determine how changes in one variable affect others, which is crucial to evaluate the effects of interventions in fields such as healthcare [15], marketing policing [74] or education [85]. In real-world scenarios, where empirical trials often are infeasible due to ethical, financial, or practical constraints, answering causal queries from observational data becomes essential. However, this is a challenging task, especially in the presence of unmeasured or hidden confounders affecting a subset of the observed variables [1, 20].

In this work, we aim to propose a practical approach for accurate causal inference on continuous variables under the presence of hidden confounders. To this end, we build on two key concepts: **i)** *causal generative models* (CGMs) [9, 27, 31], a class of generative models that can generate samples not only from the observational distribution, but also from interventional and (in some cases) counterfactual distributions; and **ii)** *proxy variables*, i.e., conditionally independent variables that yield information about the hidden confounders [45, 46, 46, 76]. Consequently, we introduce the *deconfounding causal normalizing flow* (DeCaFlow), a CGM which provides correct estimates of a broad class of interventional and counterfactual queries under hidden confounding, requiring only observational data, the causal graph, and training once per dataset. Architecturally, DeCaFlow resembles variational autoencoders [33] as it is trained with the ELBO and comprises: **i)** a causal normalizing flow (CNF) [27] as "decoder", adapted to be conditioned on the (potentially many) hidden confounders; and **ii)** a conditional normalizing flow [79] as "encoder", computing the modeled posterior distribution of the hidden confounders given the observations.

[*]Equal contribution. Correspondence to alejandro.almodovar@upm.es and ajavaloy@ed.ac.uk.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

We proved theoretically that *DeCaFlow yields correct estimates for both interventional and counterfactual queries that are identifiable with do-calculus, leveraging the information of proxy variables when do-calculus alone is insufficient.* To that end, we first extend recent advances in proximal causal inference by Miao et al. [45] and Wang and Blei [76] to include counterfactual causal queries. Then, we integrate proximal-identifiability with do-calculus, expanding the number of identifiable queries of which DeCaFlow is shown to provide correct estimates.

As proof of the flexibility that DeCaFlow offers, Fig. 1 illustrates the causal graph of the Ecoli70 dataset [66], comprising 43 observed variables and 3 hidden confounders, showing that DeCaFlow can effortlessly scale to complex settings and accurately recover diverse causal effects after a single training process. Remarkably, green edges in the figure represent direct causal effects that DeCaFlow can identify, despite the presence of hidden confounders. We additionally, provide algorithms to help practitioners easily check in the given the causal graph whether a particular query of interest can be correctly estimated by DeCaFlow.

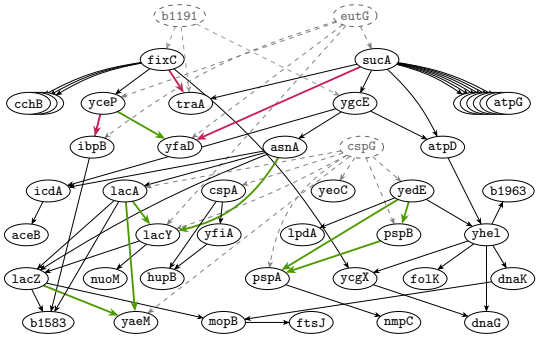


Figure 1: **DeCaFlow can be effortlessly applied to highly complex causal graphs**, as that of the Ecoli70 dataset [66], with multiple hidden confounders and dozens of variables. We dash hidden confounders, and highlight direct *hidden-confounded* effects as identifiable (and thus correctly estimated by DeCaFlow), or unidentifiable.

Finally, we empirically validate all our claims on semi-synthetic and real-world experiments, demonstrating that DeCaFlow outperforms existing alternatives while being widely applicable. An implementation of DeCaFlow can be found in `github.com/aalmodovares/DeCaFlow`.

1.1 Related works

We briefly discuss relevant works in the literature, and defer the reader to §E for further details.

Causal generative models. As mentioned above, we refer as CGMs to the class of generative models that can generate samples from the observational, interventional and, in some cases, counterfactual distributions. A common recipe to build causally consistent CGMs consists of modeling each variable as a function of its causal parents with an independent model. In terms of the choice for modeling these functions, prior works range from simple yet well-established additive noise models [24], to more complex but powerful diffusion-based causal models [9], among others [35, 51, 52, 59, 83]. Due to their sequential nature, these approaches can overfit and propagate errors to downstream variables. Alternatively, recent works have explored using a single (structurally-constrained) network to model the SCM at once, e.g., using normalizing flows [27, 31], or graph neural networks [65, 84]. However, all the aforementioned approaches assume *causal sufficiency*, i.e. the absence of hidden confounders, limiting their applicability in settings with hidden confounding.

Causal inference hidden confounding. When dealing with hidden confounding, many approaches handle only interventional queries and are tailored to a specific causal graph and a single treatment-outcome pair, requiring us to train one model for each query we want to answer. Prior works exploit instrumental variables [4], mediators [55], and, more recently, proxy variables to account for hidden confounding [2, 28, 37, 39, 44–46, 76], from which we build upon later in §4. Recent works have aimed to unify causal inference and generative modeling under hidden confounding [48, 80, 81]. In particular, Neural Causal Models perform causal identifiability and estimation under hidden confounding on discrete variables [81, 82] by, given a causal query, training two "adversarial" models and returning their estimation if they coincide. The model by Nasr-Esfahany et al. [48], instead, focuses on counterfactual queries for simple causal graphs where adjustment sets or instrumental variables are available. Our work thus aims to complement this line of work by providing a *practical and scalable* GCM to solve a broad class of causal queries, interventional or counterfactual, on continuous variables and large causal graphs with a single end-to-end training.

2 Confounded structural causal models

Definition 1. A *(confounded) Structural Causal Model (SCM)* is a triplet $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ describing a data-generating process over D observed (endogenous) variables $\mathbf{x} \coloneqq (\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_D) \in \mathcal{X}$:

$$\mathrm{x}_i \coloneqq f_i(\mathrm{pa}(i), \mathrm{u}_i, \mathbf{z}) \quad \text{for} \quad i = 1, 2, \dots, D\,, \text{ with } \mathbf{u} \coloneqq (\mathrm{u}_1, \mathrm{u}_2, \dots, \mathrm{u}_D) \sim P_{\mathbf{u}}\,,\ \mathbf{z} \sim P_{\mathbf{z}}\,, \quad (1)$$

where f_i represents the structural equation to compute the i-th endogenous variable, x_i, from its observed *causal parents*, $\mathrm{pa}(i)$, the i-th exogenous variable, u_i, and the *hidden confounders*, $\mathbf{z} \in \mathcal{Z}$.

While we make the dependence on the hidden confounders explicit for all observed variables in Eq. 1, we assume w.l.o.g. that a subset of them may not be directly affected by the hidden confounders. Furthermore, given a SCM $\mathcal{M}$, we denote by $\mathcal{G}$ the *faithful* causal graph that it induces, representing a direct causal relationship between pairs of endogenous and hidden variables *only* if it exists.

A core idea of causal inference is the do operator [56], denoted by $\mathrm{do}(\mathrm{t})$, which formalizes the action of externally intervening on the variable t, i.e., to fix t to a value independently of its parents. The do operator enables the computation of interventional and counterfactual queries in SCMs [58]:

Definition 2. A *causal query* $Q(\mathcal{M}) \coloneqq p_{\mathcal{M}}(\mathrm{y} | \,\mathrm{do}(\mathrm{t}), \mathbf{c})$ is a distribution over $\mathrm{y} \in \mathbf{x}$ (the *outcome* variable), as a result of intervening upon the variable $\mathrm{t} \in \mathbf{x}$ (the *treatment* variable). Additionally, $Q(\mathcal{M})$ denotes an *interventional* or *counterfactual* query if the variable $\mathbf{c}$ is, respectively, the empty set or the vector of observed factual values, $\mathbf{x}^{\mathrm{f}}$.

We call a causal query *identifiable* if it can be expressed as a function of the observational distribution, $p_{\mathcal{M}}(\mathbf{x})$, and the causal graph $\mathcal{G}$ [55]. As a result, any SCM inducing the same graph and matching the observational distribution produces correct estimates of that causal query. Moreover, *any* identifiable query can be rewritten this way using a set of three rules, the *do-calculus* [54]. Yet, in the presence of *hidden confounders*, this may not be possible and even applying the do-operator to evaluate causal queries would produce incorrect estimates, as unaccounted confounders would bias the results.

3 Deconfounding causal normalizing flows

In this work, we assume the existence of an underlying confounded SCM, $\mathcal{M}$, as in Def. 1, of which we have access to N i.i.d. observations as well as to the faithful causal graph, $\mathcal{G}$. Our objective is to design and learn a CGM that can *accurately estimate as many causal queries from the original SCM as possible*, despite the presence of unobserved hidden confounding. In other words, we seek a substitute model of $\mathcal{M}$ that we can use to accurately perform causal inference.

Assumptions. In addition, we assume all variables to be continuous, and the SCM $\mathcal{M}$ to: **i)** have C^1-diffeomorphic causal equations conditioned on $\mathbf{z}$, and **ii)** induce an acyclic causal graph $\mathcal{G}$.

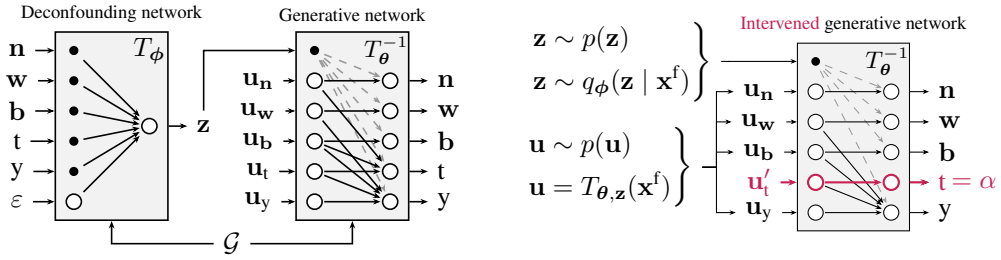
Note that assumption **i)** implies that $\mathbf{f} : \mathcal{U} \times \mathcal{Z} \to \mathcal{X}$ is invertible from $\mathbf{x}$ to $\mathbf{u}$, given $\mathbf{z}$. This is not a limiting assumption, since we never observe $\mathbf{u}$ and we can always find an invertible mapping by merging all $\mathbf{u}$ producing the same observations and taking their Knöthe-Rosenblatt transport [34, 62], while remaining causally equivalent to the original SCM assuming all other assumptions hold.

3.1 (Unconfounded) Causal normalizing flows

Causal normalizing flows (CNFs) [27] play an important role in this work, as they form the foundations of DeCaFlow, given their identifiability guarantees despite a mild set of assumptions. Given a causal graph $\mathcal{G}$, a CNF, $T_{\boldsymbol{\theta}}$, is a masked autoregressive normalizing flow [50] built such that, paired with a distribution $P_{\mathbf{u}}$, defines an unconfounded SCM $\mathcal{M}_{\boldsymbol{\theta}} = (T_{\boldsymbol{\theta}}, P_{\mathbf{u}})$ that induces graph $\mathcal{G}$ by design.

As demonstrated by Javaloy et al. [27], CNFs represent a remarkable family of CGMs, as they not only form a parametric family of *identifiable SCMs*, but they can provably approximate the underlying SCM in the three rungs of Pearl's ladder of causation [55] simply by maximizing the observed joint evidence, i.e., $\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$. Furthermore, CNFs are also equipped with an *exact do-operator* for efficient sampling of any causal query, enabling their use for complex causal-inference tasks.

Their main downside, as discussed in §1, is that CNFs need to assume causal sufficiency—on top of the assumptions made above—to guarantee the aforementioned capabilities, thus limiting their application. Next, we attempt to address this limitation without losing theoretical guarantees.



(a) DeCaFlow training computations.

(b) Computation of causal queries.

Figure 2: **Example of DeCaFlow computations** for the causal graph $\mathcal{G}$ in Fig. A.2. Circles represent input/output variables of the masked conditional normalizing flows, and black dots conditional inputs. **(a)** Steps performed during training (Eq. 4), where ε is a non-causal random variable needed to model $\mathbf{z}$ with T_ϕ. **(b)** Steps performed to compute an interventional or counterfactual query with CNFs [27], see §D, where $\mathbf{u}'_{\mathrm{t}}$ is the value for which $\mathrm{t} = \alpha$ always, i.e., $\mathbf{u}'_{\mathrm{t}} = T_{\boldsymbol{\theta},\mathbf{z}}(T^{-1}_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{u})_{\mathbf{x}\setminus\mathrm{t}}, \alpha)_{\mathrm{t}}$.

3.2 Deconfounding causal normalizing flows

We now introduce the *deconfounding causal normalizing flow* (DeCaFlow), a family of models which extend CNFs [27] to account for hidden confounding while retaining all their theoretical properties. To achieve this, DeCaFlow follows the structure of a variational autoencoder [33], i.e., DeCaFlow comprises two main components: **i)** *a generative network* that exploits structural constraints to accurately model the underlying SCM, given a substitute of $\mathbf{z}$; and **ii)** an *inference network* which approximates the *intractable* posterior distribution of $\mathbf{z}$ as modeled by the generative network, given the observed endogenous variables. In the following, we provide further details on both networks.

Generative network. We adapt CNFs [27] to take the hidden confounders as conditional inputs by using conditional masked autoregressive normalizing flows [79], instead of unconditional ones. The resulting model, $T_{\boldsymbol{\theta}}$, is thus an invertible transformation, conditioned on $\mathbf{z}$, describing a data-generating process that maps a set of exogenous variables $\mathbf{u}$ to endogenous ones and vice versa, i.e., $T_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{x}) = \mathbf{u} \sim P_{\mathbf{u}}$ and $\mathbf{x} = T^{-1}_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{u})$, where we further exploit the given causal graph $\mathcal{G}$ to ensure that the generative process is faithful, i.e., that

$$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) = \prod_{i=1}^{D} p_{\boldsymbol{\theta}}(\mathrm{x}_i \mid \mathrm{pa}(i), \mathbf{z}) \,, \tag{2}$$

defining a data-generating process similar to that of Def. 1 and, just as in that definition, in Eq. 2 *only the children of* $\mathbf{z}$ *are actually conditioned on* $\mathbf{z}$.

Deconfounding network. To model the posterior distribution of $\mathbf{z}$ given our observations as modeled by $T_{\boldsymbol{\theta}}$, i.e., the abduction step needed to compute counterfactuals [55], we use another masked autoregressive conditional normalizing flow [79], as it can approximate this distribution arbitrarily well. Once again, we exploit knowledge of $\mathcal{G}$ and mask the resulting network, $T_{\boldsymbol{\phi}}$, such that it models $\mathbf{z}$ using only the strictly necessary variables to ease learning:

$$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) = q_\phi\left(\mathbf{z} \mid \mathrm{ch}(\mathbf{z}) \cup \mathrm{pa}(\mathrm{ch}(\mathbf{z}))\right) \,. \tag{3}$$

We provide in §C a more general version of Eq. 3 that accounts for several independent hidden confounders, and empirically validate the architecture and factorization choices in §§B.2 and B.3.

Training process. We jointly train both networks as typically done in deep latent-variable models, i.e., we maximize the evidence lower bound (ELBO) [33]:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})] + \mathrm{H}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})) = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})] - \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \| \, p(\mathbf{z})] \,, \tag{4}$$

where KL is the Kullback-Leibler divergence [38], H the differential entropy [36] and $p(\mathbf{z})$ is the prior distribution of $\mathbf{z}$ which we set as a standard Gaussian. The motivation for this loss is three-fold: **i)** we want the generative network to explain the observations given samples from $q_{\boldsymbol{\phi}}$ (first term of Eq. 4); **ii)** as we do not know the optimal size for $\mathbf{z}$, we need to prevent the deconfounding network from allocating information exclusive of $\mathbf{x}$ in $\mathbf{z}$ (entropy term in Eq. 4); and **iii)** all the results introduced

next rely on DeCaFlow matching the observational distribution, $p_{\mathcal{M}}(\mathbf{x})$, which we encourage since

$$\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi},\boldsymbol{\theta}) = \min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathrm{KL}[p_{\mathcal{M}}(\mathbf{x}) \| \, p_{\boldsymbol{\theta}}(\mathbf{x})] + \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \| \, p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})] \,. \tag{5}$$

To avoid posterior collapse, i.e., that the approximate posterior matches the prior and thus having uninformative latent variables [77], we incorporate KL balancing terms [73] to prevent the KL in Eq. 4 from vanishing, ensuring that the latent representation remains informative during training. Other implementation details, e.g., the way masking both encoder and decoder with via the causal adjacency matrix, or how to adapt the do-operator of CNFs can be found in §§C and D, respectively.

3.3 Inherited causal properties

As a consequence of leveraging (causal) normalizing flows, DeCaFlow inherits many of the great properties of this family of models. For once, both components of DeCaFlow are universal density approximators [50] meaning that, given enough resources, the generative network can perfectly match the observational distribution and the deconfounding network can perfectly learn the modeled posterior. In other words, *we can perfectly minimize the two KL terms that appear in* Eq. 5. Furthermore, note that the generative network, combined with two base distributions for $\mathbf{u}$ and $\mathbf{z}$, defines a confounded SCM as in Def. 1, i.e., $\mathcal{M}_{\boldsymbol{\theta}} = (T_{\boldsymbol{\theta}}^{-1}, P_{\mathbf{u}}, P_{\mathbf{z}})$.

Causal consistency. As we leverage the causal graph $\mathcal{G}$ to appropriately mask the conditional CNF of the generative network, we have that $\mathcal{M}_{\boldsymbol{\theta}}$ respects all causal connections described by $\mathcal{G}$. In other words, $\mathcal{M}_{\boldsymbol{\theta}}$ *induces the same causal graph* $\mathcal{G}$ *as* $\mathcal{M}$. As a result, we can ensure which variable affects which when we generate observations with $T_{\boldsymbol{\theta}}$. Fig. 2 depicts these structural constraints relating $\mathbf{u}$ and $\mathbf{z}$ with $\mathbf{x}$, and we provide a detailed description in §C.

Moreover, we prove in §A.1 that DeCaFlow preserves one of the most crucial aspects of CNFs: Identifying (in the sense of Xi and Bloem-Reddy [80]) the underlying SCM concerning those variables that are not directly caused by $\mathbf{z}$:

Proposition 3.1 (Informal)**.** *If DeCaFlow induces the same causal graph and observational distribution as the underlying (confounded) SCM generating the data. Then, DeCaFlow recovers the SCM for every variable not in* $\mathrm{ch}(\mathbf{z})$*, up to an element-wise transformation of their exogenous distributions.*

DeCaFlow do-operator. While the above result ensures the causal equivalence of $\mathcal{M}_{\boldsymbol{\theta}}$ and $\mathcal{M}$ for unconfounded variables, it is still unclear how to intervene on $\mathcal{M}_{\boldsymbol{\theta}}$. To this end, we adapt the do-operator of CNFs [27], represented in Fig. 2b and detailed in §D, which provides an efficient *and exact* way of sampling from any interventional and counterfactual distribution. Namely, to sample from an interventional distribution $p(\mathbf{x}|\,\mathrm{do}(\mathrm{t} \coloneqq \alpha))$ over $\mathcal{M}_{\boldsymbol{\theta}}$ we: **i)** sample $\mathbf{z} \sim P_{\mathbf{z}}$ and $\mathbf{u} \sim P_{\mathbf{u}}$; **ii)** find the value of $\mathrm{u_t}$ that yields $\mathrm{t} = \alpha$ given $\mathbf{u}$, which we can easily do as $T_{\boldsymbol{\theta}}$ is invertible given $\mathbf{z}$; and **iii)** return the sample $\mathbf{x}^{\mathrm{do(t)}} \coloneqq T_{\boldsymbol{\theta},\mathbf{z}}^{-1}(\mathbf{u}_{\mathbf{x}\setminus\mathrm{t}}, \mathrm{u_t})$. The counterfactual case is quite similar, as the bijectivity of $T_{\boldsymbol{\theta}}$ implies that every counterfactual distribution is a delta distribution given $\mathbf{z}$, and we can simply follow the process above but using $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\,\mathbf{x}^{\mathrm{f}})$ and $\mathbf{u} = T_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{x}^{\mathrm{f}})$ for step one. As a result, we can guarantee the correctness of DeCaFlow estimations on a number of causal queries:

Corollary 3.2 (Informal)**.** *DeCaFlow provides correct estimates of any causal, interventional or counterfactual, query for which both the treatment and outcome variables are not direct children of a hidden confounder, i.e.,* $\mathrm{t}, \mathrm{y} \notin \mathrm{ch}(\mathbf{z})$.

Thus far, we have shown that for causal queries over non-children of $\mathbf{z}$, DeCaFlow inherits the theoretical guarantees of CNFs. In the following section, we investigate under which conditions DeCaFlow can also provide correct estimates of causal queries defined over children of $\mathbf{z}$.

4 Estimation of causal queries under hidden confounding

By leveraging recent results in proximal-identifiability, we next show that DeCaFlow not only preserves the properties of CNFs, but expand them. Namely, we characterize queries which DeCaFlow correctly estimates despite the hidden confounding. While we present here an intuitive summary of our main theoretical results, formal statements and derivations can be found in §A. Throughout this section, we define *proxy variables* relative to the causal query to estimate: if we are interested in the causal effect of t over y, which are confounded by $\mathbf{z}$, a proxy is an observed variable related to $\mathbf{z}$ and conditionally independent of t or y (see Prop. 4.1 next for precise definitions). Intuitively, proxy variables contain exploitable information about the hidden confounders, which we leverage in this section to provide accurate estimates of the causal queries of interest.

4.1 Interventional queries

First, we consider the identifiability of *hidden-confounded* interventional queries, i.e., queries of the form $Q(\mathcal{M}) = p_{\mathcal{M}}(\mathrm{y}|\,\mathrm{do}(\mathrm{t}))$, where $\mathrm{y}, \mathrm{t} \in \mathrm{ch}(\mathbf{z})$ are any two children of the hidden confounder. We summarize our findings in the following proposition, which we properly formalize in §A.2:

Proposition 4.1 (Informal)**.** *An interventional query of the form $Q(\mathcal{M}) = p_{\mathcal{M}}(\mathrm{y}|\,do(\mathrm{t}))$, where $\mathrm{y}, \mathrm{t} \in \mathrm{ch}(\mathbf{z})$ are two different children of $\mathbf{z}$, is identifiable if there exists a (potentially empty) subset of blocking variables $\mathbf{b} \subset \mathbf{x} \setminus \{\mathrm{t}, \mathrm{y}\}$, and two other variables $\mathbf{n}, \mathbf{w} \in \mathbf{x} \setminus \{\mathrm{t}, \mathrm{y}, \mathbf{b}\}$ such that:*

1. *$(\mathbf{b}, \mathbf{z})$ forms a valid adjustment set, i.e., $p(\mathrm{y}|\,do(\mathrm{t})) = \iint p(\mathrm{y}|\,\mathrm{t}, \mathbf{b}, \mathbf{z}) p(\mathbf{b}, \mathbf{z})\,\mathrm{d}\mathbf{b}\,\mathrm{d}\mathbf{z}$,*
2. *$\mathbf{w}$ is a proxy variable given $\mathbf{b}$, i.e., $\mathbf{w} \perp\!\!\!\perp (\mathrm{t}, \mathbf{n})|\,\mathbf{b}, \mathbf{z}$,*
3. *$\mathbf{n}$ is a null proxy variable given $\mathbf{b}$, i.e., $\mathrm{y} \perp\!\!\!\perp \mathbf{n}|\,\mathrm{t}, \mathbf{b}, \mathbf{z}$, and*
4. *both $\mathbf{w}$ and $\mathbf{n}$ yield enough information about the hidden confounder $\mathbf{z}$.*

Prop. 4.1 extends the results from Miao et al. [45] and Wang and Blei [76] to prove identifiable of queries under hidden confounding *even if treatment and outcome have observed parents in common*. In turn, these results render causal queries identifiable in the infinite-data regime by leveraging proxy information, complementing classical do-calculus [39]. Intuitively, $\mathbf{w}$ serves the purpose of building a function which "substitutes" the hidden confounder for that query, and $\mathbf{n}$ is used to ensure that this substitute yields the correct estimate. From this result, one natural step is then to extend the class of causal queries which are identifiable using do-calculus, where we introduce the queries identifiable with Prop. 4.1 as an additional base case for the do-calculus recursive steps:

Corollary 4.2. *An interventional query is identifiable if, using do-calculus, it can be reduced to a combination of observational queries and identifiable interventional queries in the sense of Prop. 4.1.*

To understand the implications of Prop. 4.1 and Cor. 4.2, consider the causal graph in Fig. 3, and suppose we want to compute $Q(\mathcal{M}) = p(\mathrm{y}_1|\,\mathrm{do}(\mathrm{t}))$. Then, we can proceed as usual and apply the rules of probability theory and do-calculus to rewrite $Q(\mathcal{M})$ as

$$Q(\mathcal{M}) = p(\mathrm{y}_1 \mid \mathrm{do}(\mathrm{t})) = \int p(\mathrm{y}_1 \mid \mathrm{t}, \mathrm{y}_2) p(\mathrm{y}_2 \mid \mathrm{do}(\mathrm{t}))\,\mathrm{d}\mathrm{y}_2\,. \quad (6)$$

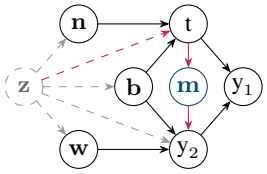


Figure 3: Illustrative causal graph where the presence or absence of some parts render $p(\mathrm{y}_1|\,\mathrm{do}(\mathrm{t}))$ identifiable using do-calculus. Else, Prop. 4.1 yields identifiability if $\mathbf{w}$ and $\mathbf{n}$ are informative proxies.

As a result, the identifiability of $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t}))$ implies that of $Q(\mathcal{M})$. We can then devise a few different scenarios:

1. If there is no edge from $\mathbf{z}$ to t, i.e., $\mathrm{t} \notin \mathrm{ch}(\mathbf{z})$, then the backdoor criterion holds for $\{\mathbf{n}, \mathbf{b}\} = \mathrm{pa}(\mathrm{t}) \subset \mathbf{x}$ and both $p(\mathrm{y}_1|\,\mathrm{do}(\mathrm{t}))$ and $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t}))$ are identifiable.
2. If there exists a mediator variable between t and y_2, $\mathbf{m}$, we can apply the front-door adjustment and both $p(\mathrm{y}_1|\,\mathrm{do}(\mathrm{t}))$ and $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t}))$ are identifiable.
3. If y_2 is not caused by t, then $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t})) = p(\mathrm{y}_2)$ and both queries are identifiable.
4. Otherwise, we can still render $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t}))$ (and thus $p(\mathrm{y}_1|\,\mathrm{do}(\mathrm{t}))$) identifiable if $\mathbf{w}$ and $\mathbf{n}$ yield sufficient information about $\mathbf{z}$ (intuitively, this means that the posterior of $\mathbf{z}$ changes enough as we change $\mathbf{w}$ and $\mathbf{n}$; we formalize this notion in Def. 4) and we can hence apply Prop. 4.1.

The example above nicely illustrates how Prop. 4.1 complements do-calculus: if we find a query unidentifiable due to reaching a dead end with do-calculus—in this case, $p(\mathrm{y}_2|\,\mathrm{do}(\mathrm{t}))$—then Prop. 4.1 provides an additional case for which the query can still be made identifiable. Moreover, this case clearly shows how Prop. 4.1 extends prior results as these *did not allow* for common observable ancestors between outcome and treatment [45, 76]. Nevertheless, note that Prop. 4.1 provides only sufficient conditions for identifiability, and there could exist identifiable queries which do not comply with the requirements of the proposition.

Finally, recall from §3.3 that, similar to CNFs [27], we can readily interpret the generative network of DeCaFlow as a parametric confounded SCM (Def. 1) of the form $\mathcal{M}_{\boldsymbol{\theta}} \coloneqq (T_{\boldsymbol{\theta}}^{-1}, P_{\mathbf{u}}, P_{\mathbf{z}})$. This SCM induces the same causal graph as the underlying $\mathcal{M}$ by design, $\mathcal{G}$, and since the family of normalizing flows are universal density approximators, $\mathcal{M}_{\boldsymbol{\theta}}$ can match the observational distribution $p_{\mathcal{M}}(\mathbf{x})$ given enough resources. We can then leverage the previous results to prove the following:

Corollary 4.3. *If DeCaFlow induces the same causal graph $\mathcal{G}$ as $\mathcal{M}$ and $p_{\mathcal{M}}(\mathbf{x}) \overset{a.e.}{=} p_{\boldsymbol{\theta}}(\mathbf{x})$, then DeCaFlow provides correct estimates of any query identifiable in the sense of Cor. 4.2.*

In other words, Cor. 3.2 guarantees that, if we match the observational distribution with DeCaFlow, then the do-operator presented in §3.3 provides a correct estimate of the identifiable query of interest.

4.2 Counterfactual queries

Next, we focus on counterfactual queries of the form $Q(\mathcal{M}) = p_{\mathcal{M}}(\mathrm{y}^{\mathrm{cf}} | \operatorname{do}(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$, where $\mathbf{x}^{\mathrm{f}}$ is the observed factual. Intuitively, this query represents *the distribution the outcome would have had, had we intervened on the treatment variable*. We demonstrate, for the first time to our knowledge, a one-to-one correspondence between proxy-identifiable interventional and counterfactual queries. More specifically, we show that (all formal derivations can be found in §A.3):

Proposition 4.4 (Informal)**.** *If an interventional query $p(\mathrm{y} | \, do(\mathrm{t}))$ is identifiable in the sense of Prop. 4.1, then its counterfactual counterpart, $p(\mathrm{y}^{\mathrm{cf}} | \, do(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$, is also identifiable.*

The proof of Prop. 4.4 exploits the notion of twin SCM [5], which duplicates the structural equations for the factual and counterfactual worlds while sharing the exogenous variables, and the fact that Prop. A.2 (the formal version of Prop. 4.1) allows for queries with additional covariates as long as they do not form colliders, which is always the case with $\mathbf{x}^{\mathrm{f}}$ in $p_{\mathcal{M}}(\mathrm{y}^{\mathrm{cf}} | \operatorname{do}(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$, as we show in the illustrative twin network of Fig. 4. We can then follow the same derivations from the previous section to show that:

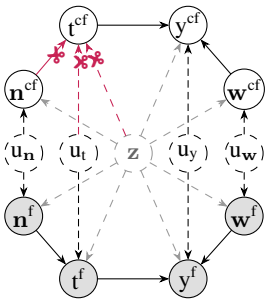


Figure 4: **Twin SCM** with observed factual nodes grayed out and edges severed to compute $p(\mathrm{y}^{\mathrm{cf}} | \operatorname{do}(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$ in red.

Corollary 4.5. *If DeCaFlow induces the same causal graph $\mathcal{G}$ as $\mathcal{M}$ and $p_{\mathcal{M}}(\mathbf{x}) \overset{a.e.}{=} p_{\boldsymbol{\theta}}(\mathbf{x})$, then DeCaFlow provides correct estimates of any counterfactual query which can be decomposed in a combination of (proxy-)identifiable queries using do-calculus.*

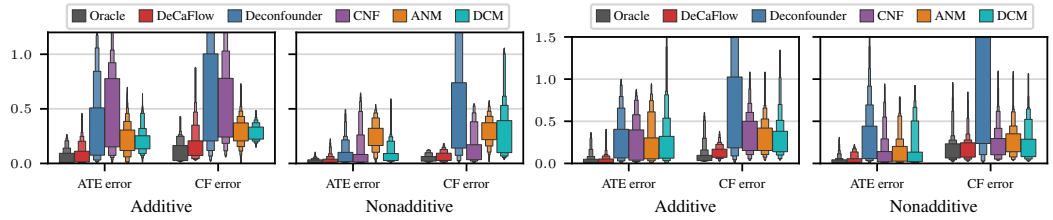
Cor. 4.5 implies that, if DeCaFlow correctly estimates an interventional query, then it also does for its counterfactual counterpart. While the above results can look surprising at first, recall that we assume continuous endogenous variables and diffeomorphic causal generators (§3). Moreover, the correct estimation of counterfactual queries does not come without challenges: **i)** we need to accurately estimate $p_{\boldsymbol{\theta}}(\mathbf{z} | \, \mathbf{x})$, which is why it is crucial to correctly design and train $q_{\boldsymbol{\phi}}$; and **ii)** given $\mathbf{z}$ and $\mathbf{x}$, we need to accurately perform the abduction step. Fortunately, the latter step is trivialized using CNFs as generative networks [27], since they are bijective given $\mathbf{z}$.

Remarks. Whilst DeCaFlow can estimate *any* causal query, *this estimation can be incorrect for unidentifiable queries*. Therefore, we must verify the identifiability for each query of interest, which we aim to ease by providing algorithms to check the graphical requirements of Prop. 4.1 in §F. Namely, Alg. 6 checks if a query that involves a specific treatment-outcome pair, which includes average treatment effects and counterfactuals, is identifiable. If we were interested in a query on all variables, e.g., as samples from an interventional distribution, we should evaluate the identifiability of the causal query for all descendants of the treatment, as proposed in Alg. 7. Similarly, note that all results above rely on the assumption that DeCaFlow matches the observational distribution, and thus it is crucial to ensure that the training completed successfully.

5 Empirical evaluation

In this section, we assess the performance of DeCaFlow relative to existing methods. Namely, we show that DeCaFlow accurately estimates interventional and counterfactual queries when the requirements of Corols. 3.2 and 4.3 are met. All experimental details are described in §B.

Common evaluation. For all experiments, we measure the estimation quality for interventional and counterfactual queries using the mean absolute error (MAE) of, respectively, the average treatment effect (ATE) and the counterfactual (CF) samples, as we have access to the ground-truth values. We use as reference (or *oracle*) a CNF [27] that *does observe* the hidden confounders. We also account for differences across observed variables by computing all errors over the standardized variables. Note that ATE and CF errors provide complementary measures of estimation quality, and their interpretation is best understood relative to the oracle performance in each metric.



(a) Sachs' dataset. (b) Ecoli70 dataset.

Figure 7: ATE and CF error boxenplots [23] of different CGMs on the **(a)** Sachs and **(b)** Ecoli70 datasets, aggregating over all identifiable direct effects (see Figs. 1 and 8) after intervening on their 25th, 50th, and 75th percentiles over 5 random initializations.

Hyperparameter selection. We choose hyperparameters based on the MMD [21] over validation observational data, following our theoretical premise that DeCaFlow correctly estimates causal queries when $p_{\mathcal{M}}(\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x})$; see §B.6.3 for details on the hyperparameter grid.

5.1 Ablation study and practical considerations

In order to provide insights into practical limitations of DeCaFlow, we first conduct an ablation study to understand the extent for which misspecifying the size of $\mathbf{z}$ affects DeCaFlow, as well as its sensitivity to the number of available proxies. For additional details and results, refer to §B.1.

Experimental setup. We consider two synthetic SCMs with linear and non-linear causal equations that follow the causal graph $\mathcal{G}$ depicted in Fig. 5, comprising two independent hidden confounders affecting every variable, and S null proxies. We evaluate how well DeCaFlow estimates $p(\mathrm{y}|\,\mathrm{do}(\mathrm{t}))$ as we change the number of proxy variables, S, and the specified latent dimensionality, $D_{\mathbf{z}}$.

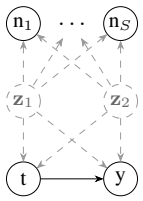


Figure 5: Ablation graph.

Proxy informativeness. The completeness condition (Prop. A.2), i.e., that proxies yield "*enough information*" (Prop. 4.1), is difficult to verify. Fortunately, Fig. 6 shows that using more proxies consistently improves the estimation of confounded causal queries in practice, as it is more likely to satisfy completeness. Thus, practitioners should aim to collect as many informative proxies as possible to ensure correct causal estimates.

Latent dimensionality. When the dimensionality of the hidden confounders is unknown, we expect the entropy term in Eq. 4 to prevent modeling exogenous variables with $\mathbf{z}$, as discussed in §3.2. Fig. 6 corroborates our intuition, showing that DeCaFlow remains robust to overspecification of the latent dimension, $D_{\mathbf{z}}$, while error increases as we underestimate it. This suggests that, in practice, choosing a large latent space is preferable.

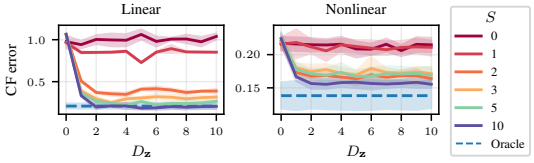


Figure 6: CF error as we increase the number of proxies, S, and the latent dimensionality, $D_{\mathbf{z}}$. Plots show mean and 95 % CI over 5 realizations, intervening on the 25th, 50th, and 75th percentiles of t.

Other ablations. We summarize here other experiments of practical interest that can be found in the appendix. §§B.2 and B.3 corroborate our design choices for the deconfounding network, namely, the use of conditional normalizing flows and the posterior factorization in Eq. 3. Then, §B.4 assess the sample efficiency of DeCaFlow, showing that both DeCaFlow and the oracle monotonically improve their estimations as the training size increases, supporting that correct causal estimates are obtained when the model accurately fits the observational distribution.

5.2 Semi-synthetic experiments

Next, we evaluate how DeCaFlow performs relative to existing approaches. To this end, we consider semi-synthetic datasets for which we have access to the ground-truth. Additional details, results, and a justification on the use of semi-synthetic data can be found in §§B.5 and B.6.

Baselines. We consider three CGMs which assume causal sufficiency and are thus *unaware* of the hidden confounders: CNFs [27]; ANMs [24]; and DCMs [9]; as well as the Deconfounder [75], which uses proxies to provide unbiased ATE estimates under hidden confounding, yet it requires to train one model per outcome. We take the oracle model as a lower bound of the error.

5.2.1 Protein-signaling networks

Following Chao et al. [9], we first experiment with the protein-signaling network dataset [63]. Namely, we randomly generate a non-linear SCM inducing the same causal graph as the original dataset, see Fig. 8, except for the root nodes, for which we use the original data. As a result, we have a bidimensional hidden confounder, `PKC` and `PKA`, and three treatment variables to intervene upon, `Raf`, `Mek`, and `Erk`. We consider additive and non-additive causal equations, measure the effect of interventions on the downstream nodes and, more importantly, ensure when generating the SCM that the randomized effect of the hidden confounder is perceptible.

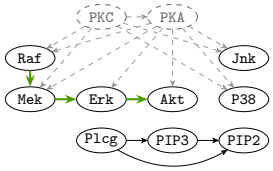


Figure 8: Sachs' graph. Green edges mark proxy-identifiable effects.

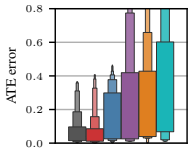
Results. We present a visualization of the results in Fig. 7a, where we can observe that DeCaFlow outperforms every considered approach in all cases, for both ATE and counterfactual errors, *staying on par with the oracle model*. Moreover, we appreciate a great difference in performance between DeCaFlow and CNFs, which corroborates the importance of the architecture and variational training employed by DeCaFlow, since a CNF is equivalent to DeCaFlow with $D_{\mathbf{z}} = 0$.

5.2.2 Gene networks

We next conduct a similar experiment as in the previous section, considering this time the causal graph of the Ecoli70 dataset [66], depicted in Fig. 1, which represents a gene network extracted from E. coli data. This time, we replace root nodes with Gaussian variables. See §B.6.

Results. Similar to the previous case, the results presented in Fig. 7b demonstrate that DeCaFlow is indeed able to closely match the performance of the oracle model, outperforming existing approaches. It is worth-noting, however, that the non-additive case shows long-tailed error distributions for all models, including the oracle, highlighting unavoidable issues of any data-centric approach, also DeCaFlow, and that have to be considered during evaluation and deployment.

We feel compelled to explain that the striking performance of the Deconfounder is an artifact of evaluating on causal queries that it cannot correctly estimate. As we discuss in §B, the Deconfounder guarantees correct ATE estimation under more stringent assumptions than those from §4. If we plot the ATE error evaluated on only those queries that meet Deconfounder's assumptions, we indeed observe that it achieves significantly lower error, as shown in the inset figure.



This experiment highlights every strength of DeCaFlow, as it needs to: **i)** model several hidden confounders affecting different sets of variables; **ii)** correctly estimate all causal queries for which we have proxy information; and **iii)** achieve the above in an agnostic manner, i.e., training the model out-of-the-box and *one single time*, despite the graph $\mathcal{G}$ having 43 observed variables.

5.3 Fairness real-world use case

Taking inspiration from the experiments by Kusner et al. [40] and Javaloy et al. [27] we test whether, by modeling the confounded SCM with DeCaFlow, we can leverage it for more than causal-query estimation and, in particular, for counterfactual-fairness prediction. See §B.7 for further details.

Dataset and objective. Our aim is to train a gradient-boosted decision tree [17] on the law school dataset [78], which comprises of 21 790 law students, that remains accurate while being fair toward the sensitive attributes of the students. In particular, we aim to predict the decile of a student in its 3rd year of university, given their undergraduate and 1st year grades, family income, race, and sex.

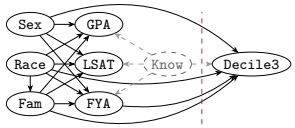


Figure 9: Assumed causal graph in §5.3. Only the classifiers consider `Decile3`.

Experimental setup. First, we train DeCaFlow assuming the causal graph in Fig. 9, excluding `Decile3`, where all grades are affected by a common "knowledge" hidden confounder [40]. Then, we train a predictor using as input the hidden confounder and non-sensitive exogenous variables extracted from DeCaFlow. If, as discussed in §3.3, DeCaFlow successfully recovers the exogenous variables, we expect the predictor to be fair yet slightly less accurate, since `Decile3` is directly affected by the sensitive attributes.

Results. Tab 1 provides the prediction error (RMSE) and the difference between group distributions (MMD) for the proposed DeCaFlow-based predictor, comparing with an Unfair predictor that uses sensitive attributes; an Unaware predictor that excludes sensitive attributes, and two fair predictors, Fair K and Fair Add, as initially proposed by Kusner et al. [40]. We see that the proposed predictor slightly increases the error, while significantly reducing the MMD between the predicted distributions between sensitive attributes. Moreover, the other fair classifiers behave just as a baseline always predicting the average prediction.

Table 1: Test RMSE on `Decile3` prediction and MMD of inter-group predictive distributions.

	Unfair	Unaware	DeCaFlow	Fair K	Fair Add	Mean
RMSE	1.413	1.419	1.604	2.817	2.826	2.83
MMD	0.163	0.147	0.0054	10^{-5}	10^{-4}	0

To provide a better intuition on the differences between predictors, we plot in Fig. 10 the distributions predicted by the gradient-boosted decision tree stratified by the sensitive classes, for the Unfair and DeCaFlow-based classifiers. Fig. 10 shows that, while both classifiers provide similarly-distributed predictions for both sex classes, female and male, we find a qualitatively significant difference between the race classes, white and non-white, with the predictions of the Unfair classifier clearly skewed, predicting much lower deciles for the non-white population. In contrast, the DeCaFlow-based classifier provides much similar predictions for all sensitive classes at the expense of a slightly higher prediction error.

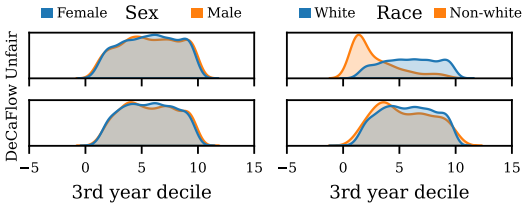


Figure 10: **Distribution of predicted** `Decile3`. A fair predictor yields similar distributions across the considered groups per attribute (`Sex` and `Race`).

6 Concluding remarks

In this work we introduced DeCaFlow, a CGM that can enable accurate estimation of interventional and counterfactual queries under hidden confounding. DeCaFlow expands CNFs, preserving their properties while offering several key advantages over prior works. Namely, DeCaFlow can be applied out-of-the-box to any given causal graph and, training once per dataset, it correctly estimates a broad class of (potentially hidden-confounded) interventional *and counterfactual* queries over continuous endogenous variables. Moreover, we theoretically characterized all queries that DeCaFlow correctly estimates as those for which do-calculus yields observational or proximal-identifiable queries, extending prior results [45, 76] to include counterfactuals and observed common ancestors. Finally, we showed that DeCaFlow outperforms existing methods across a variety of settings, accurately recovering more hidden-confounded causal effects and enabling fair downstream predictions.

Limitations. While DeCaFlow relaxes *causal sufficiency*, it still relies on the existence of sufficiently-informative proxies. This condition is untestable since we do not observe the hidden confounder, but collecting additional proxies can help satisfy it [3], as shown in §5.1. Similarly, DeCaFlow works with continuous random variables by assumption (see §3), although §5.3, Javaloy et al. [27] and de Vassimon Manela et al. [13] show that CNFs effectively approximate discrete distributions in practice. Another limitation is assuming perfect knowledge of the causal graph $\mathcal{G}$. In practice, $\mathcal{G}$ may be partially available or noisy. When graph misspecification does not involve children of the hidden confounders, DeCaFlow inherit from CNFs the ability to operate with a known causal ordering or with partially specified graphs where variables are grouped, see [27, App. A.2.2]. However, if the assumed graph incorrectly specifies the relations involving hidden confounders—and thus violating the assumptions in Prop. 4.1—our identifiability results under hidden confounding no longer hold, and estimates for confounded causal queries may become inaccurate. Alternatively, DeCaFlow could be combined with methods that jointly perform causal identification and estimation for individual queries [82], which handle cases identifiable beyond our theory but trade scalability for flexibility.

Future work. We believe DeCaFlow opens many intriguing venues we are excited to explore, such as expanding the range of queries it can estimate using instrumental variables [22], applying it to settings with time-varying treatments or where multiple interventions take place, investigating the empirical sensitivity of DeCaFlow to noisy or misspecified causal graphs [49], or extending our framework to support soft (stochastic) interventions with dedicated evaluation protocols [10]. We are also excited to see DeCaFlow applied to real-world problems such as decision support systems [64], educational analysis [47], or policy making [16], yet always validating them with interventional data.

Acknowledgments and Disclosure of Funding

The authors would like to thank Luigi Gresele for useful discussions and comments which helped improving the quality of this work. This work has been supported by the project *"Society-Aware Machine Learning: The paradigm shift demanded by society to trust machine learning,"* funded by the European Union and led by IV (ERC-2021-STG, SAML, 101040177); and the Deutsche Forschungsgemeinschaft (DFG) grant number 389792660 as part of the Transregional Collaborative Research Centre TRR 248: Center for Perspicuous Computing (CPEC) (TRR 248 – CPEC). Members of Universidad Politécnica de Madrid (AA, JP and SZ) have received the funding from the SYNTHIA project. SYNTHIA (Synthetic Data Generation framework for integrated validation of use cases and AI healthcare applications) is supported by the Innovative Health Initiative Joint Undertaking (IHI JU) under grant agreement No 101172872. Funded by the European Union, the private members, and those contributing partners of the IHI JU. In addition, *Programa Propio UPM* funded the stay of AA at Saarland University. Moreover, AJ has received funding from the *"UNREAL: a Unified Reasoning Layer for Trustworthy ML"* project (EP/Y023838/1) selected by the ERC and funded by UKRI EPSR. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the aforementioned funding agencies. Neither of the aforementioned parties can be held responsible for them.

Bibliography

[1] Jeffrey Adams, Niels Hansen, and Kun Zhang. Identification of Partially Observed Linear Causal Models: Graphical Conditions for the Non-Gaussian and Heterogeneous Cases. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 22822–22833, 2021. (Cited in page 1.)

[2] Elizabeth S. Allman, Catherine Matias, and John A. Rhodes. Identifiability Of Parameters In Latent Structure Models With Many Observed Variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009. ISSN 00905364, 21688966. (Cited in pages 2, 50, 51, and 52.)

[3] Donald WK Andrews. Examples of l2-complete and boundedly-complete distributions. 2011 (Cited in pages 10 and 27.)

[4] Joshua D Angrist and Jörn-Steffen Pischke. *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2009 (Cited in pages 2 and 51.)

[5] Alexander Balke and Judea Pearl. Probabilistic Evaluation of Counterfactual Queries. *Probabilistic and Causal Inference*, 1994. (Cited in page 7.)

[6] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *Journal of machine learning research*, 20(28):1–6, 2019 (Cited in page 45.)

[7] Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A. Mastakouri, and Dominik Janzing. DoWhy-GCM: An extension of DoWhy for causal inference in graphical causal models. *ArXiv preprint*, abs/2206.06821, 2022. (Cited in page 43.)

[8] Marine Carrasco, Jean-Pierre Florens, and Eric Renault. Linear inverse problems in structural econometrics estimation based on spectral decomposition and regularization. *Handbook of econometrics*, 6:5633–5751, 2007 (Cited in page 27.)

[9] Patrick Chao, Patrick Blöbaum, and Shiva Prasad Kasiviswanathan. Interventional and counterfactual inference with diffusion models. *ArXiv preprint*, abs/2302.00860, 2023. (Cited in pages 1, 2, 9, 38, and 39.)

[10] Juan D. Correa and Elias Bareinboim. A Calculus for Stochastic Interventions: Causal Effect Identification and Surrogate Experiments. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10093–10100. AAAI Press, 2020. (Cited in page 10.)

[11] Yifan Cui, Hongming Pu, Xu Shi, Wang Miao, and Eric Tchetgen Tchetgen. Semiparametric proximal causal inference. *Journal of the American Statistical Association*, 119(546):1348–1359, 2024 (Cited in page 51.)

[12] Alexander D'Amour. On Multi-Cause Approaches to Causal Inference with Unobserved Counfounding: Two Cautionary Failure Cases and A Promising Alternative. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 3478–3486. PMLR, 2019. (Cited in page 51.)

[13] Daniel de Vassimon Manela, Laura Battaglia, and Robin J. Evans. Marginal Causal Flows for Validation and Inference. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. (Cited in page 10.)

[14] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural Spline Flows. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7509–7520, 2019. (Cited in page 43.)

[15] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4):958–968, 2024 (Cited in page 1.)

[16] Denis Fougère and Nicolas Jacquemet. Policy evaluation using causal inference methods. In *Handbook of Research Methods and Applications in Empirical Microeconomics*, pages 294–324. Edward Elgar Publishing, 2021 (Cited in page 10.)

[17] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001 (Cited in pages 9 and 45.)

[18] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked Autoencoder for Distribution Estimation. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 881–889. JMLR.org, 2015. (Cited in page 46.)

[19] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. *Explainable and interpretable models in computer vision and machine learning*, pages 39–80, 2018 (Cited in page 53.)

[20] Sander Greenland. Basic methods for sensitivity analysis of biases. *International journal of epidemiology*, 25(6):1107–1116, 1996 (Cited in page 1.)

[21] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A Kernel Method for the Two-Sample-Problem. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 513–520. MIT Press, 2006. (Cited in pages 8 and 44.)

[22] Jason S. Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep IV: A Flexible Approach for Counterfactual Prediction. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1414–1423. PMLR, 2017. (Cited in pages 10 and 51.)

[23] Heike Hofmann, Karen Kafadar, and Hadley Wickham. Letter-value plots: Boxplots for large data. Technical report, had.co.nz, 2011 (Cited in page 8.)

[24] Patrik O. Hoyer, Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 689–696. Curran Associates, Inc., 2008. (Cited in pages 2 and 9.)

[25] Yimin Huang and Marco Valtorta. Pearl's calculus of intervention is complete. *arXiv preprint arXiv:1206.6831*, 2012 (Cited in page 30.)

[26] Amin Jaber, Adèle H. Ribeiro, Jiji Zhang, and Elias Bareinboim. Causal Identification under Markov equivalence: Calculus, Algorithm, and Completeness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. (Cited in page 53.)

[27] Adrián Javaloy, Pablo Sánchez-Martín, and Isabel Valera. Causal normalizing flows: from theory to practice. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. (Cited in pages 1, 2, 3, 4, 5, 6, 7, 9, 10, 26, 30, 31, 44, 46, 47, 48, and 53.)

[28] Nathan Kallus, Xiaojie Mao, and Madeleine Udell. Causal Inference with Noisy and Missing Covariates via Matrix Factorization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6921–6932, 2018. (Cited in pages 2, 51, and 52.)

[29] Nathan Kallus, Xiaojie Mao, and Angela Zhou. Interval Estimation of Individual-Level Causal Effects Under Unobserved Confounding. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 2281–2290. PMLR, 2019. (Cited in pages 51 and 52.)

[30] David Kaltenpoth and Jilles Vreeken. Nonlinear Causal Discovery with Latent Confounders. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15639–15654. PMLR, 2023. (Cited in page 38.)

[31] Ilyes Khemakhem, Ricardo Pio Monti, Robert Leech, and Aapo Hyvärinen. Causal Autoregressive Flows. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 3520–3528. PMLR, 2021. (Cited in pages 1 and 2.)

[32] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. (Cited in page 42.)

[33] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. (Cited in pages 1 and 4.)

[34] Herbert Knothe. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4:39–52, 1957 (Cited in pages 3 and 37.)

[35] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. (Cited in page 2.)

[36] Andrey Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, 2(4):102–108, 1956 (Cited in page 4.)

[37] Benjamin Kompa, David R. Bellamy, Thomas Kolokotrones, James M. Robins, and Andrew Beam. Deep Learning Methods for Proximal Inference via Maximum Moment Restriction. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors,

Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. (Cited in pages 2 and 51.)

[38] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951 (Cited in page 4.)

[39] Manabu Kuroki and Judea Pearl. Measurement bias and effect restoration in causal inference. *Biometrika*, 101(2):423–437, 2014 (Cited in pages 2, 6, 50, 51, and 52.)

[40] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4066–4076, 2017. (Cited in pages 9, 10, 44, and 45.)

[41] Christopher P Long and Maciek R Antoniewicz. Metabolic flux analysis of Escherichia coli knockouts: lessons from the Keio collection and future outlook. *Current opinion in biotechnology*, 28:127–133, 2014 (Cited in page 39.)

[42] Christos Louizos, Uri Shalit, Joris M. Mooij, David A. Sontag, Richard S. Zemel, and Max Welling. Causal Effect Inference with Deep Latent-Variable Models. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6446–6456, 2017. (Cited in pages 51 and 52.)

[43] Ruiyan Luo and Hongyu Zhao. Bayesian hierarchical modeling for signaling pathway inference from single cell interventional data. *The annals of applied statistics*, 5:725–745, 2011. doi: 10.1214/10-AOAS425 (Cited in page 38.)

[44] Afsaneh Mastouri, Yuchen Zhu, Limor Gultchin, Anna Korba, Ricardo Silva, Matt J. Kusner, Arthur Gretton, and Krikamol Muandet. Proximal Causal Learning with Kernels: Two-Stage Estimation and Moment Restriction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7512–7523. PMLR, 2021. (Cited in pages 2 and 51.)

[45] Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105(4):987–993, 2018 (Cited in pages 1, 2, 6, 10, 27, 29, 51, and 52.)

[46] Wang Miao, Wenjie Hu, Elizabeth L Ogburn, and Xiao-Hua Zhou. Identifying effects of multiple treatments in the presence of unmeasured confounding. *Journal of the American Statistical Association*, 118(543):1953–1967, 2023 (Cited in pages 1, 2, 27, 51, and 52.)

[47] RJ Murnane. *Methods matter: Improving causal inference in educational and social science research*. Oxford University Press, 2010 (Cited in page 10.)

[48] Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual Identifiability of Bijective Causal Models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 25733–25754. PMLR, 2023. (Cited in pages 2, 27, and 53.)

[49] Chris J Oates, Jessica Kasza, Julie A Simpson, and Andrew B Forbes. Repair of partly misspecified causal diagrams. *Epidemiology*, 28(4):548–552, 2017 (Cited in page 10.)

[50] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *J. Mach. Learn. Res.*, 22:57:1–57:64, 2021. (Cited in pages 3, 5, 30, and 38.)

[51] Álvaro Parafita and Jordi Vitrià. Estimand-agnostic causal query estimation with deep causal graphs. *IEEE Access*, 10:71370–71386, 2022 (Cited in page 2.)

[52] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep Structural Causal Models for Tractable Counterfactual Inference. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing*

Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. (Cited in page 2.)

[53] J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Penguin Books Limited, 2018. ISBN 9780241242643. (Cited in page 33.)

[54] Judea Pearl. Causal Diagrams for Empirical Research. *Biometrika*, 82(4):669–688, 1995. ISSN 00063444, 14643510. (Cited in page 3.)

[55] Judea Pearl. *Causality*. Cambridge university press, 2009 (Cited in pages 2, 3, 4, 34, and 51.)

[56] Judea Pearl. The Do-Calculus Revisited. In Nando de Freitas and Kevin P. Murphy, editors, *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pages 3–11. AUAI Press, 2012. (Cited in page 3.)

[57] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016 (Cited in page 48.)

[58] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017 (Cited in pages 3, 29, and 34.)

[59] Md. Musfiqur Rahman and Murat Kocaoglu. Modular Learning of Deep Causal Generative Models for High-dimensional Causal Inference. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. (Cited in pages 2 and 53.)

[60] Rajesh Ranganath and Adler Perotte. Multiple causal inference with latent confounding. *ArXiv preprint*, abs/1805.08273, 2018. (Cited in pages 51 and 52.)

[61] Severi Rissanen and Pekka Marttinen. A Critical Look at the Consistency of Causal Estimation with Deep Latent Variable Models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4207–4217, 2021. (Cited in page 51.)

[62] Murray Rosenblatt. Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952. ISSN 00034851. (Cited in pages 3 and 37.)

[63] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. *Science*, 308 (5721):523–529, 2005. doi: 10.1126/science.1105809. (Cited in pages 9, 38, and 39.)

[64] Pedro Sanchez, Jeremy P Voisey, Tian Xia, Hannah I Watson, Alison Q O'Neil, and Sotirios A Tsaftaris. Causal machine learning for healthcare and precision medicine. *Royal Society Open Science*, 9(8):220638, 2022 (Cited in page 10.)

[65] Pablo Sánchez-Martín, Miriam Rateike, and Isabel Valera. VACA: Designing Variational Graph Autoencoders for Causal Queries. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8159–8168. AAAI Press, 2022. (Cited in page 2.)

[66] Juliane Schäfer and Korbinian Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1), 2005 (Cited in pages 2, 9, and 39.)

[67] Marco Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03 (Cited in pages 38 and 40.)

[68] Xu Shi, Wang Miao, Jennifer C Nelson, and Eric J Tchetgen Tchetgen. Multiply robust causal inference with double-negative control adjustment for categorical unmeasured confounding. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(2):521–540, 2020 (Cited in page 51.)

[69] Ilya Shpitser and Judea Pearl. Identification of joint interventional distributions in recursive semi-Markovian causal models. In *AAAI*, pages 1219–1226, 2006 (Cited in pages 29 and 30.)

[70] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001 (Cited in page 29.)

[71] Eric J Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An introduction to proximal causal learning. *ArXiv preprint*, abs/2009.10982, 2020. (Cited in page 51.)

[72] Santtu Tikka and Juha Karvanen. Identifying causal effects with the R package causaleffect. *Journal of Statistical Software*, 76:1–30, 2017 (Cited in page 30.)

[73] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. (Cited in pages 5, 37, and 46.)

[74] Hal R Varian. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, 113(27):7310–7315, 2016 (Cited in page 1.)

[75] Yixin Wang and David M Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019 (Cited in pages 9, 40, 51, and 52.)

[76] Yixin Wang and David M. Blei. A Proxy Variable View of Shared Confounding. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR, 2021. (Cited in pages 1, 2, 6, 10, 27, 29, 40, 51, and 52.)

[77] Yixin Wang, David M. Blei, and John P. Cunningham. Posterior Collapse and Latent Variable Non-identifiability. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5443–5455, 2021. (Cited in pages 5 and 52.)

[78] Linda F Wightman. LSAC National Longitudinal Bar Passage Study. LSAC Research Report Series. 1998 (Cited in page 9.)

[79] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning Likelihoods with Conditional Normalizing Flows. *ArXiv preprint*, abs/1912.00042, 2019. (Cited in pages 1 and 4.)

[80] Quanhan Xi and Benjamin Bloem-Reddy. Indeterminacy in Generative Models: Characterization and Strong Identifiability. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 6912–6939. PMLR, 2023. (Cited in pages 2, 5, 26, and 53.)

[81] Kevin Xia, Kai-Zhan Lee, Yoshua Bengio, and Elias Bareinboim. The Causal-Neural Connection: Expressiveness, Learnability, and Inference. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 10823–10836, 2021. (Cited in pages 2 and 52.)

[82] Kevin Muyuan Xia, Yushu Pan, and Elias Bareinboim. Neural Causal Models for Counterfactual Identification and Estimation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. (Cited in pages 2, 10, and 52.)

[83] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Structured causal disentanglement in variational autoencoder. *ArXiv preprint*, abs/2004.08697, 2020. (Cited in page 2.)

[84] Matej Zečević, Devendra Singh Dhami, Petar Velivcković, and Kristian Kersting. Relating graph neural networks to structural causal models. *ArXiv preprint*, abs/2109.04173, 2021. (Cited in page 2.)

[85] Siyuan Zhao and Neil Heffernan. Estimating Individual Treatment Effect from Educational Studies with Residual Counterfactual Networks. *International Educational Data Mining Society*, 2017 (Cited in page 1.)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction assert that DeCaFlow provides correct estimates for every do-calculus and proximal identifiable causal query, including counterfactuals, under hidden confounding and outperforms prior models. These points are rigorously proven in the theory sections (§4) and demonstrated by comprehensive empirical results (§5).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated "Limitations" paragraph at the end of §6 explicitly states that DeCaFlow depends on sufficiently informative variables—an untestable assumption—and on a C^1-diffeomorphic confounded SCM, thereby clarifying when the method may fail and why.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Every theorem, proposition and corollary in §4 is stated with explicit numbered assumptions and the correspoding formal proofs are given in §A and cross referenced. As an example, Prop. A.2 is the formal version of Prop. 4.1, which includes a list of independence, completeness and regularity conditions and then supplies a step-by-step proof.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The main paper explicitly refers to "all experiments details" to §3.2 and §C, §B and §D, where it specified the data-generation pipelines, the causal graphs, the architecture, the ELBO training objective with its regularization process, the do-operator, the metrics and the evaluation protocol In addition, the authors commit to releasing the full codebase, together with the hyper-parameter search and seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general.

releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: he datasets employed are publicly available or fully described, a user-friendly implementation of DeCaFlow is available in the link of the introduction of this paper (`github.com/aalmodovares/DeCaFlow`), as well as examples of use, identifiability check algorithms and visualizations. In addition, we offer the whole training, hp tuning and validation pipeline, for reproducibility, under request to correspondence authors.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: §B is entirely devoted to experimental details, describing for each dataset the generation pipeline, train-test splits, the number of runs, the interventions percentiles and exact metrics. In addition, §C supplements with implementation hyper-parameters, warm-up schedule, posterior factorization and masking strategy.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Every key figure and table includes statistical uncertainty. Fig. 7 plots boxenplots (with all percentiles), and tables provide mean and standard deviation of all metrics across all seeds and all evaluated causal effects. Tables also include significantly better results related with statistical tests and intervals included in the captions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details about the execution resources and times are provided in §C,

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: All experiments rely on publicly available or semi-synthetic benchmark data (Sachs protein signalling, Ecoli70 gene network, and the anonymised LSAC law-school dataset) and introduce no new personal data collection, human-subject interaction, or high-risk model release; the work explicitly addresses fairness (§5) and reports moderate compute usage, thereby avoiding the privacy, discrimination, security, or environmental concerns enumerated in the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We explain how DeCaFlow can enable better decision-making in domains such as healthcare and education while cautioning that causal assumptions must be validated, especially in sensitive applications, and state that the method introduces no additional ethical risks beyond those already known for causal-inference models.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The release comprises only the DeCaFlow training and evaluation code plus lightweight models trained on publicly available or semi-synthetic benchmarks; because no large pretrained generative models or scraped datasets with dual-use potential are distributed, special safeguards are unnecessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The licenses and all the copyright information will be included in every asset in the code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new, fully self-contained DeCaFlow repository released under the General Public License. No new datasets or personal data are created, so consent and privacy disclosures are unnecessary.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The study does not involve any crowdsourcing or prospective research with human participants; it relies exclusively on pre-existing, publicly available or synthetic datasets (Sachs, Ecoli70, LSAC), so participant instructions, screenshots, and compensation details are not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The work uses only pre-existing benchmark datasets—Sachs protein-signalling, a semi-synthetic Ecoli70 gene network, and the publicly released LSAC law-school dataset—without recruiting new participants or collecting personal data, so human-subjects review and IRB disclosure are not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The authors have not used LLMs for important tasks of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

Table of Contents

A Causal identifiability

A.1 Model identifiability

We briefly discuss the identifiability (in the sense of Xi and Bloem-Reddy [80]) of those variables that are indirectly confounded by $\mathbf{z}$ or not confounded at all, i.e., of those variables that are not children of any hidden confounder. As we discuss now, we can reduce our SCM (Def. 1) to a conditional one that only models these aforementioned variables, recovering the identifiability guarantees from Javaloy et al. [27]. To prove model identifiability, we resort to what we call the induced conditional SCM, which intuitively represents the original SCM where we restrict our view to a subset of variables, and assume the rest of the variables are given.

Definition 3 (Induced conditional SCM)**.** Given a SCM $\mathcal{M} = (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$, and a subset of observed variables $\mathbf{x}' \subset \mathbf{x}$, we define the *induced conditional SCM of $\mathcal{M}$ given $\mathbf{x}'$*, denoted by $\mathcal{M}_{|\mathbf{x}'}$, to the SCM result of having observed $\mathbf{x}'$, and where causal generators and exogenous variables are restricted to only those associated with the unconditioned variables, i.e., $\mathbf{x} \setminus \mathbf{x}'$.

(a) Confounded SCM. (b) Conditional unconfounded SCM.

Figure A.1: Example of: **(a)** a confounded SCM $\mathcal{M}$; and **(b)** its induced conditional counterpart, $\mathcal{M}_{|\mathbf{x}'}$ where the children of the hidden confounder are observed and fixed, $\mathbf{x}' = \mathrm{ch}(\mathbf{z}) = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_7\}$. Note that $\mathcal{M}_{|\mathbf{x}'}$ does not exhibit hidden confounding.

We provide a visual depiction of this idea in Fig. A.1. Using this definition, we can observe that, if we were to condition on the children of the hidden confounder, we would be left with a (conditional) *unconfounded SCM*, as the influence of the hidden confounder has been completely blocked by conditioning on its children. Now, if we have two models that perfectly match their marginal distributions, this means that they perfectly match their induced conditional SCM, no matter which value we observed for $\mathrm{ch}(\mathbf{z})$, and we can thus leverage existing results from Javaloy et al. [27] for unconfounded SCMs. More specifically:

Corollary A.1. *Assume that we have two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$ that are Markov-equivalent—i.e., they induce the same causal graph—and which coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. Then, both SCMs, restricted to every variable other than $\mathrm{ch}(\mathbf{z})$, are equal up to an element-wise transformation of the exogenous distributions.*

Proof. The proof follows almost directly from [27, Theorem 1]. First, note that the two induced conditional SCMs are no longer influenced by $\mathbf{z}$ once that we have observed a specific realization of $\mathrm{ch}(\mathbf{z})$, so that we can drop $\mathbf{z}$ from their structure, i.e., we can rewrite them instead as unconfounded SCMs, $\mathcal{M}_{|\mathrm{ch}(\mathbf{z})} = (\mathbf{f}_{|\mathrm{ch}(\mathbf{z})}, P_{\mathbf{u}|\mathrm{ch}(\mathbf{z})})$ and $\tilde{\mathcal{M}}_{|\mathrm{ch}(\mathbf{z})} = (\tilde{\mathbf{f}}_{|\mathrm{ch}(\mathbf{z})}, P_{\tilde{\mathbf{u}}|\mathrm{ch}(\mathbf{z})})$. To ease notation, let us call $\mathbf{x}^{\mathsf{c}} \coloneqq \mathbf{x} \setminus \mathrm{ch}(\mathbf{z})$ the variables that are not children of $\mathbf{z}$.

Next, note that for almost every realization of $\mathrm{ch}(\mathbf{z})$, we have that $p(\mathbf{x}^{\mathsf{c}} | \, \mathrm{ch}(\mathbf{z})) \overset{a.e.}{=} \tilde{p}(\mathbf{x}^{\mathsf{c}} | \, \mathrm{ch}(\mathbf{z}))$ since $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$ by assumption and $p(\mathbf{x}) = p(\mathbf{x}^{\mathsf{c}} | \, \mathrm{ch}(\mathbf{z}))p(\mathrm{ch}(\mathbf{z}))$. As a result, for each realization of $\mathrm{ch}(\mathbf{z})$ we can apply Theorem 1 of Javaloy et al. [27], which yields that the two induced conditional SCMs are equal up to an element-wise transformation of the exogenous distribution.

Finally, since the causal generators and exogenous distributions of the induced SCMs are, for almost every $\mathrm{ch}(\mathbf{z})$, identical to their counterparts in the original SCMs (as we have just discarded those components associated with $\mathrm{ch}(\mathbf{z})$), we get that, those elements in both SCMs associated with $\mathbf{x}^{\mathbf{c}}$, are identical up to said (possibly $\mathrm{ch}(\mathbf{z})$-dependent) component-wise transformation. □

A.2 Query identifiability

We now prove the identifiability of the causal queries considered in the main text. To this end, one key property that we will use in the following is that of completeness (as, e.g., in the work of Wang and Blei [76]). Intuitively, we say that a random variable $\mathbf{z}$ is complete given another random variable $\mathbf{n}$ if "any infinitesimal change in $\mathbf{z}$ is accompanied by variability in $\mathbf{n}$" [46], yielding enough information to recover the posterior distribution of $\mathbf{z}$. This concept is similar in spirit to that of variability in the case of discrete random variables [48]. In practice, completeness is more likely to be achieved the more proxies we measure [3].

Definition 4 (Completeness)**.** We say that a random variable $\mathbf{z}$ is complete given $\mathbf{n}$ for almost all $\mathbf{c}$ if, for any square-integrable function $g(\cdot)$ and almost all $\mathbf{c}$, $\int g(\mathbf{z}, \mathbf{c}) p(\mathbf{z} | \mathbf{c}, \mathbf{n}) \, \mathrm{d}\mathbf{z} = 0$ for almost all $\mathbf{n}$, if and only if $g(\mathbf{z}, \mathbf{c}) = 0$ for almost all $\mathbf{z}$.

The following proposition (informally simplified in Prop. 4.1) is a generalization of the results previously presented by Miao et al. [45] and Wang and Blei [76], where we include an additional covariate $\mathbf{c}$ to the causal query, and make no implicit assumptions on the causal graph allowing, e.g., for the treatment and outcome variables to hame some observed parents in common. However, note that $\mathbf{c}$ cannot be a collider (e.g., forming a subgraph of the form $\mathbf{n} \to \mathbf{c} \leftarrow \mathrm{y}$). Otherwise, conditioning on $\mathbf{c}$ would make independent variables dependent (in the example, y and $\mathbf{n}$), and the causal effect of t on y would not be identifiable.

Proposition A.2 (Query identifiability)**.** *Given two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$, assume that they are Markov-equivalent—i.e., they induce the same causal graph—and which coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. Then, they compute the same causal query, $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c}) = \tilde{p}(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$, where $\mathrm{y}, \mathrm{t}, \mathbf{c} \subset \mathbf{x}$, if there exists two proxies $\mathbf{w}, \mathbf{n} \subset \mathbf{x}$ and $\mathbf{b} \subset \mathbf{x}$, none of them overlapping nor containing variables from the previous subsets, s.t.:*

i) $\mathbf{w}$ *is conditionally independent of* $(\mathrm{t}, \mathbf{n})$ *given* $\mathbf{b}$*,* $\mathbf{z}$ *and* $\mathbf{c}$*. That is,* $\mathbf{w} \perp\!\!\!\perp (\mathrm{t}, \mathbf{n}) | \, \mathbf{b}, \mathbf{z}, \mathbf{c}$.

ii) $\mathbf{n}$ *is conditionally independent of* y *given* t*,* $\mathbf{b}$*,* $\mathbf{z}$ *and* $\mathbf{c}$*. That is,* $\mathrm{y} \perp\!\!\!\perp \mathbf{n} | \, \mathrm{t}, \mathbf{b}, \mathbf{z}, \mathbf{c}$.

iii) $(\mathbf{b}, \mathbf{z})$ *forms a valid adjustment set for the query* $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$*. That is, given* $\mathbf{c}$*, they are independent of* t *after severing any incoming edges to it,* $\mathrm{t} \perp\!\!\!\perp_{\mathcal{G}_{\overline{\mathrm{t}}}} (\mathbf{b}, \mathbf{z}) | \, \mathbf{c}$*, and they block every backdoor path from* t *to* y.

iv) $\mathbf{z}$ *is complete given* $\mathbf{n}$ *for almost all* t*,* $\mathbf{b}$*, and* $\mathbf{c}$*,*

v) $\tilde{\mathbf{z}}$ *is complete given* $\mathbf{w}$ *for almost all* $\mathbf{b}$ *and* $\mathbf{c}$*,*

and the following regularity conditions also hold:

vi) $\iint \tilde{p}(\tilde{\mathbf{z}} | \, \mathbf{w}, \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{w} | \, \tilde{\mathbf{z}}, \mathbf{b}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} \, \mathrm{d}\mathbf{w} < \infty$ *for all* $\mathbf{b}$*,* $\mathbf{c}$*, and*

vii) $\int \tilde{p}(\mathrm{y} | \, \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c})^2 \tilde{p}(\tilde{\mathbf{z}} | \, \mathbf{b}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} < \infty$ *for all* t*,* $\mathbf{b}$*, and* $\mathbf{c}$*.*

Proof. First, note that the first three independence assumptions hold for both models, $\mathcal{M}$ and $\tilde{\mathcal{M}}$, as they induce the same causal graph. Following the same arguments as Miao et al. [45, Proposition 1], we have that assumptions **v)**, **vi)**, and **vii)** guarantee the existence of a function $\tilde{h}$ such that it solves the integral equation over $\tilde{\mathcal{M}}$,

$$\tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) = \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) \tilde{p}(\mathbf{w} \mid \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \,, \tag{7}$$

since assumption **vi)** ensures that the conditional expectation operator is compact [8], assumption **v)** that all square-integrable functions are in the image of the operator (i.e., the operator is surjective), and assumption **vii)** that $\tilde{p}(\mathrm{y} | \, \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c})$ is indeed part of the image.

We can show that $\tilde{h}$ also solves a similar integral equation, this time over the other SCM, $\mathcal{M}$, as follows:

$$
\begin{aligned}
p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) &= \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) && [\textit{equal marginals}] && (8)\\
&= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} && [\textit{augment with } \tilde{\mathbf{z}}] && (9)\\
&= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} && [\textit{assumption } \textbf{\textit{ii)}}] && (10)\\
&= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) \tilde{p}(\mathbf{w} \mid \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} \, \mathrm{d}\mathbf{w} && [\textit{plug Eq. 7}] && (11)\\
&= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) \tilde{p}(\mathbf{w} \mid \mathbf{b}, \tilde{\mathbf{z}}, \mathrm{t}, \mathbf{n}, \mathbf{c}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\tilde{\mathbf{z}} \, \mathrm{d}\mathbf{w} && [\textit{assumption } \textbf{\textit{i)}}] && (12)\\
&= \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \, . && [\textit{equal marginals}] && (13)
\end{aligned}
$$

Note that Eq. 13 is a Fredholm equation of the first kind that is implicitly solved by modeling the observational data. Similarly, we can relate the expression for the interventional distribution of both models:

$$
\begin{aligned}
\tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) &= \int \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\mathbf{b}, \tilde{\mathbf{z}} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\tilde{\mathbf{z}} && [\textit{augment and ass. } \textbf{\textit{iii)}}] && (14)\\
&= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\mathbf{b}, \tilde{\mathbf{z}} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\tilde{\mathbf{z}} && [\textit{backdoor criterion}] && (15)\\
&= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) \tilde{p}(\mathbf{w} \mid \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c}) \tilde{p}(\mathbf{b}, \tilde{\mathbf{z}} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{w} \, \mathrm{d}\tilde{\mathbf{z}} && [\textit{plug Eq. 7}] && (16)\\
&= \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{b}, \mathbf{w} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{w} && [\textit{equal marginals}] && (17)\\
&= p(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) \, , && && (18)
\end{aligned}
$$

where the last equality is a consequence of Eq. 13 as we will show now. More specifically, we have that

$$
\begin{aligned}
p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) &= \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{w} && [\textit{Eq. 13}] && (19)\\
&= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathbf{b}, \mathbf{z}, \mathrm{t}, \mathbf{n}, \mathbf{c}) p(\mathbf{z} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\mathbf{z} \, , && [\textit{augment with } \mathbf{z}] && (20)\\
&= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathbf{b}, \mathbf{z}, \mathbf{c}) p(\mathbf{z} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\mathbf{z} \, . && [\textit{assumption } \textbf{\textit{i)}}] && (21)
\end{aligned}
$$

Similarly, we have that

$$
\begin{aligned}
p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) &= \int p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{z}, \mathbf{c}) p(\mathbf{z} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{z} && [\textit{augment with } \mathbf{z}] && (22)\\
&= \int p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{z}, \mathbf{c}) p(\mathbf{z} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{z} \, . && [\textit{assumption } \textbf{\textit{ii)}}] && (23)
\end{aligned}
$$

Now, equating both expressions we have that

$$
0 = \iint \left\{ p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{z}, \mathbf{c}) - \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathbf{b}, \mathbf{z}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \right\} p(\mathbf{z} \mid \mathrm{t}, \mathbf{b}, \mathbf{n}, \mathbf{c}) \, \mathrm{d}\mathbf{z} \, , \tag{24}
$$

which, due to assumption **iv)**, implies that

$$p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{z}, \mathbf{c}) \overset{\text{a.e.}}{=} \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathbf{b}, \mathbf{z}, \mathbf{c}) \, \mathrm{d}\mathbf{w} \,. \tag{25}$$

Finally, putting all together we see that we can write the interventional distribution of the original model using $\tilde{h}$,

$$p(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) = \iint p(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{b}, \mathbf{z}, \mathbf{c}) p(\mathbf{b}, \mathbf{z} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{z} \qquad [\textit{augment and assumption iii)}] \tag{26}$$

$$= \iint p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{z}, \mathbf{c}) p(\mathbf{b}, \mathbf{z} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{z} \qquad [\textit{backdoor criterion}] \tag{27}$$

$$= \iint \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{w} \mid \mathbf{b}, \mathbf{z}, \mathbf{c}) p(\mathbf{b}, \mathbf{z} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{z} \, \mathrm{d}\mathbf{w} \qquad [\textit{Eq. 25}] \tag{28}$$

$$= \int \tilde{h}(\mathrm{y}, \mathrm{t}, \mathbf{b}, \mathbf{w}, \mathbf{c}) p(\mathbf{b}, \mathbf{w} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \, \mathrm{d}\mathbf{w} \,, \qquad [\textit{equal marginals}] \tag{29}$$

which justifies the last equality in Eq. 18. □

In Prop. A.2, assumptions **i)-iii)** regard the conditional independence of different variables in the causal graph, which can be directly verified given a faithful causal graph. Assumptions **iv)** and **v)** regard the information that $\mathbf{w}$ and $\mathbf{n}$ contain of the hidden confounders which, intuitively, means that the posterior of the hidden confounder varies enough as we change the values of $\mathbf{w}$ and $\mathbf{n}$, in order to properly perform inference on it. This assumption is harder to verify but, as we show in §B.1, the more proxy variables we have, the better the estimation of the hidden confounder's effect and the more accurate the causal query estimation is. Finally, assumptions **vi)** and **vii)** are standard regularity conditions [45, 76] that are (almost surely) fulfilled in practice, as long as the random variables are well behaved, e.g., having finite moments. Such conditions are typically satisfied by most continuous and discrete distributions used in probabilistic modeling, including Gaussian, exponential family, and bounded-support distributions, making them mild and non-restrictive assumptions.

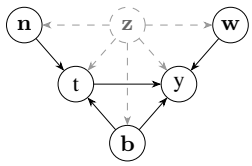


Figure A.2: Example for which Prop. A.2 applies, and where $\mathbf{b} \neq \emptyset$.

Using a causal graph similar to the one presented by Miao et al. [45], we now provide some intuition on the semantics of each random variable in Prop. A.2. More specifically, consider the causal graph that we depict in Fig. A.2, and say that we want to check if the causal query $p(\mathrm{y}|\,\mathrm{do}(\mathrm{t}))$ is identifiable (note that this the same query as in Prop. A.2 but with $\mathbf{c} = \emptyset$). As it is common in the causal inference literature [58, 70], t and y represent the treatment and outcome random variables. More specific to Prop. A.2 are $\mathbf{w}$ and $\mathbf{n}$. Here, $\mathbf{w}$ is a proxy variable whose role is that of distinguishing the information from $\mathbf{z}$ and other variables, to reconstruct the information of $\mathbf{z}$ and block the backdoor path that $\mathbf{z}$ would usually block. Similarly, the variable $\mathbf{n}$ is another proxy variable which, in this case, serves the purpose of verifying that the substitute formed with $\mathbf{w}$ is indeed a good one. Finally, the variable $\mathbf{b}$ serves the purpose of blocking all the remaining backdoor paths that $\mathbf{z}$ may not block, so that we can apply the backdoor criterion.

Moreover, note that for all interventional queries we let $\mathbf{c}$ be the empty set, similar to the results proved by Miao et al. [45] and Wang and Blei [76]. We will consider cases when $\mathbf{c}$ is not empty later in §A.3 to prove counterfactual identifiability. Note also that Prop. A.2 reduces to previous results when $\mathbf{c} = \mathbf{b} = \emptyset$.

We now turn our attention towards proving Cor. 4.2, i.e., towards broadening the concept of query identifiability by introducing Prop. A.2 as a base case of do-calculus. To this end, we introduce the concept of a *hedge* which will be use later, but we still strongly recommend reading the work by Shpitser and Pearl [69].

Definition 5 (Hedge, [69, Def. 6])**.** Let $\mathrm{y}, \mathrm{t} \subset \mathbf{x}$ be disjoint sets of variables in $\mathcal{G}$. Let F, F' be $\boldsymbol{r}$-rooted C-forests (see [69, Def. 5]) such that $F \cap \mathrm{t} \neq \emptyset$, $F' \cap \mathrm{t} = \emptyset$, $F' \subset F$, and $\boldsymbol{r}$ is a subset of the ancestors of y after severing the incoming edges of t. Then F and F' form a hedge for $p(\mathrm{y}|\operatorname{do}(\mathrm{t}))$ in $\mathcal{G}$.

Corollary 4.2. *An interventional query is identifiable if, using do-calculus, it can be reduced to a combination of observational queries and identifiable interventional queries in the sense of Prop. 4.1.*

Proof. With the additional notion of proxy-identifiability provided by Prop. A.2 (informally presented in Prop. 4.1), the result is just a consequence of applying the identifiability algorithm provided by Shpitser and Pearl [69]. See also [25, 72] for other references.

Since the do-calculus rules are complete in the classical sense of identifiability, a query is not identifiable if the aforementioned algorithm yields a `FAIL` status (i.e., it executes line 5 of Figure 3 in [69]). If that is the case, then it means that, at the specific recursive call for which the algorithm failed, the local graph $\mathcal{G}$ contains a hedge and the interventional query $p(\mathrm{y}|\operatorname{do}(\mathrm{t}))$ is not identifiable in the classical sense.

Crucially, this hedge (F, F') expresses the inability of identifying an interventional query of the form $p(\boldsymbol{r}|\operatorname{do}(\mathrm{t}'))$ where the root $\boldsymbol{r}$ is a subset of ancestors of $\mathrm{y}' \subseteq \mathrm{y}$ and $\mathrm{t}' \subseteq \mathrm{t}$. Then, this local query can still be proxy-identifiable if Prop. A.2 can be applied, and thus we can continue running the identification algorithm.

The stated result is then a consequence of successfully applying the logic above each time we find a `FAIL` status, yielding a final `FAIL` status otherwise. □

To be even more explicit regarding the identifiability of the queries proven in corollary above, let us call $\mathcal{M}$ the original SCM as usual, and $\tilde{\mathcal{M}}$ another SCM inducing the same causal graph as $\mathcal{M}$ and which matches the observational marginal distribution of $\mathcal{M}$, i.e., $p(\mathbf{x}) \overset{\text{a.e.}}{=} \tilde{p}(\mathbf{x})$. Then, the output of the identifiability algorithm from the corollary above *for both SCMs* will be two identical expressions `EXP` composed of sum, integrals, and products of observational quantities (i.e., marginals and conditionals of subsets of $\mathbf{x}$) as well as proxy-identifiable queries of the form $p(\mathrm{y}|\operatorname{do}(\mathrm{t}))$ as in Prop. A.2. Therefore,

$$Q(\mathcal{M}) = \texttt{EXP}(\mathcal{M}) = \texttt{EXP}(\tilde{\mathcal{M}}) = Q(\tilde{\mathcal{M}})\,, \tag{30}$$

where the second equality is a consequence of both SCMs having equal observational distributions (and thus any other quantity than can derived exclusively from $p(\mathbf{x})$) and of applying Prop. A.2 for any interventional query that appears in the expression.

Corollary 4.3. *If DeCaFlow induces the same causal graph $\mathcal{G}$ as $\mathcal{M}$ and $p_{\mathcal{M}}(\mathbf{x}) \overset{\text{a.e.}}{=} p_{\boldsymbol{\theta}}(\mathbf{x})$, then DeCaFlow provides correct estimates of any query identifiable in the sense of Cor. 4.2.*

Proof. The proof is a direct consequence of the corollary above and the fact that we can interpret DeCaFlow as a dense parametric family of confounded SCMs inducing the same causal graph as $\mathcal{M}$ (similar to the interpretation of Javaloy et al. [27] as bijective SCMs) by considering the triplet $\mathcal{M}_{\boldsymbol{\theta}} \coloneqq (T_{\boldsymbol{\theta}}^{-1}, P_{\mathbf{u}}, P_{\mathbf{z}})$, where $T_{\boldsymbol{\theta}}^{-1}$ is the inverse of the generative network that transforms $\mathbf{u}$ into $\mathbf{x}$ given $\mathbf{z}$. This family being dense is a consequence of the generative networks forming a family of universal density approximators [27, 50]. □

To be completely exhaustive, in the following we explore the general proposition Prop. A.2 on all scenarios where t and y may or may not be directly caused by the hidden confounder, as we show in the following subsections.

A.2.1 Fully hidden-confounded case

In the case where both variables are children of $\mathbf{z}$, we must see whether we can apply do-calculus with Prop. A.2 as an additional base case, as described in Cor. 4.2.

A.2.2 Hidden-unconfounded case

Assume the case where neither t nor y are children of the hidden confounder, i.e., $\mathrm{y}, \mathrm{t} \notin \mathrm{ch}(\mathbf{z})$. In this case, the proof of Prop. A.2 can be simplified and drop the requirement of finding valid proxy variables.

Corollary A.3. *Given two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$, assume that they are Markov-equivalent—i.e., they induce the same causal graph—and coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. If $\mathrm{y}, \mathrm{t} \notin \mathrm{ch}(\mathbf{z})$, then, $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c}) = \tilde{p}(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$, where $\mathrm{y}, \mathrm{t}, \mathbf{c} \subset \mathbf{x}$.*

Proof. The proof follows directly by applying Prop. A.2 with the minimal subset $\mathbf{b} \subset \mathrm{pa}(\mathrm{t}) \setminus \{\mathbf{c}\}$ that blocks all the backdoor paths, and by noticing that in this case there is no need to use the variables $\mathbf{z}$ and $\tilde{\mathbf{z}}$. That is, we can go from Eq. 14 to Eq. 18 directly by using only $\mathbf{b}$ and the equal-marginals assumption:

$$\tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) = \int \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{b} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \tag{31}$$

$$= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{b} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \tag{32}$$

$$= \int p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{c}) p(\mathbf{b} \mid \mathbf{c}) \, \mathrm{d}\mathbf{b} \tag{33}$$

$$= p(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) \, . \tag{34}$$

□

Even though we can leverage and simplify Prop. A.2 as shown above, it is worth remarking that, for this particular case, the model identifiability results described in §A.1 are stronger, as it provides results on the identifiability of the causal generators and exogenous distributions, and therefore of any causal query derived from them.

A.2.3 Confounded outcome case

For the case where only the outcome variable is a child of the hidden confounder, we can apply a similar reasoning as we did in the previous case, although this time we cannot leverage the stronger results from Javaloy et al. [27]. More specifically:

Corollary A.4. *Given two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$, assume that they are Markov-equivalent—i.e., they induce the same causal graph—and coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. Assume that $\mathrm{t} \notin \mathrm{ch}(\mathbf{z})$. Then, $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c}) = \tilde{p}(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$, where $\mathrm{y}, \mathrm{t}, \mathbf{c} \subset \mathbf{x}$.*

Proof. The proof is identical to that of Cor. A.3. □

Front-door example. While the proof above is trivial given the previous results, it is worth stressing the importance of modeling the hidden confounder as we do in this work with DeCaFlow. As an example, consider the SCM depicted in Fig. A.3, where we have that the outcome is directly confounded by $\mathbf{z}$, while t is not. In this case, DeCaFlow can correctly estimate the causal effects of $\mathbf{b}$ and t on y, i.e., to correctly estimate $p(\mathrm{y} | \, \mathrm{do}(\mathrm{t}))$ and $p(\mathrm{y} | \, \mathrm{do}(\mathbf{b}))$, using $\tilde{\mathbf{z}}$ to model the influence of $\mathbf{b}$ onto y that is not explained through t. Other models that do not model $\mathbf{z}$—e.g., an unaware CNF [27]—would be able to match the observed marginal distribution (as they are universal density approximators) and therefore to estimate $p(\mathrm{y} | \, \mathrm{do}(\mathbf{b}))$ (as it is identifiable through the mediator t using the front-door criterion), yet they would necessarily fail to estimate $p(\mathrm{y} | \, \mathrm{do}(\mathrm{t}))$, since they assume that $\mathrm{y} \perp\!\!\!\perp \mathbf{b} | \, \mathrm{t}$ yet we know that $\mathrm{y} \not\perp\!\!\!\perp \mathbf{b} | \, \mathrm{t}$ in the true model. In other words, an unaware CNF would hold that $p(\mathrm{y} | \, \mathrm{do}(\mathrm{t})) = p(\mathrm{y} | \, \mathrm{t})$ which is clearly false by looking at Fig. A.3.

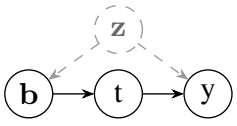


Figure A.3: Example of a front-door causal.

To be even more explicit, in this case we would have a data-generating process that factorizes as

$$\tilde{p}(\mathbf{b}, \mathrm{t}, \mathrm{y}, \tilde{\mathbf{z}}) = \tilde{p}(\tilde{\mathbf{z}})\tilde{p}(\mathbf{b} \mid \tilde{\mathbf{z}})\tilde{p}(\mathrm{t} \mid \mathbf{b})\tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}})\,, \tag{35}$$

and hence the estimated interventional distribution from DeCaFlow matches the true one:

$$\begin{aligned}
p(\mathrm{y} \mid \mathrm{do}(\mathrm{t})) &= \int p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}) p(\mathbf{b})\, \mathrm{d}\mathbf{b} && [\mathbf{b} \textit{ forms a valid adjustment set}] && (36)\\
&= \int \left\{ \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathrm{t}, \mathbf{b})\, \mathrm{d}\tilde{\mathbf{z}} \right\} \tilde{p}(\mathbf{b})\, \mathrm{d}\mathbf{b} && [\textit{Factorization and eq. marginals}] && (37)\\
&= \iint \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}) \tilde{p}(\tilde{\mathbf{z}} \mid \mathbf{b}) \tilde{p}(\mathbf{b})\, \mathrm{d}\mathbf{b}\, \mathrm{d}\tilde{\mathbf{z}} && [\textit{Factorization in Eq. 35}] && (38)\\
&= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}) \tilde{p}(\tilde{\mathbf{z}})\, \mathrm{d}\tilde{\mathbf{z}} && [\textit{marginalize } \mathbf{b}] && (39)\\
&= \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}))\,. && && (40)
\end{aligned}$$

A.2.4 Hidden-confounded treatment case

When only the treatment variable t is a child of $\mathbf{z}$, we can face two different scenarios: **i)** we find a valid adjustment set $\mathbf{b}$ blocking all backdoor paths, in which case we can reason just as in the other partially hidden-confounded case, and **ii)** we cannot, and then rely on do-calculus and the identifiability w.r.t. $\mathbf{b}$. For example, if $\mathbf{b}$ happens to be a parent of y which is directly caused by the treatment variable t and the hidden confounder $\mathbf{z}$ as in Fig. A.4, we cannot find a valid adjustment set for the causal query, but it may still serve us if we can identify the same query with the adjustment set as outcome variable.

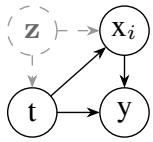


Figure A.4: Case with no valid adjustment set.

Corollary A.5. *Given two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$, assume that they are Markov-equivalent—i.e., they induce the same causal graph—and coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. If $\mathrm{y} \notin \mathrm{ch}(\mathbf{z})$ then, $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c}) = \tilde{p}(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$, where $\mathrm{y}, \mathrm{t}, \mathbf{c} \subset \mathbf{x}$ if there exists $\mathbf{b} \subset \mathbf{x}$ not containing variables from the previous subsets, such that one of the following two conditions are true:*

i) $\mathbf{b}$ *forms a valid adjustment set for the query* $p(\mathrm{y} | \, do(\mathrm{t}), \mathbf{c})$.

ii) $\mathbf{b}$ *blocks all backdoor paths and the query* $p(\mathbf{b} | \, do(\mathrm{t}), \mathbf{c})$ *is identifiable.*

Proof. If condition **i)** holds, then we have a valid adjustment set, and the proof is identical to that of Cor. A.3.

Otherwise, if condition **ii)** holds, we have that the interventional query on y equals the observational query when conditioned on $\mathbf{b}$, but that now $\mathbf{b}$ is not independent of $\mathrm{do}(\mathrm{t})$, i.e.,

$$\begin{aligned}
\tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c}) &= \int \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{b} \mid \mathrm{do}(\mathrm{t}), \mathbf{c})\, \mathrm{d}\mathbf{b} && (41)\\
&= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{b} \mid \mathrm{do}(\mathrm{t}), \mathbf{c})\, \mathrm{d}\mathbf{b} && (42)\\
&= \int p(\mathrm{y} \mid \mathrm{t}, \mathbf{b}, \mathbf{c}) p(\mathbf{b} \mid \mathrm{do}(\mathrm{t}), \mathbf{c})\, \mathrm{d}\mathbf{b} && (43)\\
&= p(\mathrm{y} \mid \mathrm{do}(\mathrm{t}), \mathbf{c})\,, && (44)
\end{aligned}$$

where we needed to use that the query $p(\mathbf{b} | \, \mathrm{do}(\mathrm{t}), \mathbf{c})$ is identifiable in the third equality. □

A.2.5 Napkin example

Finally, we want to show one last illustrative example where DeCaFlow provides correct estimates of a causal query that is identifiable by the do-calculus, but neither the backdoor nor the front-door criteria are applicable. While redundant (as the query is identifiable in the classical sense, and then Cor. 4.2 applies), we believe it can be a good exercise to convince the reader. Namely, the graph of Fig. A.5 appears as the napkin graph in Pearl and Mackenzie [53, Fig. 7.5]. What is particularly interesting in this graph is that $\mathbf{w}$ is not a valid adjustment set since, despite blocking the backdoor path from t to y through $\mathbf{b}$, it forms a collider of $\mathbf{z}_1$ and $\mathbf{z}_2$.

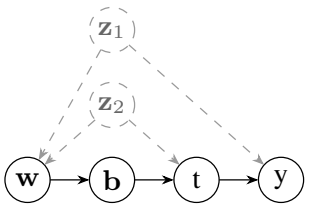


Figure A.5: Napkin causal graph [53].

However, $\mathbf{z}_1$ only affects the outcome and $\mathbf{z}_2$ only affects the treatment. Following from our previous results, the causal effect from t to y should be correctly estimated by DeCaFlow. Here, we show that this is the case. First, let us express the causal query of interest in another form applying do-calculus:

$$p(\mathrm{y} \mid \mathrm{do}(\mathrm{t})) = p(\mathrm{y} \mid \mathrm{do}(\mathrm{y}), \mathrm{do}(\mathrm{t})) = \qquad \text{[\textit{Rule 3 of do-calculus since} } \mathrm{y} \perp\!\!\!\perp_{\mathcal{G}_{\bar{\mathrm{t}},\bar{\mathbf{b}}}} \mathbf{b} \mid \mathrm{t}] \tag{45}$$

$$= p(\mathrm{y} \mid \mathrm{t}, \mathrm{do}(\mathbf{b})) = \qquad \text{[\textit{Rule 2 of do-calculus} } \mathrm{y} \perp\!\!\!\perp_{\mathcal{G}_{\bar{\mathbf{b}},\underline{\mathrm{t}}}} \mathrm{t} \mid \mathbf{b}] \tag{46}$$

$$= \frac{p(\mathrm{y}, \mathrm{t} \mid \mathrm{do}(\mathbf{b}))}{p(\mathrm{t} \mid \mathrm{do}(\mathbf{b}))} \qquad \text{[\textit{Conditional probability}]} \tag{47}$$

Once we have this expression, let us work on the numerator, considering that DeCaFlow is Markov-equivalent with the graph in Fig. A.5:

$$p(\mathrm{y}, \mathrm{t} \mid \mathrm{do}(\mathbf{b})) = \int p(\mathrm{y}, \mathrm{t} \mid \mathbf{b}, \mathbf{w}) p(\mathbf{w}) \, \mathrm{d}\mathbf{w} \qquad \text{[\textit{Backdoor criterion}]} \tag{48}$$

$$= \iiint \tilde{p}(\mathrm{y}, \mathrm{t}, \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2 \mid \mathbf{b}, \mathbf{w}) p(\mathbf{w}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\tilde{\mathbf{z}}_1 \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad \text{[\textit{Eq. marginals}]} \tag{49}$$

$$= \iiint \tilde{p}(\mathrm{y} | \mathrm{t}, \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \mathbf{b}, \mathbf{w}) \tilde{p}(\mathrm{t} | \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2, \mathbf{b}, \mathbf{w}) \tilde{p}(\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2 | \mathbf{w}) p(\mathbf{w}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\tilde{\mathbf{z}}_1 \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad \text{[\textit{Factorization}]} \tag{50}$$

$$= \iiint \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}_2) \tilde{p}(\mathrm{t} \mid \tilde{\mathbf{z}}_2, \mathbf{b}) \tilde{p}(\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2 \mid \mathbf{w}) p(\mathbf{w}) \, \mathrm{d}\mathbf{w} \, \mathrm{d}\tilde{\mathbf{z}}_1 \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad \text{[\textit{Do-calculus rule 1}]} \tag{51}$$

$$= \int \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}_2) \tilde{p}(\mathrm{t} \mid \tilde{\mathbf{z}}_2, \mathbf{b}) \tilde{p}(\tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_2) \, \mathrm{d}\tilde{\mathbf{z}}_1 \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad \text{[\textit{Marginalize} } \mathbf{w}] \tag{52}$$

$$= \int \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}_2) \tilde{p}(\mathrm{t} \mid \tilde{\mathbf{z}}_2, \mathbf{b}) \tilde{p}(\tilde{\mathbf{z}}_1) \tilde{p}(\tilde{\mathbf{z}}_2) \, \mathrm{d}\tilde{\mathbf{z}}_1 \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad [\tilde{\mathbf{z}}_1 \perp\!\!\!\perp_{\mathcal{G}} \tilde{\mathbf{z}}_2] \tag{53}$$

$$= \int \tilde{p}(\mathrm{y} \mid \mathrm{t}, \tilde{\mathbf{z}}_2) \tilde{p}(\tilde{\mathbf{z}}_1) \, \mathrm{d}\tilde{\mathbf{z}}_1 \int \tilde{p}(\tilde{\mathbf{z}}_2) \tilde{p}(\mathrm{t} \mid \tilde{\mathbf{z}}_2, \mathbf{b}) \, \mathrm{d}\tilde{\mathbf{z}}_2 \qquad \text{[\textit{Separate integrals}]} \tag{54}$$

$$= \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t})) \; \tilde{p}(\mathrm{t} \mid \mathrm{do}(\mathbf{b})) \qquad \text{[\textit{DeCaFlow estimate}]} \tag{55}$$

Note also that, as shown in Eq. 40, DeCaFlow correctly estimates $p(\mathrm{t} | \, \mathrm{do}(\mathbf{b}))$. Therefore, if we substitute Eq. 55 in Eq. 47, we have that

$$p(\mathrm{y} \mid \mathrm{do}(\mathrm{t})) = \frac{\tilde{p}(\mathrm{y} | \, \mathrm{do}(\mathrm{t})) \; p(\mathrm{t} | \, \mathrm{do}(\mathbf{b}))}{p(\mathrm{t} | \, \mathrm{do}(\mathbf{b}))} = \tilde{p}(\mathrm{y} \mid \mathrm{do}(\mathrm{t})) \, . \tag{56}$$

That is, we have explicitly shown that DeCaFlow correctly estimates the true causal query $p(\mathrm{y} | \, \mathrm{do}(\mathrm{t}))$.

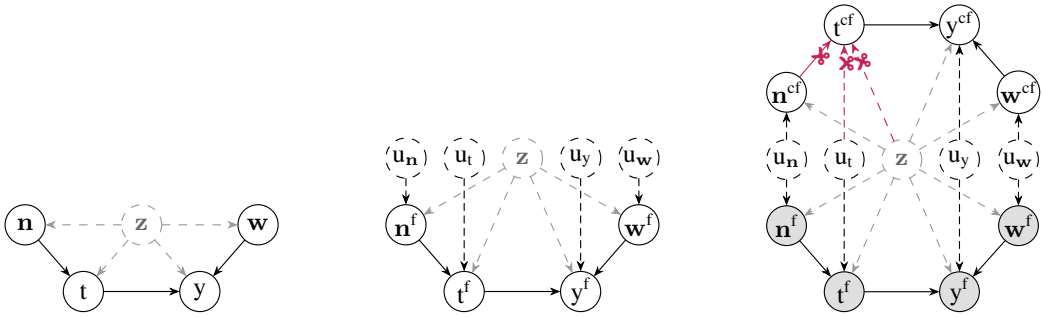


Figure A.6: Example of the transition from **(a)** the regular depiction of a (confounded) SCM, to **(b)** an explicit SCM where the exogenous variables are drawn, and **(c)** a counterfactual twin SCM where the data-generating process is replicated in the "factual and counterfactual worlds". Figure **(c)** also depicts which nodes are observed and which are severed in order to compute a counterfactual query of the type $p(\mathrm{y}^{\mathrm{cf}}|\,\mathrm{do}(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$.

A.3 Counterfactual query identifiability

In this section, we show that counterfactual query identifiability is a direct result of the interventional query identifiability from the previous section.

In order to formally define counterfactuals, in this section we introduce the concept of counterfactual SCMs in a rather untraditional fashion. Namely, we combine the concepts of twin networks from Pearl [55] (which replicates the data-generating process) and that of counterfactual SCMs from Peters et al. [58] (which defines a counterfactual *prior* to the intervention) as follows:

Definition 6 (Counterfactual twin SCM)**.** Given a SCM $\mathcal{M} = (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$, we define its counterfactual twin SCM as a SCM $\mathcal{M}^{\mathrm{cf}}$ where all structural equations are duplicated, and the exogenous noise is shared across replications, and where additionally one of the halves is observed ("the factual world"), and the other half is unobserved ("the counterfactual world").

We provide in Fig. A.6 a more intuitive depiction on the construction of these counterfactual twin networks. From this definition, one can recover the counterfactual SCM defined by Peters et al. [58] by just focusing on the replicated part of the counterfactual twin network, and conditioning the exogenous noise and hidden confounder on the observed half, i.e., $(\mathbf{f}, P_{\mathbf{u}|\,\mathbf{x}^{\mathrm{f}}}, P_{\mathbf{z}|\,\mathbf{x}^{\mathrm{f}}})$. Similarly, one can compute the usual counterfactual query by performing an intervention on the counterfactual twin network, i.e., by replacing the intervened equations by the constant intervened value, and computing the query conditioned on the factual variables, $p(\mathrm{y}^{\mathrm{cf}}|\,\mathrm{do}(\mathrm{t}^{\mathrm{cf}}), \mathbf{x}^{\mathrm{f}})$. This is visually represented in Fig. A.6c.

In order to prove query identifiability in the counterfactual setting, we need to use the following technical result regarding the completeness of a random variable:

Lemma A.6. *If a random variable* $\mathbf{z}$ *is complete given* $\mathbf{n}$ *for almost all* $\mathbf{b}$*, as given by Def. 4, then it is complete given* $\mathbf{n}$ *for almost all* $\mathbf{b}$ *and* $\mathbf{c}$*, where* $\mathbf{c}$ *is another continuous random variable.*

Proof. We prove this result by contradiction. Assume that the result does not hold, then there must exist a non-zero measure subset of the space of $\mathbf{b} \times \mathbf{c}$for which there exists a square-integrable function $g(\cdot)$ such that $\int g(\mathbf{z}, \mathbf{b}, \mathbf{c}) p(\mathbf{z}|\,\mathbf{b}, \mathbf{c}, \mathbf{n})\,\mathrm{d}\mathbf{z} = 0$ for almost all $\mathbf{n}$, but $g(\mathbf{z}, \mathbf{b}, \mathbf{c}) \neq 0$ for almost all $\mathbf{z}$.

Since this subset has positive measure, there must contain an ε-ball within. If we now focus on the $\mathbf{b}$-projection of this ball where we fix $\mathbf{c}$ to its value on the center, we have that it is a subset of non-zero measure in the space of $\mathbf{b}$ (as otherwise it would be zero-measure in the Cartesian-product measure), where the function $g(\cdot, \mathbf{c})$ breaks our initial assumption of the completeness of $\mathbf{z}$. Thus, we reach a contradiction. □

Given Def. 6, it is rather intuitive that, if a causal query is identifiable in a SCM $\mathcal{M}$, then it has to be identifiable in both halves of its induced counterfactual twin SCM $\mathcal{M}^{\text{cf}}$, as they are identical. More importantly, we can now leverage again Prop. A.2, this time with $\mathbf{c} = \mathbf{x}^{\text{f}}$, to prove counterfactual query identifiability whenever we have interventional query identifiability.

Proposition A.7 (Counterfactual identifiability)**.** *Given two SCMs $\mathcal{M} \coloneqq (\mathbf{f}, P_{\mathbf{u}}, P_{\mathbf{z}})$ and $\tilde{\mathcal{M}} \coloneqq (\tilde{\mathbf{f}}, P_{\tilde{\mathbf{u}}}, P_{\tilde{\mathbf{z}}})$, assume that they are Markov-equivalent—i.e., they induce the same causal graph—and that they coincide in their marginal distributions, $p(\mathbf{x}) \overset{a.e.}{=} \tilde{p}(\mathbf{x})$. Then, if a query $p(\mathrm{y}|\, do(\mathrm{t}))$ is identifiable in the sense of Prop. A.2, where $\mathrm{y}, \mathrm{t} \subset \mathbf{x}$, the query $p(\mathrm{y}^{\text{cf}}|\, do(\mathrm{t}^{\text{cf}}), \mathbf{x}^{\text{f}})$ is also identifiable in the induced counterfactual twin SCM as long as the regularity conditions still hold, i.e., if:*

i) $\iint \tilde{p}(\tilde{\mathbf{z}}|\, \mathbf{w}, \mathbf{b}, \mathbf{c}) \tilde{p}(\mathbf{w}|\, \tilde{\mathbf{z}}, \mathbf{b}, \mathbf{c})\, \mathrm{d}\tilde{\mathbf{z}}\, \mathrm{d}\mathbf{w} < \infty$ *for almost all* $\mathbf{b}$, $\mathbf{c}$, *and*

ii) $\int \tilde{p}(\mathrm{y}|\, \mathrm{t}, \mathbf{b}, \tilde{\mathbf{z}}, \mathbf{c})^2 \tilde{p}(\tilde{\mathbf{z}}|\, \mathbf{b}, \mathbf{c})\, \mathrm{d}\tilde{\mathbf{z}} < \infty$ *for almost all* t, $\mathbf{b}$, *and* $\mathbf{c}$.

Proof. We essentially need to prove that the independence and completeness assumptions keep holding when we add the factual covariate, $\mathbf{c} = \mathbf{x}^{\text{f}}$.

For the independence, we need to show that, if we have a set of variables that fulfill the independence conditions from Prop. A.2, then this set of variables keeps holding them if we include $\mathbf{c} = \mathbf{x}^{\text{f}}$. This is, however, easy to show since factual and counterfactual variables only have "tail-to-tail" dependencies, i.e., they are connected only through the shared exogenous variables. As a result, if two variables from the same half are conditionally independent given a third set of variables, conditioning on the other half cannot change this independence.

For the completeness, we need to show that introducing the factual variable retains the completeness assumed in Prop. A.2, which is direct to show using Lemma A.6. Specifically, it holds that

i) $\mathbf{z}$ is complete given $\mathbf{n}$ for almost all t, $\mathbf{b}$, and $\mathbf{c}$, and

ii) $\tilde{\mathbf{z}}$ is complete given $\mathbf{w}$ for almost all $\mathbf{b}$ and $\mathbf{c}$.

Therefore, the requirements of Prop. A.2 hold when we append a factual variable to the twin network, and thus we can reapply all the results from the previous sections to the counterfactual cases. □

Once proven the result above, proving Cor. 4.3 is direct by following the exact same steps as we did in §A.2 to the counterfactual twin network instead of the original network.

It is important to note that, while the results above provide counterfactual identifiability whenever we have interventional identifiability, we still rely on how much of a good approximation the encoder is to the inverse of the decoder in the proposed DeCaFlow model. That is, the quality of the encoder determines how well we can perform the abduction step to compute counterfactuals. This consideration is unique to counterfactuals, as we just have to sample from the prior of $\mathbf{z}$ in the case of interventional queries.

B Experimental details and additional results

This section presents a series of ablation studies designed to answer practical questions about the behavior of DeCaFlow and to justify key design choices. These analyses provide empirical guidance for practitioners, clarifying how model performance depends on factors such as training data size, latent dimensionality, and proxy quality. Beyond validating theoretical claims, the results offer concrete recommendations for effectively applying DeCaFlow in realistic scenarios.

Finally, we include complementary experimental details and extended comparisons with baseline methods, covering dataset descriptions, data-generating processes, and additional quantitative results and visualizations that extend those presented in §5.

B.1 Ablation study on latent dimension and number of proxies

We include here additional results of the ATE error, complementary to those of §5.1. If we observe Fig. B.1, we extract the same conclusion as observing counterfactual error, the causal effect is not recoverable with less than two proxies, and more proxies result in better estimates. On the other hand, the selection of the dimension of the latent space bigger than the true dimension of the latent confounders does not affect the performance negatively.

Overall, these findings indicate that DeCaFlow is robust to latent space over-specification, thanks to KL regularization, and that, in practice, providing more and better proxies leads to more accurate estimation of causal effects even when confounding structure is unknown.

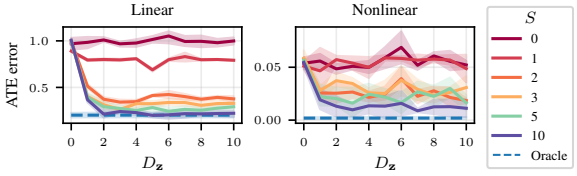


Figure B.1: ATE absolute error as we change the number of proxy variables, S, and the latent dimensionality, $D_{\mathbf{z}}$. We plot mean and 95 % CI over 5 realizations, intervening on the 25th, 50th, and 75th percentile value of t. Oracle represents a causal normalizing flow that observes $\mathbf{z}$.

Details of the generative process. We show the equations that we have used for the ablation study. There exist two unobserved confounders, $\mathbf{z}_1$ and $\mathbf{z}_2$. Note that the proxies available in the nonlinear experiment are bounded or periodic, especially sigmoids and hyperbolic tangents saturate and $\max(0, \mathrm{x})$ loses all the information about the confounder for negative values and sines and cosines are periodic functions. In other words, the distributions $p(\mathbf{z} \mid \mathbf{n}_i)$ are not complete, we lose information about $\mathbf{z}$ when in the transformations to each $\mathbf{n}$. However, if we add more proxies of the confounders, the information that the proxies contain about the confounder is higher, and the causal effect of x_1 on x_2 becomes recoverable.

Linear

$$
\begin{cases}
\mathbf{z}_1 \sim P_{\mathbf{z}_1} \\
\mathbf{z}_2 \sim P_{\mathbf{z}_2} \\
\mathrm{t} = 1.5 \cdot \mathbf{z}_1 + 0.5 \cdot \mathbf{z}_2 + 0.4 \cdot \mathrm{u_t} \\
\mathrm{y} = -0.75 \cdot \mathbf{z}_1 + 0.6 \cdot \mathbf{z}_2 + 0.9 \cdot \mathrm{t} + 0.3 \cdot \mathrm{u_y} \\
\mathbf{n}_1 = -0.5 \cdot \mathbf{z}_1 + 0.3 \cdot \mathbf{z}_2 + 0.5 \cdot \mathrm{u}_2 \\
\mathbf{n}_2 = 0.75 \cdot \mathbf{z}_1 - 0.4 \cdot \mathbf{z}_2 + 0.4 \cdot \mathrm{u}_2 \\
\mathbf{n}_3 = -0.85 \cdot \mathbf{z}_1 + 0.6 \cdot \mathbf{z}_2 + 0.6 \cdot \mathrm{u}_3 \\
\mathbf{n}_4 = 0.6 \cdot \mathbf{z}_1 + 0.6 \cdot \mathbf{z}_2 + 0.55 \cdot \mathrm{u}_4 \\
\mathbf{n}_5 = -0.8 \cdot \mathbf{z}_1 + 0.4 \cdot \mathbf{z}_2 + 0.4 \cdot \mathrm{u}_5 \\
\mathbf{n}_6 = 0.9 \cdot \mathbf{z}_1 - 0.7 \cdot \mathbf{z}_2 + 0.6 \cdot \mathrm{u}_6 \\
\mathbf{n}_7 = -0.72 \cdot \mathbf{z}_1 + 0.5 \cdot \mathbf{z}_2 + 0.56 \cdot \mathrm{u}_8 \\
\mathbf{n}_8 = 0.78 \cdot \mathbf{z}_1 + 0.4 \cdot \mathbf{z}_2 + 0.58 \cdot \mathrm{u}_8 \\
\mathbf{n}_9 = -0.55 \cdot \mathbf{z}_1 + 0.7 \cdot \mathbf{z}_2 + 0.6 \cdot \mathrm{u}_9 \\
\mathbf{n}_{10} = 0.88 \cdot \mathbf{z}_1 + 0.3 \cdot \mathbf{z}_2 + 0.4 \cdot \mathrm{u}_{10}
\end{cases}
$$

Nonlinear

$$
\begin{cases}
\mathbf{z}_1 \sim P_{\mathbf{z}_1} \\
\mathbf{z}_2 \sim P_{\mathbf{z}_2} \\
\mathrm{t} = \dfrac{\mathbf{z}_1^2}{4} \cdot \sin\left(\dfrac{\mathbf{z}_2}{2}\right) + \mathbf{z}_1 + 0.6 \cdot \mathrm{u_t} \\
\mathrm{y} = \dfrac{\mathbf{z}_1 \cdot \mathrm{t}}{4} + 0.8 \cdot \mathbf{z}_2 + 0.5 \cdot \mathrm{t} + \mathrm{x}_1 \cdot \mathrm{u}_2 \cdot 0.3 + 0.2 \cdot \mathrm{u_y} \\
\mathbf{n}_1 = 0.6 \cdot \mathbf{z}_1^2 + \left(\dfrac{\mathbf{z}_2}{4}\right)^3 + 0.3 \cdot \sin\left(\dfrac{\mathbf{z}_2}{2}\right) + 0.5 \cdot \mathrm{u}_1 \\
\mathbf{n}_2 = \sin\left(\dfrac{\mathbf{z}_1}{2}\right) + \cos\left(\dfrac{\mathbf{z}_2}{3}\right) + 0.4 \cdot \mathrm{u}_2 \\
\mathbf{n}_3 = \cos\left(\dfrac{\mathbf{z}_1}{2}\right) - \tanh\left(\dfrac{\mathbf{z}_2}{3}\right) + 0.6 \cdot \mathrm{u}_3 \\
\mathbf{n}_4 = \tanh\left(\dfrac{\mathbf{z}_1}{2}\right) + \sigma\left(\dfrac{\mathbf{z}_2}{2}\right) + 0.55 \cdot \mathrm{u}_4 \\
\mathbf{n}_5 = \sigma\left(\dfrac{\mathbf{z}_1}{2}\right) + \max(0, -\mathbf{z}_2) + 0.4 \cdot \mathrm{u}_5 \\
\mathbf{n}_6 = \max(0, \mathbf{z}_1) - 0.5 \cdot \max(0, \mathbf{z}_2) + 0.6 \cdot \mathrm{u}_6 \\
\mathbf{n}_7 = \max(0, -\mathbf{z}_1) + 0.3 \cdot \max(0, -\mathbf{z}_2) + 0.5 \cdot \mathbf{z}_1 \cdot \mathrm{u}_7 \\
\mathbf{n}_8 = 0.8 \cdot \max(0, \mathbf{z}_1) + 0.3 \cdot \max(0, \mathbf{z}_2) + 0.58 \cdot \mathrm{u}_8 \\
\mathbf{n}_9 = 0.75 \cdot \max(0, -\mathbf{z}_1) + 0.5 \cdot \max(0, \mathbf{z}_2) + 0.6 \cdot \mathrm{u}_9 \\
\mathbf{n}_{10} = 0.3 \cdot \mathbf{z}_1^3 + 0.5 \cdot |\mathbf{z}_2| + 0.4 \cdot \mathrm{u}_{10}
\end{cases}
$$

B.2 Ablation study for encoder selection

We have performed an ablation study for selecting the encoder in the Sachs' dataset, where we evaluate the errors in the estimations of causal queries using a conditional normalizing flow (Flow) and a multilayer

perceptron (MLP) as encoders. We also evaluate the impact of using the warm-up regularization [73] in the KL term. We can observe in Fig. B.2 that we achieve lower errors when applying a regularized flow. This is able to model dependent latent variables and provides a more flexible representation. In addition, we can appreciate that applying the warm-up regularization term is useful and does not produce negative effects.

The improvement achieved by the flow is explained by the following practical aspects of the conditional normalizing flows. First, we can efficiently introduce the factorization proposed in Eq. 3, taking advantage of the structure of the causal graph (see Fig. C.1 for an example), while this factorization implies the use of several MLP. Second, normalizing flows are universal density approximators and do not need to assume specific posterior distributions (i.e. Gaussians). Note that every continuous distribution can be modeled by a conditional normalizing flow, following the Knöthe-Rosenblatt transport [34, 62].

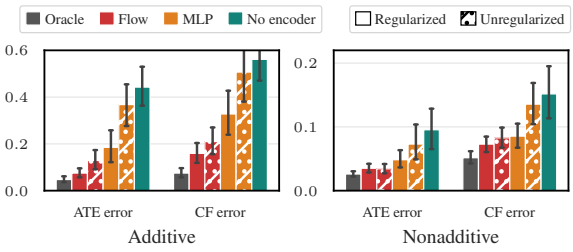


Figure B.2: Ablation for encoder selection in Sachs' dataset. Metrics and 95% CI over 5 realization and all confounded identifiable effects, intervening on percentiles 25, 50 and 75 of each intervened variable. Oracle represents a causal normalizing flow that observes all the confounders.

B.3 Ablation on encoder factorization

Using a conditional normalizing flow as the encoder allows us to model the dependencies between the observations and the posterior of the latent variables as desired.

We propose in Eq. 3 (extended in Eq. 60) a factorization in which each hidden confounder depends on its parents (other hidden confounders), its children and the parents of its children, avoiding cycles. If a child of an unobserved confounder, c, has other parents, then that child is a collider between the hidden confounders and the other parents of c. Therefore, conditioned on c, the hidden confounder is dependent of the other parents of c, given c. That is the reason because we consider sensible to include the other parents of c in the factorization of the hidden confounder, $\mathbf{z}$.

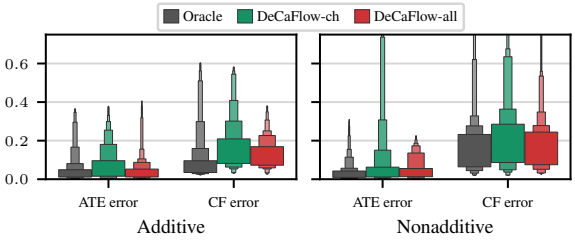


Figure B.3: Ablation for posterior factorization in Ecoli70 dataset. Boxenplots of error metrics in the identifiable edges of Fig. 1. DeCaFlow-ch uses Eq. 57 and DeCaFlow-all uses Eq. 60 for posterior factorization.

However, we also provide an ablation study on the Ecoli70 dataset, where we show that this factorization indeed helps to the estimation of causal queries. Note that in the Ecoli70 dataset, `lacY` is a collider between `eutG` and `cspG`. Therefore, conditioned on `lacY`, the two hidden confounders `eutG` and `cspG` become dependent. The factorization of Eq. 60 implies that the posterior of `cspG` is modeled employing all the children of `cspG` and also the parents of its children, with `eutG` among them. This dependency can be modeled by our encoder in an autoregressive manner.

This factorization incorporates more variables to approximate the posterior of the hidden confounders, compared with a simpler approach that consist in modeling only children dependencies:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \prod_{k=1}^{D_\mathbf{z}} q_\phi\left(\mathbf{z}_k \mid \text{ch}(\mathbf{z}_k)\right) \tag{57}$$

As shown in Fig. B.3, leveraging the factorization of Eq. 60 reduces the errors estimating causal queries in complex graphs, where colliders and dependent hidden confounders are present.

B.4 Ablation on train size

We have proven theoretically that DeCaFlow is able to produce correct estimates of the identifiable causal queries, having that DeCaFlow achieves a perfect matching of the observational distribution, $p_{\mathcal{M}}(\mathbf{x})$.

Although normalizing flows are universal density approximators [50], as a machine learning method, its performance improves as we increase the size of the dataset.

Therefore, to further investigate the behavior of DeCaFlow under varying data availability, we conducted an ablation study on the *training data size*. This analysis allows us to assess how the model's ability to estimate causal queries evolves as the number of observed samples increases. Since the objective of DeCaFlow is to recover the underlying causal mechanisms by matching the observational distributions, it is crucial to understand how data scarcity affects this matching process and, consequently, the accuracy of downstream tasks such as ATE and counterfactual estimation.

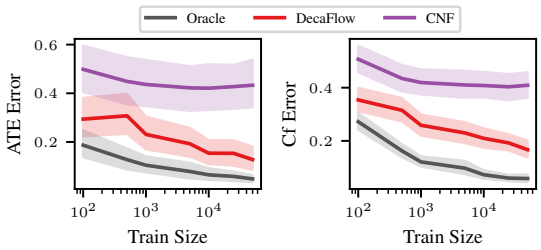


Figure B.4: ATE and Counterfactual error in Sachs' additive dataset, varying the number of train samples. Test size are the same for all realizations. Metrics and 95% CI over 10 realizations and all confounded identifiable effects, intervening in percentiles 25 and 75 of each intervened variable. Oracle represents a CNF that observes the confounders.

Fig. B.4 reports both ATE and CF estimation errors as a function of the training set size. As expected, the errors systematically decrease when more data are available, since the model obtains a more accurate approximation of the data distribution. Notably, DeCaFlow exhibits a similar trend to the oracle, with both ATE and counterfactual errors monotonically decreasing as the number of training samples grows.

In contrast, the CNF (unaware of confounders) also benefits from larger datasets, but shows a slower improvement rate and an earlier plateau, since it does not have guarantees of correct causal estimation even if it matches the observational distribution. These results empirically validate our theoretical claims: as the training distribution approaches to the true observational distribution, the guarantees of DeCaFlow hold, leading to vanishing estimation errors.

B.5 Semi-synthetic Sachs' dataset

This dataset represents a network of protein-signaling in human T lymphocytes. Every variable, except `PKA` and `Plcg` can be intervened upon; therefore, there is not only one causal query of interest, but tens of possible causal queries can arise in this setting. This highlights one of the strengths of DeCaFlow, because we only need a single trained model to answer all identifiable causal queries.

The original data contains a total of 853 observational samples; however, we have decided to evaluate our model on semi-synthetic data because of the following reasons:

- The original network of Sachs et al. [63] contains cycles, which is a violation of one of our assumptions. However, we have found different versions of the causal graph [30, 43] that do not contain cycles. Therefore, we have decided to employ the causal graph that appears in the library *bnlearn* [67]—a recognized library for Bayesian Network learning—as ground truth causal graph. The best way to ensure that the causal graph used is the ground truth is by generating samples according to the causal graph. In addition, that causal graph is the one used by Chao et al. [9].
- We can compare our model with one of the baseline models, DCM, with the same dataset as Chao et al. [9] used.
- Semi-synthetic data allow us to compute all metrics to evaluate causal queries, having the ground truth.

Table B.1: Performance metrics on Sachs datasets. Mean_{std} over five runs and all causal queries of interest. Interventions on `Raf`, `Mek` and `Akt` and evaluating on confounded identifiable effects. Bold indicates significantly better results (95% CI from a Mann-Whitney U test). Lower error values indicate better performance.

		Additive				Non-additive			
	Model	MMD obs $\times 10^4$	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$	MMD obs $\times 10^4$	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$
Oracle	CNF	$4.84_{1.84}$	$7.50_{6.17}$	$6.05_{6.83}$	$10.03_{10.29}$	$5.96_{2.37}$	$6.71_{2.97}$	$2.34_{2.02}$	$4.84_{3.43}$
Aware	DeCaFlow	$\mathbf{2.15_{0.54}}$	$\mathbf{7.04_{3.87}}$	$\mathbf{4.49_{6.76}}$	$\mathbf{12.95_{8.00}}$	$5.12_{2.42}$	$7.58_{16.92}$	$\mathbf{5.16_{5.61}}$	$1.83_{1.65}$
	Deconfounder	–	–	$34.34_{33.45}$	$71.13_{86.98}$	–	–	$8.14_{10.69}$	$63.15_{79.12}$
Unaware	CNF	$5.80_{1.58}$	$73.94_{88.78}$	$44.49_{39.12}$	$56.09_{38.89}$	$5.11_{1.90}$	$12.79_{20.73}$	$9.74_{15.71}$	$15.15_{15.37}$
	ANM	$83.86_{13.41}$	$110.28_{112.43}$	$22.42_{14.06}$	$29.40_{12.22}$	$81.90_{7.21}$	$60.40_{144.08}$	$23.88_{13.94}$	$28.97_{12.44}$
	DCM	$87.80_{2.95}$	$125.59_{118.20}$	$21.21_{11.34}$	$28.25_{6.96}$	$14.23_{4.57}$	$69.74_{390.81}$	$8.44_{7.96}$	$27.50_{23.71}$

- The interventions made in the real world dataset are *soft interventions*, i.e., an external factor is used that modifies one of the variables, changing. On the other hand, DeCaFlow performs *hard interventions*, making it unclear how to compare the two causal queries.

For generating the data in this experiment, we have followed the procedure proposed by Chao et al. [9], where they take the causal graph of Sachs et al. [63] and the empirical distribution of the root nodes, and generate the rest of the variables with random non-linear mechanisms. In addition, exogenous variables have been included in an additive and non-additive manner, respectively.

In the following, we complement the figures presented in §5 with a table that summarizes all the interesting metrics, evaluated on the confounded identifiable causal queries shown in Fig. 8. Interventional distributions and counterfactuals have been computed intervening in percentiles 25, 50 and 75 of the intervened variable.

Since observational MMD is computed only once, the statistics given in Tab B.1 are calculated *only* over 5 runs. On the other hand, we have as many interventional MMDs per run as interventions have been made. However, the statistics of interventional MMD are computed over all the interventions of all intervened variables and 5 runs (5 runs $\times$ 3 intervened variables = 15 samples). Finally, statistics over counterfactual error and ate error aggregate all the intervention-outcome pairs over the five runs. For example, in this case we intervene in 3 variables, performing 3 different interventions and evaluate in 3, 2, and 1 variable, respectively, for each intervened variable, and we have a total of (3+2+1)$\times$3$\times$5 = 90 different measurements to compute the statistics.

The metrics in Tab B.1 indicate that DeCaFlow outperforms all baselines across all interventional and counterfactual causal queries in both settings of the semi-synthetic datasets. However, as discussed in §6, a limitation of our empirical approach is that the differences in observational MMD, the selection criterion for CGMs, are marginal between the *oracle*, DeCaFlow, and CNF. Notably, DeCaFlow even achieves a lower MMD than the *oracle*. This discrepancy arises because the number of variables is large, and the MMD differences are on the order of 10^{-4}.

B.6 Semi-synthetic Ecoli70 dataset

The Ecoli70 dataset represent the gene expression of 46 genes of the RNA sequence of the *Escherichia coli* bacteria. The assumed causal graph comes from the study of [66], which provides insight into the regulatory mechanisms governing *E. coli* gene expression. Examples of interventions in these networks are gene knockout and gene over-expression [41]. A priori, there could be several variables in which intervening can be interesting in evaluating the effects in the cell.

For this experiment, we have generated the data in the same way as done with Sachs' dataset with random mechanisms, but in this case, since we do not have enough samples, root nodes follow standard Gaussian distributions. We have included an additive and a non-additive ways of including exogenous variables. In this

Table B.2: Performance metrics on Ecoli70 dataset. ATE and CF error statistics computed aggregating all causal queries and 5 runs. Intervened and evaluated on the direct confounded identifiable causal effects of Fig. 1. Bold indicates significantly better results (95% CI from a Mann-Whitney U test). Lower error values indicate better performance.

		Additive				Non-additive			
	Model	MMD obs $\times 10^4$	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$	MMD obs $\times 10^4$	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$
Oracle	CNF	$2.34_{0.62}$	$6.05_{5.28}$	$5.04_{7.42}$	$9.91_{12.46}$	$1.49_{0.57}$	$4.05_{8.22}$	$3.51_{4.84}$	$1.67_{1.64}$
Aware	DeCaFlow	$2.42_{0.82}$	$\mathbf{7.04_{3.87}}$	$\mathbf{4.49_{6.76}}$	$\mathbf{12.95_{8.00}}$	$1.58_{0.65}$	$9.22_{22.38}$	$\mathbf{8.79_{17.91}}$	$2.15_{2.10}$
	Deconfounder	–	–	$27.35_{26.17}$	$82.15_{116.90}$	–	–	$30.00_{33.24}$	$9.90_{9.47}$
Unaware	CNF	$2.98_{1.15}$	$10.25_{12.13}$	$23.91_{25.16}$	$34.02_{23.90}$	$1.95_{0.77}$	$10.20_{20.87}$	$12.72_{19.21}$	$2.45_{2.06}$
	ANM	$32.80_{2.81}$	$44.33_{17.62}$	$21.88_{23.89}$	$31.33_{20.64}$	$13.17_{3.95}$	$27.56_{31.57}$	$15.04_{18.18}$	$2.71_{1.88}$
	DCM	$31.65_{0.27}$	$49.50_{36.83}$	$24.45_{33.31}$	$30.22_{24.83}$	$18.78_{6.01}$	$33.37_{36.14}$	$15.07_{22.37}$	$2.36_{2.08}$

case, we have used a semi-synthetic dataset because the real dataset available in *bnlearn* [67] contains only 9 samples.

In Fig. 1 is presented the causal graph of this setting. In addition, note that Fig. 1 has been extracted from our Alg. 6 of causal effect identifiability. That is, we have specified the causal graph and the variables that are unmeasured, and our Algorithm returns (in green) all the paths that are identifiable by DeCaFlow. Consider that black arrows are also identifiable, not only by DeCaFlow, but also for any CGM that approximates the observed data. In red, arrows that are not identifiable by DeCaFlow because there are not enough proxies to infer an unbiased causal effect.

A table summarizing the results obtained in the estimation confounded identifiable causal queries are presented in Tab B.2. The statistics have been computed in the same way as in Sachs' dataset. In the case of ATE and CF error, they have been computed only on the *direct* confounded identifiable paths, i.e., the green paths in Fig. 1.

DeCaFlow significantly outperforms the baselines in ATE and counterfactual estimation in the additive setting and in ATE estimation in the non-additive setting. The MMD differences, both observational and interventional, are negligible between the *oracle*, DeCaFlow, and CNF, likely due to the high number of variables diluting estimation bias. Counterfactual differences in the non-additive setting are also insignificant. However, compared to the *oracle*, the gap between the *oracle* and *unaware* CGMs is smaller than in the additive case. While DeCaFlow reaches an intermediate point, the difference remains insignificant.

B.6.1 Comment on the deconfounder results

One may realize that the errors committed by the deconfounder of [75, 76] are greater than those from unaware models. First, we want to underline that, although the deconfounder allows us to predict counterfactual queries, the algorithm does not present any guarantees of a correct counterfactual estimation since it does not model the exogenous variables of the SCM. We hypothesize this to be the reason behind its performance in counterfactual estimation.

Moreover, let us explain some of the other paths where the errors of the deconfounder are greater than for unaware models. In Sachs' dataset, to model the causal effect `Ekt`→`Akt`, the factorization model of the deconfounder uses `Raf, Mek, Jnk` and `P38` to extract the substitute confounder; the factorization model assumes that all those variables are independent conditioned to $\tilde{z}$, while that is not the case in the true SCM and, therefore, this SCM violates the independence assumption of [75]. The same argument is valid for the paths `yceP`→`yfaD`, `lacA`→`yaeM`, `yceP`→`yfaD`, `ydeE`→`pspA` and `pspB`→`pspA`.

On the other hand, the paths `lacZ`→`yaeM`, `asnA`→`lacY` are frontdoor paths that DeCaFlow can identify because it models the hidden confounder following the true causal graph. However, the deconfounder is not designed to model this paths. To evaluate its performance for frontdoor paths, deconfounder uses the same variables as DeCaFlow to extract the substitute of the confounder. However, the deconfounder assumes independence

Table B.3: Performance metrics on Ecoli70 dataset. Statistics computed an all samples over 5 runs, intervening and evaluating only in the causal effects that deconfounder should solve. Bold indicates significantly better results (95% CI from a Mann-Whitney U test). Lower error values indicate better performance.

	Model	$\lvert\text{ATE err}\rvert \times 10^2$	$\lvert\text{CF err}\rvert \times 10^1$
Oracle	CNF	$8.31_{10.95}$	$1.49_{1.86}$
Aware	DeCaFlow	$\mathbf{7.78_{7.30}}$	$\mathbf{1.87_{1.50}}$
	Deconfounder	$14.35_{15.24}$	$12.03_{15.81}$
Unaware	CNF	$27.82_{30.17}$	$4.01_{3.62}$
	ANM	$27.63_{29.74}$	$3.64_{3.15}$
	DCM	$42.45_{54.23}$	$4.08_{4.12}$

conditioned to the substitute confounder and that is not the case; therefore, we are violating the independence assumption again.

The only two paths that meet the deconfounder assumptions in Fig. 1 are `lacA`→`lacY` and `yedE`→`pspB`. In consequence, we can observe in Fig. B.5 that in those paths, the deconfounder performs at least as well as unaware methods. On the other hand, all the factor models used for the deconfounder implementation (PPCA, Deep exponential families and Variational autoencoder) assume additive noise. Therefore, interventional distributions in non-additive settings are not computable theoretically with these models.

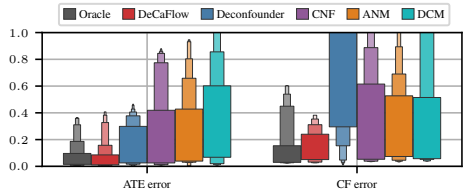


Figure B.5: ATE and CF error evaluating only links where the deconfounder should work in the additive case.

B.6.2 Metrics on the other paths

In this subsection we include a comparison between all the models in the *unconfounded* and the unidentifiable effects. For *unconfounded effects*, our expectation is to observe that all the CGMs achieve a performance comparable with the *oracle*. On the other hand, we expect to have higher errors in unidentifiable effects, since we do not have theoretical guarantees.

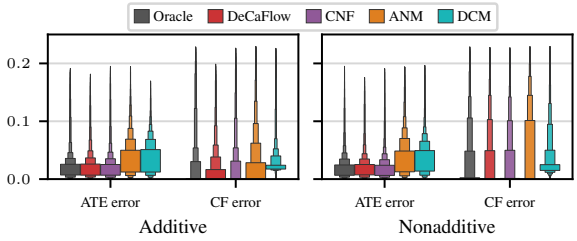


Figure B.6: Error boxenplots on the Ecoli70 dataset for different CGMs, averaged over all *unconfounded* direct effects (see Fig. 1) after intervening in their 25th, 50th, and 75th percentiles and 5 random realizations of the experiment.

Unconfounded Effects. The results for *unconfounded effects* are summarized in Fig. B.6 and Tab B.4, considering only direct effects for ATE and counterfactual error computations. As expected, DeCaFlow and CNF achieve metrics comparable to the *oracle* in both ATE and counterfactual estimations, particularly evident in Fig. B.6, where error distributions are nearly identical. B.4 does not show statistically significative differences between DeCaFlow and CNF. Notably, architectures based on causal normalizing flows outperform ANM and DCM, which model each causal mechanism, f_i, with separate networks. This difference is crucial in settings with many variables and complex relations, where scalability is essential. Unlike ANM and DCM, which suffer from error propagation and limited scalability, causal normalizing flows leverage a single amortized model, making them more efficient in high-dimensional scenarios.

Table B.4: Performance metrics on Ecoli70 dataset. Statistics computed on all *unconfounded* direct effects and 5 runs. Bold indicates significantly better results (95% CI from a Mann-Whitney U test). Lower error values indicate better performance.

	Model	Additive			Non-additive		
		MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$
Oracle	CNF	$3.72_{3.73}$	$2.00_{2.27}$	$1.27_{3.49}$	$1.94_{2.96}$	$1.92_{1.99}$	$1.76_{4.10}$
Aware	DeCaFlow	$4.53_{4.98}$	$2.00_{2.07}$	$1.31_{2.93}$	$2.83_{6.36}$	$1.93_{1.95}$	$1.62_{3.87}$
Unaware	CNF	$4.77_{6.09}$	$2.02_{2.21}$	$1.22_{3.18}$	$2.97_{7.64}$	$1.95_{1.92}$	$1.71_{3.93}$
	ANM	$34.72_{8.56}$	$3.57_{3.02}$	$2.02_{4.09}$	$15.13_{12.57}$	$3.53_{3.15}$	$2.64_{5.34}$
	DCM	$36.23_{14.29}$	$3.48_{2.75}$	$2.69_{2.30}$	$21.22_{13.68}$	$3.42_{2.63}$	$3.00_{3.42}$

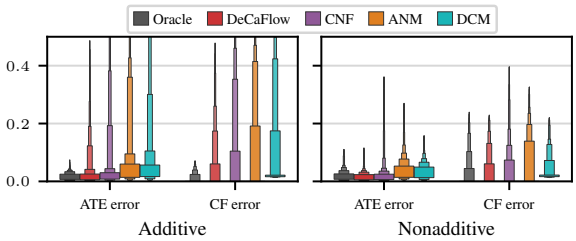


Figure B.7: Error boxenplots on the Ecoli70 dataset for different CGMs, averaged over all unidentifiable direct effects (see Fig. 1) after intervening in their 25th, 50th, and 75th percentiles and 5 random realizations of the experiment.

Finally, note that the deconfounder has not been included in these metrics because it is not designed for *unconfounded queries* and there are many queries, while one deconfounder model is needed for each query.

Unidentifiable Effects. The results for unidentifiable effects—causal queries that violate the assumptions in §4—are summarized in Fig. B.7 and Tab B.5. Notably, the *oracle* performs significantly better than the other CGMs. As seen in Fig. B.7, error distributions are highly skewed, with ATE and counterfactual errors reaching extreme values—considering that metrics are computed on the standardized variables. Tab B.5 shows no significant differences between the metrics achieved by DeCaFlow and CNF.

B.6.3 Hyper-parameters and splits

We have performed a hyperparameter grid search over *validation* data in both experiments on semi-synthetic datasets, exploring a large combination of hyperparameters for each model and dataset.

These are the parameters that were modified for each model:

- CNF: the number of neurons and hidden layers of the single-layer flow, the type of flow (MAF, NSF). LR scheduler reducing on plateau and early stopping were applied with Adam optimizer [32].
- DeCaFlow: number of neurons and hidden layers of the single-layer causal flow (generative network), type of generative network architecture (MAF, NSF), number of neurons and hidden layers of the single-layer encoder flow (inference network), type of encoder architecture (MAF, NSF), KL regularization (True, False). LR scheduler reducing on plateau and early stopping was applied with the Adam optimizer [32].
- Deconfounder: type of factorization model (PPCA, VAE, Deep Exponential Families), number of neurons and hidden layers (in case of deep models), type of outcome model (MLP, random forest, linear regression), number of neurons and hidden layers of the outcome model (in case of deep models).

Table B.5: Performance metrics on Ecoli70 dataset. Statistics computed on all unidentifiable direct effects and 5 runs. Bold indicates significantly better results (95% CI from a Mann-Whitney U test). Lower error values indicate better performance

		Additive			Non-additive		
	Model	MMD int $\times 10^4$	|ATE err| $\times 10^2$	|CF err| $\times 10^3$	MMD int $\times 10^5$	|ATE err| $\times 10^2$	|CF err| $\times 10^2$
Oracle	CNF	$3.71_{3.52}$	$1.79_{1.36}$	$5.88_{15.16}$	$16.98_{6.87}$	$1.75_{1.59}$	$1.62_{4.57}$
Aware	DeCaFlow	$3.80_{3.61}$	$3.95_{7.89}$	$33.62_{80.37}$	$23.02_{21.96}$	$1.75_{1.66}$	$1.88_{4.97}$
Unaware	CNF	$4.54_{4.81}$	$4.75_{10.65}$	$44.76_{126.36}$	$20.22_{6.68}$	$2.32_{3.80}$	$2.13_{6.25}$
	ANM	$34.38_{5.17}$	$7.43_{12.64}$	$52.70_{137.99}$	$130.71_{41.64}$	$4.01_{3.82}$	$2.93_{7.21}$
	DCM	$35.49_{4.95}$	$7.67_{13.93}$	$67.46_{132.21}$	$198.23_{58.62}$	$3.43_{2.76}$	$3.29_{3.92}$

- DCM: number of neurons and hidden layers of each network, learning rate and number of iterations (we have not introduced early stopping or learning rate scheduler). The rest of hyperparameters were selected to the default value in the original code.
- ANM: an automatic search was performed across several models in the original DCM code. This search is performed with the DoWhy package [7].

The selection was based on the matching of the observational for the causal generative models, using MMD. In the deconfounder, the factorization networks were selected by the likelihood of the observed variables and the outcome models with maximum likelihood.

Although including all hyperparameters would be very extensive, we give here a sample of the hyperparameters selected for DeCaFlow in the Ecoli70 additive dataset:

- Hidden neurons of causal flow (generative network): 3×128
- Type of causal flow (generative network): neural spline flow (NSF) [14].
- Hidden neurons of encoder flow (inference network): 3×64
- Type of normalizing flow (inference network): neural spline flow (NSF) [14].
- Regularize: True (warm-up: 30 epochs)
- Total number of parameters: 182k.

Both experiments were performed with 25,000 data, split into $80\%, 10\%, 10\%$ (train, validation, and test). All metrics are given over the *test* dataset, and hyperparameter search was performed over the *validation* dataset.

B.6.4 Processing times

All the experiments were conducted on CPU. Although the experiments were carried out on a cluster of different CPU, we include here two tables for the two semi-synthetic datasets (Tab B.6 and Tab B.7) with the processing times measured in a CPU Intel(R) Core(TM) i7-13650HX laptop, just to show that even in a laptop CPU, the training and inference times are sensible even for large datasets as the Ecoli70 dataset.

Note that DeCaFlow takes more time in training. This is because the network is more complex, due to the inference network, and that we have to sample from the posterior distribution. However, the difference in inference is not that relevant. In fact, DeCaFlow takes less time than the oracle in inference, even when they are sampling the same number of variables (hidden confounders + observed variables). The unaware causal normalizing flow (CNF) only samples from the observed variables. That is why the inference time is lower.

Table B.6: Computation times per model across training and evaluation regimes for Ecoli70 additive dataset. Mean and standard deviation of the training and inference time over 100 epochs in training and over 7 interventions in inference.

Model	Epoch Tr. [s] (20000 samples)	Interventional [s] (2500 samples)	CF [s] (2500 samples)
Oracle	$0.64_{0.06}$	$0.30_{0.02}$	$0.36_{0.03}$
DeCaFlow	$0.98_{0.10}$	$0.28_{0.02}$	$0.35_{0.04}$
CNF	$0.60_{0.07}$	$0.26_{0.01}$	$0.32_{0.05}$

Table B.7: Computation times on the Sachs' Additive Dataset. Mean and standard deviation of the training and inference time over 100 epochs in training and over 3 interventions in inference.

Model	Epoch Tr. [s] (20000 samples)	Interventional [s] (2500 samples)	CF [s] (2500 samples)
Oracle	$0.32_{0.06}$	$0.08_{0.001}$	$0.102_{0.010}$
DeCaFlow	$0.75_{0.12}$	$0.05_{0.004}$	$0.086_{0.005}$
CNF	$0.33_{0.06}$	$0.048_{0.003}$	$0.065_{0.006}$

B.7 Law school fairness use-case

The experiment with real-world data was inspired by Kusner et al. [40] and Javaloy et al. [27]. The goal is to find a fair estimator of the decile of the grades each student will occupy in their third year of university.

The dataset contains information on 27 000 law students who were admitted by the Law School Admissions Council (LSAC) from 1991 to 1997. We have performed an experiment similar to that carried out by Kusner et al. [40], where race and sex were treated as sensitive attributes. We have considered the following variables to include in our study:

- `Race`: binary indicator of the race that distinguish between white and non-white.
- `Sex`: binary indicator of the sex that distinguish between male and female.
- `Fam`: family income.
- `LSAT`: the grade achieved in the Law School Admission Test (LSAT).
- `UGPA`: the undergraduate grade point average (GPA) of the student previous to the admission.
- `FYA`: first-year average grade.
- `Decile3`: the decile of the grades in the third year of university. This is the variable to predict.

We consider that an estimator $\hat{y}$ is fair if it meets *Demographic parity*, defined as follows [40, Def. 3]: A predictor $\hat{y}$ satisfies demographic parity if the predicted distributions for different values of a sensitive attribute are equal: $p(\hat{y} \mid t = 0) = p(\hat{y} \mid t = 1)$. We evaluate the difference between predicted distributions using Maximum Mean Discrepancy (MMD) [21], where a lower distance between the predictions of two sensitive groups denotes a fairer predictor.

The assumed causal graph is slightly different from that of Kusner et al. [40], since their purpose is to make a fair prediction `FYA` accounting only for `Race`, `Sex`, `LSAT` and `UGPA`. However, we include `Fam` and `FYA` as predictors and the task is to predict `Decile3` and the assumed causal graph is the one of Fig. 9.

Proposed fair predictor with DeCaFlow. We propose to model the confounded SCM presented in Fig. B.8, where are explicitly shown the exogenous variables, that are independent of the other variables of the graph except of their associated endogenous variable.

Afterwards, we predict the outcome, `Decile3` from the extracted latent variable that acts as substitute of the `knowledge` and the exogenous variables of `FYA` and `Fam`, following the causal graph of Fig. 9, using a gradient-

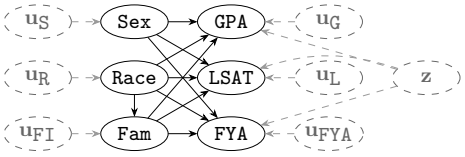


Figure B.8: Confounded SCM modeled by DeCaFlow.

boosted decision tree [17]: $\tilde{p}(\texttt{Decile3} \mid \mathbf{u}_{\texttt{FI}}, \mathbf{u}_{\texttt{FYA}}, \mathbf{z})$. DeCaFlow models $\mathbf{z}$ and the exogenous variables as independent from `Race` and `Sex`. Therefore, the prediction of `Decile3` should be fair.

Baselines. We consider as baselines the methods *Fair K* and *Fair add* proposed by Kusner et al. [40].

Fair K is a fair predictor categorized in Level 2 in Kusner et al. [40], which postulates that the student's knowledge, `know` affects `GPA`, `LSAT`, `FYA` and `Decile 3`, following the distributions described below.

$$
\begin{aligned}
\texttt{Fam} &\sim \mathcal{N}\left(b_{Fam} + w^R_{Fam}\texttt{Race}, 1\right), \\
\texttt{GPA} &\sim \mathcal{N}\left(b_G + w^K_G\texttt{know} + w^R_G\texttt{Race} + w^S_G\texttt{Sex} + w^{Fam}_G\texttt{Fam}, \sigma^2_G\right), \\
\texttt{LSAT} &\sim \text{Poisson}\left(\exp(b_L + w^K_L\texttt{know} + w^R_L\texttt{Race} + w^S_L\texttt{Sex} + w^{Fam}_L\texttt{Fam})\right), \\
\texttt{FYA} &\sim \mathcal{N}\left(w^K_F\texttt{know} + w^R_F\texttt{Race} + w^S_F\texttt{Sex} + w^{Fam}_F\texttt{Fam}, 1\right), \\
\texttt{Decile3} &\sim \text{Poisson}\left(\exp(w^K_D\texttt{know} + w^R_D\texttt{Race} + w^S_D\texttt{Sex} + w^{Fam}_D\texttt{Fam})\right), \\
\texttt{know} &\sim \mathcal{N}(0, 1).
\end{aligned}
\tag{58}
$$

Then, the posterior distribution `know` is inferred using Monte Carlo with the probabilistic programming language Pyro [6]. The outcome is predicted using the inferred `know` using a gradient-boosted decision tree [17]: $\tilde{p}(\texttt{Decile3} \mid \texttt{know})$.

On the other hand, *Fair Add* predicts the outcome from the residuals of predicting each variable with each parent, which guarantees that these residuals are independents of `Race` and `Sex`. That is, the predictor estimates the distribution $p(\texttt{Decile3} \mid \mathbf{r}_{\texttt{Fam}}, \mathbf{r}_{\texttt{UGPA}}, \mathbf{r}_{\texttt{LSAT}}, \mathbf{r}_{\texttt{FYA}})$, where these residuals are computed as:

$$
\begin{aligned}
\mathbf{r}_{\texttt{Fam}} &= \texttt{Fam} - \mathbb{E}[\texttt{Fam} \mid \texttt{Sex}, \texttt{Race}] \\
\mathbf{r}_{\texttt{UGPA}} &= \texttt{UGPA} - \mathbb{E}[\texttt{GPA} \mid \texttt{Sex}, \texttt{Race}, \texttt{Fam}] \\
\mathbf{r}_{\texttt{LSAT}} &= \texttt{LSAT} - \mathbb{E}[\texttt{LSAT} \mid \texttt{Sex}, \texttt{Race}, \texttt{Fam}] \\
\mathbf{r}_{\texttt{FYA}} &= \texttt{FYA} - \mathbb{E}[\texttt{FYA} \mid \texttt{Sex}, \texttt{Race}, \texttt{Fam}]
\end{aligned}
\tag{59}
$$

All predictors used are gradient-boosted decision trees [17].

Discussion of Results. Although the *fair* methods proposed by Kusner et al. [40] achieve significantly better *demographic parity* than our approach using DeCaFlow (as indicated by a much lower MMD), their predictive performance is substantially inferior. Specifically, their performance is comparable to predicting the outcome using only the mean of the distribution, which serves as a baseline in Tab 1. In contrast, DeCaFlow achieves a 98% reduction in MMD while incurring only an 11% increase in RMSE, as illustrated in Fig. 10.

These experiments demonstrate that leveraging DeCaFlow to model confounded Structural Causal Models is beneficial beyond causal query estimation, leading to improved overall performance.

Algorithm 1 KL regularization term in the training loop

function ELBO COMPUTATION(epoch, warmup, θ, ϕ)
  if epoch $<$ warmup:
    $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\,\mathbf{z})] - \beta \cdot \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\,\mathbf{x}) \| \, p(\mathbf{z})]$
  else:
    $\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\,\mathbf{z})] - \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\,\mathbf{x}) \| \, p(\mathbf{z})]$
  return $\mathcal{L}$
end function

C Implementation details

C.1 Posterior factorization of the deconfounding network

DeCaFlow is capable of modeling confounded SCMs that contain several hidden confounders, $\mathbf{z} = \{\mathbf{z}_k\}_{k=1}^{D_{\mathbf{z}}}$, as in the Sachs' dataset (Fig. 8), Ecoli70 dataset (Fig. 1) or the Napkin graph (Fig. A.5). In such cases, the posterior over latent variables factorizes. We propose a factorized posterior in which each hidden confounder is conditioned on its children and on the parents of its children.

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \prod_{k=1}^{D_{\mathbf{z}}} q_\phi \left(\mathbf{z}_k \mid \mathrm{pa}(\mathbf{z}_k) \cup \mathrm{ch}(\mathbf{z}_k) \cup \bigcup_{c \in \mathrm{ch}(\mathbf{z}_k)} (\mathrm{pa}(c) \setminus \{\mathbf{z}_j : j \geq k\}) \right) \tag{60}$$

Since we propose to use a conditional normalizing flow as the encoder, the dependencies between hidden confounders are modeled in an autoregressive manner. The rightmost part of the conditioning set accounts for collider-induced associations: conditioning on a child of $\mathbf{z}_k$, c, makes $\mathbf{z}_k$ dependent on other parents of c. Other parents of c can also be hidden confounders. To model this, a causal ordering of the $\mathbf{z}$ components is assumed to avoid cycles in factorization, but it does not affect estimation, as collider associations have no inherent causal direction.

C.2 Regularization of the Kullback-Leibler term in ELBO

We propose the implementation of a warm-up adaptive regularization term that weights the contribution of the Kullback-Leibler term in the ELBO, to avoid posterior collapse [73]. During training, if the current epoch is lower than the predefined warm-up parameter, the KL term is weighted by β, which we define as $\beta = \min(1, \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\,\mathbf{x}) \| \, p(\mathbf{z})])$, as shown in Alg. 1.

In this way, we encourage the model to focus on data reconstruction on the first epochs, ignoring the KL term if the posterior is very similar to the prior, i.e., if $KL \approx 0$, then $\beta \approx 0$ and $\mathcal{L}(\phi, \theta) \approx \mathbb{E}_{q_{\boldsymbol{\phi}}}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\,\mathbf{z})]$. After the warm-up epoch, the loss is equivalent to the usual expression for the ELBO. We have tested in the ablation study of §B.2 that the inclusion of the regularization term is useful in the Sachs' dataset. On the other hand, when posterior collapse does not occur, the β term will be upper bounded by 1, therefore, not affecting the training process.

C.3 Structural inductive bias

As presented in the original paper by Javaloy et al. [27], the **adjacency matrix** that represents the causal graph is used to build the normalizing flow. In practice, this is implemented following the usual implementation of autoregressive normalizing flows using a Masked Autoencoder for Distribution Estimation (MADE) hypernetwork [18] that uses the causal graph for masking. In this case, we introduce the structural constrains between **i)** exogenous and endogenous variables and **ii)** conditional variables and endogenous variables.

As a result, our deconfounding network factorizes the posterior distribution as shown in Eq. 3, modeling each hidden confounder as a function of its children, its parents and the parents of its children. Similarly, the

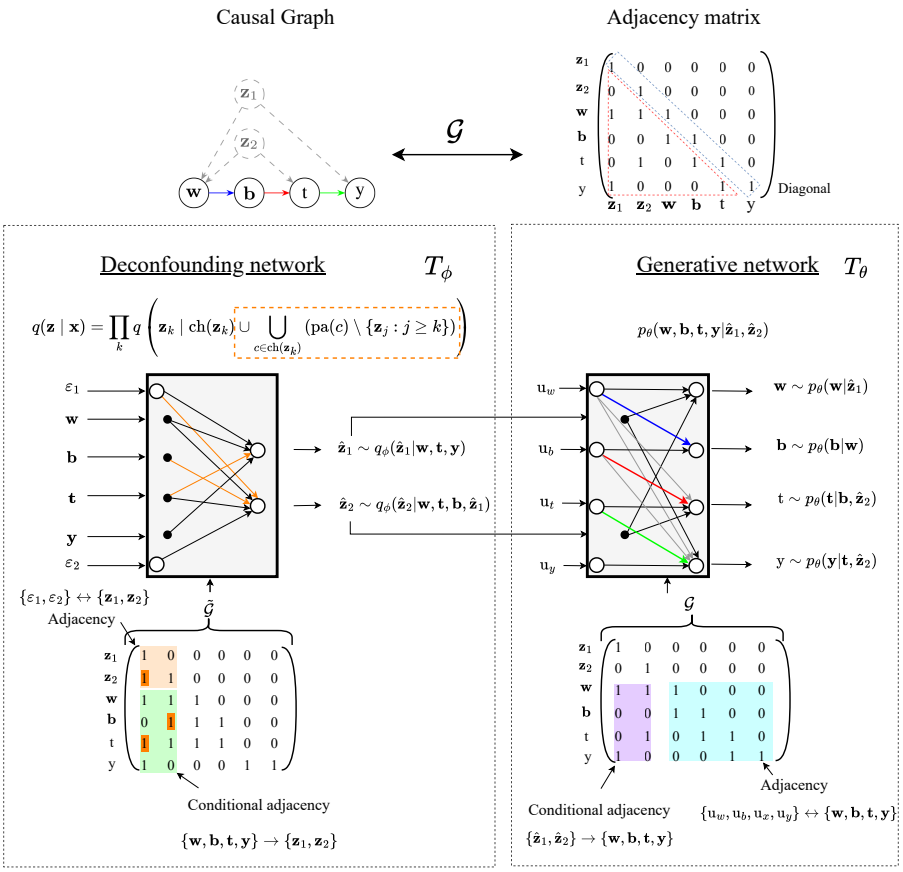


Figure C.1: Complete illustration of DeCaFlow architecture, expanding Fig. 2 applied to the specific graph of Fig. A.5. Both the deconfounding and the generative networks are conditional normalizing flows that factorize the distributions of the posterior and endogenous variables following Eq. 3 and Eq. 2, respectively. Within these networks, functional dependencies are represented following the compacted version from Javaloy et al. [27, Fig. 4(c)]. The orange edges of the encoder corresponds to the collider association in the posterior factorization, and $\tilde{\mathcal{G}}$ encodes that associations.

structural information in the generative network allows us to model each endogenous variable exclusively from its parents, whether these are other endogenous variables or hidden confounders, following Eq. 2.

We include in Fig. C.1 an expanded version of Fig. 2 for the Napkin causal graph (Fig. A.5), where it is shown in detail how its structural constraint is introduced in each conditional normalizing flow. Finally, note that the do-operator is inherited from the CNFs [27], and details on its extension for DeCaFlow can be found in §D.

D Do-operator

We introduce now the algorithms that DeCaFlow employ to generate interventional and counterfactual samples. First, we include those of Javaloy et al. [27]. Note that the notation applied for DeCaFlow is slightly different from the that used for CNFs by Javaloy et al. [27], naming the intervened variable as t, instead of x_i, in order to be consistent with the notation used in §§2 and 4.

D.1 Do-operator in causal normalizing flows

Algorithm 2 Algorithm to sample from $P(\mathbf{x} \mid \mathrm{do}(\mathrm{x}_i = \alpha))$. From Javaloy et al. [27].

function SAMPLEINTERVENEDDIST(i, α)
  $\mathbf{u} \sim P_{\mathbf{u}}$ ▷ Sample a value from the observational distribution.
  $\mathbf{x} \leftarrow T_\theta^{-1}(\mathbf{u})$
  $\mathrm{x}_i \leftarrow \alpha$ ▷ Set x_i to the intervened value α.
  $\mathbf{u}_i \leftarrow T_\theta(\mathbf{x})_i$ ▷ Change the i-th value of $\mathbf{u}$.
  $\mathbf{x} \leftarrow T_\theta^{-1}(\mathbf{u})$
  return $\mathbf{x}$ ▷ Return the intervened sample.
end function

Traditionally the computation of counterfactual samples follows the *abduction, action and prediction* steps postulated by Pearl et al. [57]. The *abduction* step consists of using the observations to determine the value of the exogenous variables. Then, the *action* step computes the intervention, modifying the causal mechanism of the intervened variable and *prediction* consist of using the exogenous variables and the modified SCM to compute the counterfactual. The computation of interventional samples follows a similar pattern, yet the exogenous values are directly sampled, i.e., skipping the abduction step. Javaloy et al. [27] proposed an alternative implementation where, instead of modifying the causal mechanisms in the action step, the distribution of the exogenous variable associated with the intervened variable is changed instead, as described in Algorithms 2 and 3.

Algorithm 3 Algorithm to sample from $P(\mathbf{x}^{\mathrm{cf}} \mid \mathrm{do}(\mathrm{x}_i = \alpha), \mathbf{x}^{\mathrm{f}})$. From Javaloy et al. [27].

function GETCOUNTERFACTUAL($\mathbf{x}^{\mathrm{f}}, i, \alpha$)
  $\mathbf{u} \leftarrow T_\theta(\mathbf{x}^{\mathrm{f}})$ ▷**Abduction:** Get $\mathbf{u}$ from the factual sample.
  $\mathrm{x}_i{}^{\mathrm{f}} \leftarrow \alpha$ ▷**Action:** Set x_i to the intervened value α.
  $\mathbf{u}_i \leftarrow T_\theta(\mathbf{x}^{\mathrm{f}})_i$ ▷**Action:** Change the i-th value of $\mathbf{u}$.
  $\mathbf{x}^{\mathrm{cf}} \leftarrow T_\theta^{-1}(\mathbf{u})$ ▷**Prediction:** Get counterfactual
  return $\mathbf{x}^{\mathrm{cf}}$ ▷ Return the counterfactual value.
end function

D.2 Do-operator in interventional distributions with DeCaFlow

The sampling process consists of first sampling from the prior distribution of the latent variables and from the exogenous distribution. Then, one can use the generative network ($T_{\boldsymbol{\theta}}$) to generate interventional sampling, changing the components of $\mathbf{u}$ associated with t as described in the previous section for CNFs. Note that $\mathbf{z}$ is not an input of the normalizing flow, but a condition (or *context*). Therefore, $\mathbf{z}$ is transformed neither in the forward nor reverse pass of the normalizing flow.

Algorithm 4 Algorithm to sample from the interventional distribution, $P(\mathbf{x} \mid \text{do}(\text{t} = \alpha))$ with DeCaFlow.

function SAMPLEINTERVENEDDIST(t, α)
  $\mathbf{z} \sim P_{\mathbf{z}}$ ▷ Sample a value from the prior of $\mathbf{z}$.
  $\mathbf{u} \sim P_{\mathbf{u}}$ ▷ Sample a value from the observational distribution.
  $\mathbf{x} \leftarrow T^{-1}_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{u})$
  $\text{t} \leftarrow \alpha$ ▷ Set t to the intervened value α.
  $\text{u}_\text{t} \leftarrow T_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{x})_\text{t}$ ▷ Change the component of $\mathbf{u}$ associated with t.
  $\mathbf{x} \leftarrow T^{-1}_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{u})$
  return $\mathbf{x}$ ▷ Return the intervened sample.
end function

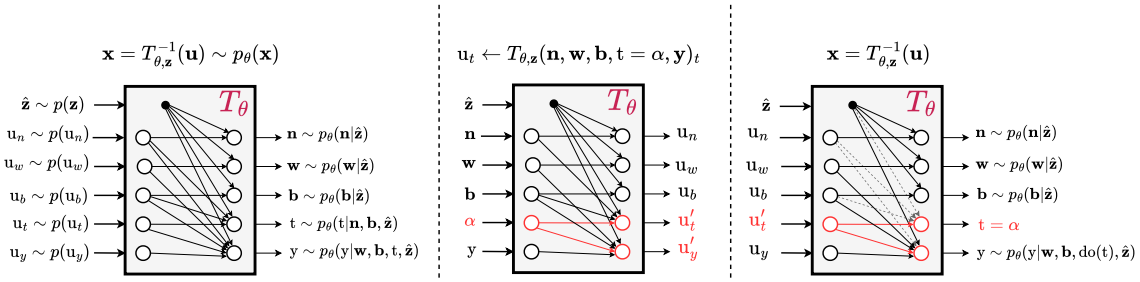


Figure D.1: Schematic of the sampling process for an interventional distribution using the graph from Fig. 3 and intervening in t. By sampling from the prior of the hidden confounders, $p(\mathbf{z})$, and the exogenous distribution, $p(\mathbf{u})$, we obtain samples of the interventional distribution by appropriately setting u_t, i.e., samples from $p_\theta(\text{y}|\,\text{do}(\text{t}))$. Note that sampling from the interventional distribution only requires the generative network, $T_{\boldsymbol{\theta}}$. Dashed gray arrows represent the cancellation of causal effects due to the intervention.

As we can easily sample from interventional distributions, we compute the average treatment effect (ATE) via Monte Carlo. For example, to compute the ATE comparing two interventions (α_1, α_2) in the variable t, we would generate samples of both interventional distributions, $p(\mathbf{x} \mid \text{do}(\text{t} = \alpha_1)), p(\mathbf{x} \mid \text{do}(\text{t} = \alpha_1))$, and approximate their expectations by taking the sample average:

$$\text{ATE}_{\mathbf{x}}(\alpha_1, \alpha_2) = \mathbb{E}[\mathbf{x} \mid \text{do}(\text{t} = \alpha_2)] - \mathbb{E}[\mathbf{x} \mid \text{do}(\text{t} = \alpha_1)] \tag{61}$$

$$\approx \left(\frac{1}{N} \sum_{\mathbf{x} \sim P(\mathbf{x}|\,\text{do}(\text{t}=\alpha_2))} \mathbf{x} \right) - \left(\frac{1}{N} \sum_{\mathbf{x} \sim P(\mathbf{x}|\,\text{do}(\text{t}=\alpha_1))} \mathbf{x} \right) \tag{62}$$

If we were interested in the ATE of a subset of variables, e.g., y, we would simply need to generate samples of $\mathbf{x}$ and take only those from the variable of interest, y.

D.3 Do-operator in counterfactuals with DeCaFlow

As part of the abduction step, our model estimates the posterior distribution of hidden confounders given a factual datapoint, $q_\phi(\mathbf{z} \mid \mathbf{x}^\text{f})$. Therefore, we can sample from the inferred posterior of the hidden confounders, and use those samples as the context for the generative network.

Algorithm 5 Algorithm to sample from the counterfactual distribution, $P(\mathbf{x} \mid \text{do}(\text{t} = \alpha))$ with DeCaFlow.

function GETCOUNTERFACTUAL($\mathbf{x}^{\text{f}}, \text{t}, \alpha$)
  $q_\phi(\mathbf{z} \mid \mathbf{x}^{\text{f}}) \leftarrow$ Deconfounding network($\mathbf{x}^{\text{f}}$) ▷ **Abduction:** Get $\mathbf{z}$ from the factual sample.
  $\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x}^{\text{f}})$ ▷**Abduction**: Sample the posterior distribution.
  $\mathbf{u} \leftarrow T_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{x}^{\text{f}})$ ▷**Abduction:** Get $\mathbf{u}$ from the factual sample.
  $\text{t}^{\text{f}} \leftarrow \alpha$ ▷**Action:** Set t to the intervened value α.
  $\text{u}_{\text{t}} \leftarrow T_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{x}^{\text{f}})_{\text{t}}$ ▷**Action:** Change the component of $\mathbf{u}$ associated with t.
  $\mathbf{x}^{\text{cf}} \leftarrow T^{-1}_{\boldsymbol{\theta},\mathbf{z}}(\mathbf{u})$ ▷ **Prediction:** compute the counterfactual
  return $\mathbf{x}^{\text{cf}}$ ▷ Return the counterfactual value.
end function

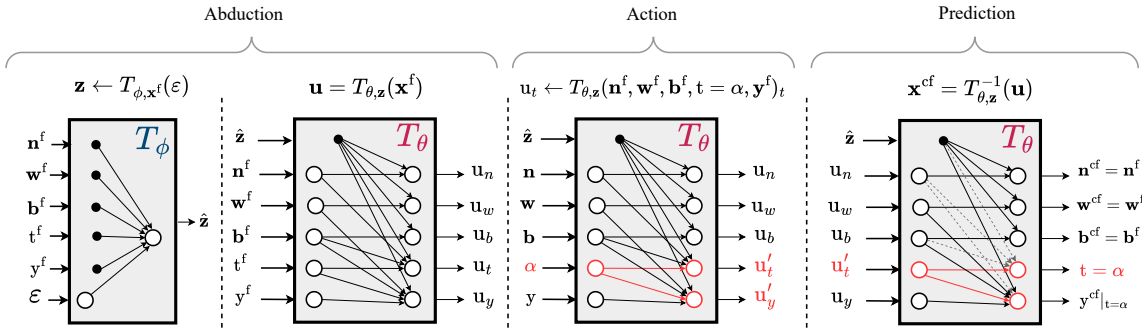


Figure D.2: Schematic of the process of performing counterfactual inference with the causal graph from Fig. 3 intervening in t. Both the deconfounding network, $T_{\boldsymbol{\phi}}$, and the generative network, $T_{\boldsymbol{\theta}}$, are needed to generate counterfactual samples. Dashed gray arrows represent the cancellation of causal effect due to the intervention.

E Additional details on related work of causal inference with hidden confounders

E.1 Methods tailored to graph and query

First, we want to remark that all the following methods are designed to address causal inference queries in specific causal graphs (or sub-graphs), therefore they can be used when these causal relationships hold. We summarize the causal graphs assumed by each work in Fig. E.1. In the following, we assume the notation introduced in §2, where $\mathbf{z}$ is the hidden confounder, t is the intervened variable (or treatment) and y is the outcome, i.e., the variable where we want to evaluate the causal effects.

We have classified the different approaches depending on the graph that they are designed to address. However, there are two considerations that are common for all these approaches. First, these methods follow a two-stage process: **i)** extracting a substitute of the unobserved confounder using variables affected by the confounder or instrumental variables, $\tilde{\mathbf{z}}$, and **ii)** estimating the outcome given this substitute, $\tilde{\text{y}} \sim p(\text{y} \mid \tilde{\mathbf{z}}, \text{t})$. In this case, one predictor must be trained per outcome, as well as one extractor per independent confounder. Second, none of these methods estimate *counterfactual distributions*, since they do not model exogenous variables.

Presence of null proxies independent of t **(Fig. E.1a).** We say $\mathbf{n}$ to be a null proxy of $\mathbf{z}$ if it is a child of $\mathbf{z}$ independent of the outcome, y, given $\mathbf{z}$, i.e., $\mathbf{n} \perp\!\!\!\perp \text{y} \mid \mathbf{z}$. When null proxies of the confounder are available and they are also independent of the intervened variable, $\mathbf{n} \perp\!\!\!\perp \text{t} \mid \mathbf{z}$, these proxies can be used to infer a substitute of the hidden confounder. Among these works, Allman et al. [2] and Kuroki and Pearl [39] study the case in

which the confounder is categorical, and use matrix factorization to extract a substitute when, either, there exist three Gaussian proxies [2], when the conditional distribution of the confounder given the proxy is known [39], or when other proxies are available [39]. Kallus et al. [28] also employ matrix factorization for cases where the confounder is continuous and the relation with the covariates and treatment (but not with the outcome) is linear. Similarly, Kallus et al. [29] uses kernel functions to extract the substitute confounder when the generators are nonlinear. The most relevant method based on deep generative methods is the one proposed by Louizos et al. [42], where a variational autoencoder (VAE) is used to extract the substitute confounder when several null proxies are available, although no theoretical guarantees were provided and it was later shown to struggle in practice with complex distributions [61]. Finally, Miao et al. [46] offer a regression-based approach to estimate the unobserved confounder under *equivalence*, which assumes that any model of the joint achieves element-wise transformations of the latent variables, something that is not feasible to check: $\tilde{p}(\mathrm{t}, \mathbf{z} \mid \mathbf{n}) = p(\mathrm{t}, V(\mathbf{z}) \mid \mathbf{n})$.

Presence of two proxies: null and not null (Fig. E.1b). When the null proxies affect treatment (notice that in Fig. E.1b the proxy $\mathbf{n}$ affects the treatment t), Miao et al. [45] offer theoretic guarantees of causal identifiability in the presence of another proxy, $\mathbf{w}$, and completeness conditions. The proxy $\mathbf{w}$ can be active, that is, it can directly affect y. Then, Tchetgen et al. [71] introduced the two-stage proximal least squares (P2SLS), which infers the substitute confounder from $p(\mathbf{w} \mid \mathrm{t}, \mathbf{n})$. P2SLS can be implemented using neural networks to achieve greater flexibility. Several works have followed-up the ideas introduced by Miao et al. [45], aiming to estimate the bridge function, i.e., finding an explicit form for the function $\tilde{h}$ shown in Eq. 13. For example, Cui et al. [11] designed a doubly-robust estimator of the ATE by estimating the bridge function semiparametrically, and Mastouri et al. [44] and Kompa et al. [37] applied moment restrictions to estimate the bridge function using deep neural networks. Other works have proposed multiple-robust methods when the confounders are categorical [68].

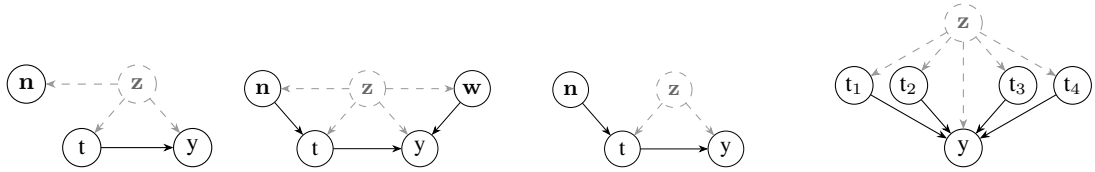
Instrumental variable (Fig. E.1c). Another condition that enables causal inference is the presence of instrumental variables (IVs), i.e. variables that affect only the treatment and are independent of both the unobserved confounder and the outcome, given the treatment (in Fig. E.1c, $\mathbf{n}$ is an IV). In the linear case, Pearl [55] and Angrist and Pischke [4] demonstrated how a two-stage regression process can mitigate the confounding bias, as the only effect that occurs from the IV to the outcome is through the treatment variable. A substitute of the confounder is then extracted by computing the conditional distribution of the treatment given the IV, i.e., $\tilde{\mathbf{z}} \sim p(\mathrm{t} \mid \mathbf{n})$. Furthermore, Hartford et al. [22] extended this idea to include arbitrarily complex nonlinear data-generating processes, designing a two-step deep approach based on neural networks.

Multitreatment affected by a common confounder (Fig. E.1d). Finally, the multitreatment scenario has been studied by Wang and Blei [75] and Ranganath and Perotte [60], where it is called multitreatment since all covariates can be seen as treatments over the outcome variable, y. Here, it is assumed that in the true causal model there exist several covariates that are independent given the unobserved confounder. Therefore, Wang and Blei [75] proposed to use a factorization model to infer the substitute confounder, such as probabilistic PCA or Poisson matrix factorization. In short, a factorization model assumes that the distribution of all the treatments factorizes as follows: $p(\mathbf{t}, \mathbf{z}) = p(\mathbf{z}) \prod_{i=1}^{d} p(\mathrm{t}_i \mid \mathbf{z})$, which should allow to construct a substitute of the confounder from the posterior of $\mathbf{z}$: $\tilde{\mathbf{z}} \sim \tilde{p}(\tilde{\mathbf{z}} \mid \mathbf{t})$. Later, D'Amour [12] provided counterexamples showing that the deconfounder does not achieve nonparametric identification without additional assumptions and, notably, one of the alternatives proposed by D'Amour [12] highlights the use of proxy variables, which is the approach adopted by DeCaFlow.

Similar to Wang and Blei [75], Ranganath and Perotte [60] proposed to use a VAE as the factorization model, adding a regularization term to reduce the additional mutual information between the estimated confounder and the treatment t_j, given the rest of treatments, $\mathbf{t}_{-j}$. However, the theoretical guarantees of this approach require an infinite number of treatments to achieve unbiased estimates of the causal effects. Wang and Blei [76] connect the ideas of Miao et al. [45] and Wang and Blei [75] ensuring causal identification in the multitreatment setting when we know that some of the treatments *can act as null proxies*, that is, when they do not affect the outcome. This assumption allows them to provide theoretical guarantees when the number

of treatments does not tend to be infinite. In despite of that, a factorization model such as the one Wang et al. [77] propose can only model independent treatments given the hidden confounder, which greatly limits its practical utility.

What is the relation of the deconfounder Wang and Blei [75, 76] with DeCaFlow? Similar to DeCaFlow, the deconfounder infers the posterior distribution of the confounder substitute from observational data using a generative model. However, the application of a factorization model restricts the structural dependencies that the it can model. For example, the deconfounder cannot model the structural dependencies of Fig. E.1b, since the factorization model assumes $\mathbf{n} \perp\!\!\!\perp \mathrm{t} \perp\!\!\!\perp \mathbf{w} \mid \mathbf{z}$. In contrast, DeCaFlow leverages CNFs which can model these dependencies since the causal graph is encoded in the normalizing flow architecture. It is also important to stress that DeCaFlow models the whole confounded SCM, including the exogenous variables. This allows us to compute *counterfactuals* and train in a query-agnostic manner. In contrast, the deconfounder cannot compute counterfactuals and needs of a separate model per causal query.



(a) One proxy. (b) Two proxies. (c) Instrumental variable. (d) Multitreatment.

Figure E.1: Graphs assumed by prior works. **(a)** Allman et al. [2], Kallus et al. [28, 29], Kuroki and Pearl [39], Louizos et al. [42], Miao et al. [46] address the case where $\mathbf{n}$ is independent of t. **(b)** Miao et al. [45] assumes the case where there exist two proxies. **(c)** Graph with an instrumental variable. **(d)** Ranganath and Perotte [60], Wang and Blei [75, 76] work with the multitreatment setting.

E.2 CGM with unobserved confounders

There exist several works that employ causal generative models (CGMs) in the presence of hidden confounders. We explain here the differences with our proposal, highlighting the practical advantages of DeCaFlow.

Neural Causal Models (NCMs). Xia et al. [81] proposed a class of sequential causal generative models where each structural equation—i.e., the functional relationship between a variable and its parents in the causal graph—is modeled by a different neural network. The model is trained end-to-end to jointly learn all structural mechanisms. Beyond estimation, NCMs aim to determine whether a given causal query is identifiable from the data-generating process. To assess identifiability, two NCMs are trained: one that maximizes the causal query, subject to a perfect observational fitting, and one that minimizes it. If both NCMs yield the same outcome, the query is deemed identifiable. Interestingly, this approach formalizes identifiability as an empirical condition based on optimization agreement.

However, the framework presents significant practical constraints: **i)** it only supports finite discrete variables, typically binary and low-dimensional, due to tractability constraints; **ii)** it assumes that the true observational distribution is available for training; **iii)** two NCMs are trained per query, leading to high computational cost; and **iv)** identifiability is only revealed post-training, offering no guidance before the model is trained. To perform counterfactual inference, Xia et al. [82] extended NCMs to estimate queries involving latent exogenous variables. However, their approach relies on rejection sampling to perform the abduction step, which is inefficient and unsuitable for continuous or high-dimensional settings, thus limiting its applicability in real-world scenarios.

In contrast, DeCaFlow addresses these limitations. First, we provide a principled criterion to estimate the identifiability of a query *prior* to model training. Second, our framework supports continuous variables and scales to high-dimensional settings. Third, we train a single model that jointly estimates all causal mechanisms and enables efficient inference of counterfactual queries. Fourth, we use variational inference

to approximate the posterior of hidden confounders, avoiding the inefficiency of rejection-based methods. Finally, we guarantee the identifiability of unconfounded exogenous variables (in the sense of Xi and Bloem-Reddy [80]) by leveraging the theoretical framework of CNFs [27]. As a result, DeCaFlow is substantially more efficient and suited for real-world applications.

Modular Causal Generative Models. Rahman and Kocaoglu [59] introduce a modular framework for high-dimensional causal inference, where variables influenced by the same hidden confounder are modeled jointly in end-to-end submodules. A key advantage of this approach is the ability to incorporate pretrained models into submodules, enabling flexible modeling of complex or structured variables when the modular criterion holds. The method supports continuous and discrete variables and uses adversarial training to match observational distributions. Symbolic identifiability is computed using the algorithm of Jaber et al. [26], and they prove that identifiable queries remain estimable under their modular decomposition. However, the framework does not support counterfactual inference nor proximal learning, and it relies on adversarial optimization.

In comparison, DeCaFlow trains a single end-to-end model, estimates both observational and counterfactual distributions also in proximal settings, and enables efficient inference with broad applicability to real-world settings.

Counterfactual Identifiability of Bijective Causal Models. Nasr-Esfahany et al. [48] propose a sequential causal model using conditional normalizing flows to map exogenous to endogenous variables. The model focuses on counterfactual inference under backdoor and instrumental variable (IV) settings, with identifiability proven only for discrete variables. Proxy variables are not considered, and the use of invertible mappings over discrete domains makes theoretical claims less robust. Although the model claims support for continuous data, guarantees are restricted to discrete IV scenarios. Moreover, it does not model observational nor interventional distributions, and lacks parameter amortization due to its sequential structure.

In contrast, DeCaFlow supports continuous variables, models both observational and interventional distributions, and enables counterfactual inference under general confounding and proxy settings. It also requiresa single end-to-end model and scales efficiently to real-world data.

Learning Functional Causal Models with Generative Neural Networks. Goudet et al. [19] propose a method for causal discovery rather than causal inference under unobserved confounding. Given a Markov equivalence class (or graph skeleton), their approach uses generative neural networks to model each causal direction, selecting the graph that best matches the observational distribution evaluated via maximum mean discrepancy (MMD). The model is trained sequentially and assumes no hidden confounders. While not directly comparable to our work, such causal discovery tools may serve as a preprocessing step when the causal graph is unknown, enabling downstream application of models—such as ours—that assume a known and correct structure.

F Algorithms for causal query identification

As explained in §4.2, we can ask DeCaFlow to estimate any causal query, but we do not have the guarantee that the estimation DeCaFlow does is correct unless the query is identifiable. Therefore, we provide the practitioner with algorithms to check the identifiability of causal queries.

Specific treatment-outcome pair. We start presenting in Alg. 6 an algorithm to identify a causal query, given a pair of treatment and outcome variables, which is valid for estimating the interventional distribution of the outcome, $p(\mathrm{y}|\,\mathrm{do}(\mathrm{t}), \mathbf{c})$, and the counterfactual one, $p(\mathrm{y}^{\mathrm{cf}}|\,\mathrm{do}(\mathrm{t}), \mathbf{x}^{\mathrm{f}})$, as we postulated in §4 that the latter is identifiable if the former is.

We have employed Alg. 6 on all direct paths of the Sachs and Ecoli70 datasets to check their identifiability, in order to get a visual representation of the queries that DeCaFlow can estimate in such complex graphs. If one is interested in evaluating a query which involves several outcomes, $\{\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_O\}$, one causal query per outcome variable should be evaluated.

Algorithm 6 Identification of causal queries that include intervention and outcome (t, y)

Require: Graph $\mathcal{G}$, intervention variable t, outcome variable y, covariates $\mathbf{c}$, hidden variables $\mathbf{z}$
Ensure: Boolean indicating if query is identifiable
$\mathbf{z} \leftarrow$ hidden variables that are parents of both t and y
return True **if** $\mathbf{z}$ is $\emptyset$ ▷ Unconfounded is identifiable
for all $\mathbf{z}_k \in \mathbf{z}$ **do**
  Comment: Each $\mathbf{z}_k$ is an independent component of $\mathbf{z}$
  $\mathbf{n}$-proxies $\leftarrow$ children of $\mathbf{z}_k$ d-separated from t given $(\mathbf{z}, \mathbf{c})$
  $\mathbf{w}$-proxies $\leftarrow$ children of $\mathbf{z}_k$ d-separated from y given $(\mathbf{z}, \mathbf{c})$
  if there exist $\mathbf{n} \in \mathbf{n}$-proxies and $\mathbf{w} \in \mathbf{w}$-proxies such that $\mathbf{n}$ is d-separated from $\mathbf{w}$ given $(\mathbf{z}, \mathbf{c})$ **then**
    $\mathbf{z}_k$ is deconfounded
  end if
end for
return all $\mathbf{z}_k$ are deconfounded

Evaluation on all the variables. Although Alg. 7 consists of iteratively applying Alg. 6, we also find it interesting to include the extension to identify causal queries evaluated on all variables in the dataset, which is useful for the case where we DeCaFlow as a generative model for the joint interventional distribution, $p(\mathbf{x} \mid \text{do}(\text{t}))$, or to generate joint counterfactual samples intervening in a specific variable, $\text{t} \subset \mathbf{x}$, $p(\mathbf{x}^{\text{cf}} \mid \text{do}(\text{t}), \mathbf{x}^{\text{f}})$.

Algorithm 7 Identification of causal queries, intervening in t and evaluating in all variables

Require: Graph $\mathcal{G}$, intervention variable t, hidden variables $\mathbf{z}$
Ensure: Boolean indicating if the interventional distribution is identifiable
$\mathbf{z} \leftarrow$ hidden variables that are parents of t
for all $\text{x}_i \in$ descendants of t **do**
  Comment: Evaluate only on descendants of the intervention
  Check (t, x_i) identifiability with Alg. 6
end for
return all $(\text{t}, \mathbf{x}_i)$ are identifiable

F.1 Pipeline for using DeCaFlow

Our framework provides a systematic approach to estimating causal queries by integrating DeCaFlow, a model trained on observational data, with algorithms designed for query identifiability analysis.

As depicted in the pipeline, the framework takes as input a dataset $\mathcal{D}$, a causal graph $\mathcal{G}$, and a set of N interesting queries $\{Q_i\}_{i=1}^N$. The process begins by training DeCaFlow on $\mathcal{D}$ and $\mathcal{G}$, enabling it to learn the confounded SCM, $\mathcal{M}$.

Simultaneously, the identifiability of each causal query Q_i is assessed using dedicated algorithms (Alg. 6 and Alg. 7). If Q_i is identifiable, the trained DeCaFlow is used to estimate $Q_i(\mathcal{M})$ (Alg. 4 and Alg. 5), yielding the estimated causal effect $\hat{Q}_i(\mathcal{M})$. If Q_i is not identifiable, the framework indicates that answering the query is not feasible given the available data and causal structure. Other causal queries can be answered by the model without retraining, provided that their identifiability is verified beforehand.

This workflow ensures a principled approach to causal inference, leveraging both data-driven modeling and theoretical guarantees on identifiability.

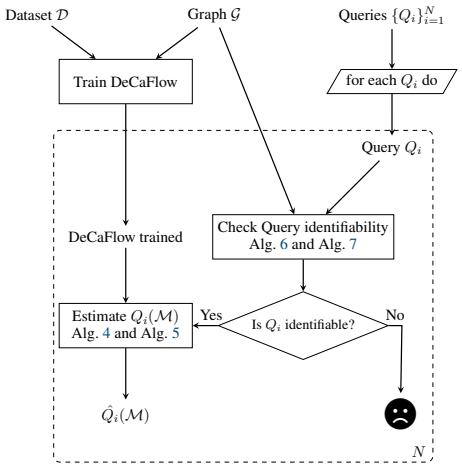


Figure F.1: **Block diagram of our pipeline.**

Validation with interventional data. As a final step in the pipeline for real-world scenarios, especially in sensitive applications, we encourage practitioners to validate the framework with interventional data. Causal queries such as the *average treatment effects* (ATEs) can be validated if a randomized experiment is available in which interventions are carried out on the treatment variable.