CAUSALPROFILER: GENERATING SYNTHETIC BENCHMARKS FOR RIGOROUS AND TRANSPARENT EVALUATION OF CAUSAL MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal machine learning (Causal ML) aims to answer “what if” questions using machine learning algorithms, making it a promising tool for high-stakes decision-making. Yet, empirical evaluation practices in Causal ML remain limited. Existing benchmarks often rely on a handful of hand-crafted or semi-synthetic datasets, leading to brittle, non-generalizable conclusions. To bridge this gap, we introduce CausalProfiler, a synthetic benchmark generator for Causal ML methods. Based on a set of explicit design choices about the class of causal models, queries, and data considered, the CausalProfiler randomly samples causal models, data, queries, and ground truths constituting the synthetic causal benchmarks. In this way, Causal ML methods can be rigorously and transparently evaluated under a variety of conditions. This work offers the first random generator of synthetic causal benchmarks with coverage guarantees and transparent assumptions operating on the three levels of causal reasoning: observation, intervention, and counterfactual. We demonstrate its utility by evaluating several state-of-the-art methods under diverse conditions and assumptions, both in and out of the identification regime, illustrating the types of analyses and insights the CausalProfiler enables.

1 INTRODUCTION

Causal machine learning (Causal ML) seeks to estimate the effects of interventions and counterfactuals using machine learning techniques (Kaddour et al., 2022), enabling principled decision making—for example in medicine and policy. Despite the theoretical maturity and growing relevance of Causal ML, current research practices lack rigourous evaluations of how proposed methods would perform under realistic and diverse conditions, limiting their practical utility (Curth et al., 2024; Feuerriegel et al., 2024; Poinsot et al., 2025; Berrevoets et al., 2024).

In Causal ML, evaluation is particularly challenging due to the unobservability of counterfactual outcomes (Holland, 1986). Researchers can rely only on scarce real-world data sources such as randomized controlled trials, considered the gold standard, which are expensive, are ethically constrained, and often encompass a low amount of data (Greenland & Brumback, 2002; Tennant et al., 2021). As a result, existing benchmarks often rely on a few semi-synthetic datasets (e.g., Syntren (den Bulcke et al., 2006), ACIC2016 (Dorie et al., 2019)) or model-driven synthetic datasets generated from fitted causal mechanisms (Neal et al., 2020; Parikh et al., 2022; Athey et al., 2024; de Vassimon Manela et al., 2024). However, these datasets encode assumptions that are rarely made explicit and whose validity is difficult to generalize beyond the original study context (Poinsot et al., 2025). In parallel, many researchers define handcrafted synthetic datasets, useful for theory but fragile for empirical evaluation: a few manually chosen models can overstate performance by aligning with method-specific assumptions (Gentzel et al., 2019). Moreover, lessons from predictive machine learning show that narrow, static benchmarks can give a false sense of reliability (Geirhos et al., 2020; Herrmann et al., 2024; Freiesleben & Grote, 2023; Longjohn et al., 2024), underscoring the need for structured diversity: systematic variation of tasks under explicit, controllable assumptions.

In this work, we take a concrete step toward addressing these fundamental concerns about the field. Specifically, we introduce a synthetic benchmark generator, the CausalProfiler, that enables robust empirical evaluations grounded in transparently defined synthetic causal datasets. Central to our approach is the notion of a *Space of Interest (SoI)* (Definition 5.1), defining the domain from which causal datasets are sampled. Given an *SoI*, CausalProfiler samples Structural Causal Models (SCMs), data, and queries, and estimates the ground truth value of the queries to enable the evaluation of Causal ML methods. The assumptions are explicit, and dataset characteristics can be systematically varied through the *SoI*. Hence, CausalProfiler enables transparent, controlled, repeatable, and diverse sampling of synthetic causal datasets.

CausalProfiler shifts the focus of empirical evaluation from performance on individual datasets to trends and patterns across a well-characterized *SoI*, reframing the evaluation question from "what dataset to use" to specifying a *SoI* that defines the scope of evaluation. This enables researchers to evaluate performance across a well-defined set of conditions—on graph density, or causal mechanisms complexity, for instance—and to understand under which conditions a method succeeds or fails, helping practitioners identify methods that remain reliable when their causal assumptions are violated. Compared to conventional evaluations in the current literature, using CausalProfiler yields more robust and reliable performance estimates; it enables the systematic exploration of failure modes, generalization limits, and assumption sensitivities that remain hidden in conventional evaluations.

Although synthetic evaluation cannot replace real data, it offers the only reliable access to ground-truth causal queries, since counterfactuals are unobservable and many assumptions are unfalsifiable (Holland, 1986). CausalProfiler brings a much needed complement to real-world studies by enabling transparent, diverse, and controlled synthetic experiments to support method development.

We make two primary contributions. First, we present CausalProfiler[1] (Section 5), the first open-source benchmark generator that enables principled sampling of synthetic causal datasets with coverage guarantees, thereby promoting transparency and reproducibility in Causal ML evaluation across the three levels of causal reasoning. Secondly, we demonstrate through experiments (Section 6) how evaluation with CausalProfiler yields richer and more robust insights than the current standard practice.

2 Related Work

Evaluating Causal ML methods. Causal ML currently lacks a rigorous, systematic paradigm for empirical evaluation, whether synthetic or semi-synthetic. Semi-synthetic datasets, such as synthetic outcome datasets (Dorie et al., 2019; Shimoni et al., 2018; Hill, 2011) and model-based semi-synthetic datasets (Neal et al., 2020; Parikh et al., 2022; Athey et al., 2024; de Vassimon Manela et al., 2024), combine real covariates and simulated outcomes under assumed structural models. On the other hand, fully synthetic datasets are generated entirely from researcher-defined SCMs, allowing for greater control and access to ground truth. Yet both synthetic and semi-synthetic approaches share critical limitations. First, synthetic evaluations often lack realism, relying on overly simplistic mechanisms such as additive noise or linear functions, and frequently omitting robustness analyses (Gentzel et al., 2019; Curth et al., 2024; Poinsot et al., 2024; 2025). Such evaluations rarely reflect the complexity of real-world causal processes and are insufficient to test the limits of modern causal inference methods. Secondly, synthetic and semi-synthetic datasets are shaped by researcher-defined design choices, including the causal graph structure, the form of the outcome function, and the noise distribution. These decisions, often made implicitly, can unintentionally introduce hidden biases that favor certain methods (Curth et al., 2021; Cheng et al., 2022; Feuerriegel et al., 2024). Such assumptions are rarely documented or systematically varied, hindering reproducibility and fair method comparison (Poinsot et al., 2024; 2025). Additionally, these benchmarks are typically small in scale and narrow in scope, often covering only a limited range of causal settings. As a result, empirical evaluations raise concerns about overfitting and generalization (Gentzel et al., 2019; Berrevoets et al., 2024). For instance, it has been shown that even small changes to the data-generating process can lead to dramatic shifts in performance rankings (Curth et al., 2021). Moreover, methods are often evaluated only under the very conditions that guarantee their identifiability, offering little insight into robustness under assumption violations, as is common in real-world settings (Petersen, 2024; Hutchinson et al., 2022). In short, without broader and more transparent evaluation across diverse causal settings, the field risks drawing conclusions that do not generalize. For Causal ML to have wide impact in practice, there is a need to move beyond fixed benchmarks toward frameworks that support transparent, controlled, and diverse experimentation across well-defined spaces of causal assumptions.

[1] The code is provided in the supplementary material and will be publicly available after the review process.

Recent benchmarking efforts. Recent works have sought to address some of these problems, introducing tools to generate synthetic SCMs for causal discovery (Kalainathan et al., 2020; Gupta et al., 2023; Rudolph et al., 2023) or support query estimation from hand-specified models (Sharma & Kiciman, 2020; Textor et al., 2017; Abril-Pla et al., 2023). However, none of these frameworks support all components required for robust evaluation of causal machine learning methods. First, the causal discovery benchmarks do not compute ground truth for intervention or counterfactual queries. Further, query estimation frameworks often require manual SCM specification and do not support random sampling, diversity control, or analysis of the distribution of tasks. Even when SCMs are sampled (Rudolph et al., 2023; Xia et al., 2023), key properties (e.g., positivity) are neither reported nor constrained. Moreover, the absence of randomness in the graph structures limits generalization. In contrast, CausalProfiler integrates SCM sampling, query ground-truth computation, and coverage guarantees into a unified framework. To the best of our knowledge, this is the first benchmark generator that enables systematic exploration of how Causal ML methods behave across spaces of SCMs and queries defined by user-specified constraints.

3 Background & Notation

We use capital letters for random variables (e.g., X), lowercase for realizations (e.g., x), and boldface for vectors (e.g., $\mathbf{x}$). For a more complete background, please refer to Appendix B and Pearl (2009).

The **Pearl Causal Hierarchy (PCH)** (Pearl & Mackenzie, 2018) classifies causal reasoning into three levels: $\mathcal{L}_1$ (associational), $\mathcal{L}_2$ (interventional), and $\mathcal{L}_3$ (counterfactual). Associative questions use only observed data, whereas interventional and counterfactual questions require assumptions about the data-generating process. Importantly, lower levels are insufficient to answer higher-level questions in almost all causal models (Bareinboim et al., 2022).

The class of **Structural Causal Models (SCMs)** (Pearl, 2009) provide a representation allowing reasoning on the three levels of the PCH. An SCM is a tuple $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$, where $\mathbf{V}$ is a set of endogenous variables, $\mathbf{U}$ is a set of exogenous variables, $\boldsymbol{\mathcal{F}}$ is a set of structural equations $V_i = f_i(\boldsymbol{PA}(V_i), \mathbf{U}_{V_i})$, also called *causal mechanisms*, and $P(\mathbf{U})$ defines a distribution over the exogenous variables $\mathbf{U}$. SCMs induce a distribution $P_{\mathcal{M}}(\mathbf{V})$ over the endogenous variables $\mathbf{V}$, called the *entailed distribution*. We consider two types of endogenous variables: the observed variables, denoted $\mathbf{V}_O$, and the unobserved variables, denoted $\mathbf{V}_H$, where $\mathbf{V} = \mathbf{V}_O \cup \mathbf{V}_H$ and $\mathbf{V}_O \cap \mathbf{V}_H = \emptyset$. We represent causal relationships using the **causal graph** $\mathcal{G}$ of a SCM. This is an acyclic directed mixed graph over the endogenous variables. Directed edges $X \to Y$ encode causal dependencies via causal mechanisms where $X \in \boldsymbol{PA}(Y)$ is called a parent of Y, while bidirected edges $X \leftrightarrow Y$ indicate latent confounding due to shared exogenous causes. With SCMs one can represent intervention and counterfactual questions. An **intervention** replaces one or more structural equations to model external manipulations. A common example is a *hard intervention*, $\boldsymbol{do}(T = t)$, which fixes a variable's value, disconnecting it from its causes. This defines a new SCM and alters the induced distribution. **Counterfactual** questions build on this idea: given an observed realization called the *factual* realization, they ask what would have happened under an intervention different from the one actually taken. They are evaluated by conditioning on observed variables (abduction), modifying the SCM with the intervention (action), and predicting outcomes under the new distribution (prediction)—a process known as the *three-step procedure* (Pearl, 2009).

More generally, a **causal query** refers to a probabilistic statement about the effect of hypothetical manipulations of the data-generating process. This includes *intervention queries*, such as Average Treatment Effect (ATE), and *counterfactual queries*, such as Counterfactual Total Effect (Ctf-TE) (Plečko & Bareinboim, 2024). A query is *identifiable* if its value can be uniquely determined from data, given a set of assumptions (e.g., a causal sufficiency) (Pearl, 2009). In other words, **identifiability** refers to whether causal queries can be empirically estimated, and under what assumptions.

4 Problem Formulation

Causal inference aims to answer causal queries using data drawn from an unknown SCM. Let $\mathcal{M}^\star = (\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U}))$ denote the unknown ground truth SCM. A causal query Q (e.g., ATE) defined over $\mathcal{M}^\star$ has ground truth value $Q^\star = Q(\mathcal{M}^\star)$. As $\mathcal{M}^\star$ is unknown, causal estimators rely on causal assumptions $\mathbf{H}$ (e.g., causal sufficiency) and available data D drawn from $\mathcal{M}^\star$ to produce an estimate $\hat{Q}$ of the target quantity $Q^\star$. Definition 4.1 below formalizes the elements of a causal dataset.

Definition 4.1 (Causal Dataset)**.** A **causal dataset** is a tuple $\mathcal{D} = \{Q, Q^\star, D, \mathcal{G}^\star, \mathbf{H}^\star\}$ constructed from a known SCM $\mathcal{M}^\star = (\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U}))$ where:

- Q is a causal query defined over $\mathbf{V}$;
- $Q^\star = Q(\mathcal{M}^\star)$ is the exact value of the query Q;
- $D = \{D_k \sim P_{\mathcal{M}^\star}(\mathbf{V} \mid \boldsymbol{do}(\mathbf{V}_k) = \mathbf{v}_k)\}_{k=1}^{I}$ is a collection of samples under I interventional settings[1];
- $\mathcal{G}^\star$ is the causal graph associated with $\mathcal{M}^\star$;
- $\mathbf{H}^\star$ is the set of assumptions satisfied by $\mathcal{M}^\star$.

[1] Observational setting can be achieved by setting $I = \emptyset$

In this work, we develop a generator of causal datasets following Definition 4.1 such that, given an error metric $E(\hat{Q}, Q^\star)$, Causal ML methods can be evaluated both in the identification-consistent regime—where the assumed causal graph and assumptions used by the estimator, denoted $(\mathcal{G}, \mathbf{H})$, match the ground truth $(\mathcal{G}^\star, \mathbf{H}^\star)$—and under controlled misspecification. Here, $\mathcal{G}$ represents the graph provided to a method (e.g., a partial or misspecified graph for robustness testing), and $\mathbf{H}$ represents the assumptions that the method relies on during estimation. This setup enables systematic comparison of Causal ML methods both under ideal conditions, where identification holds, and under realistic deviations from the ground truth that test robustness.

Remark on causal discovery. Causal datasets, as defined above, can also be used for evaluating causal discovery algorithms. Each dataset already includes the ground-truth causal graph $\mathcal{G}^\star$, allowing direct assessment of discovery methods. Thus, the query Q can be left empty.

5 Sampling Causal Datasets with the CausalProfiler

To generate causal datasets, CausalProfiler relies on a parametric specification of the sampling domain, called the *Space of Interest (SoI)*. Given an *SoI*, CausalProfiler samples an SCM (Section 5.2) and generates a corresponding causal dataset (Section 5.3).

Appendices C to F contain the pseudocode for the sampling algorithms. Appendix I presents a visual overview of the sampling strategy.

5.1 Defining a Space of Interest

The central abstraction of our framework is the *Space of Interest (SoI)* (Definition 5.1), which provides a standardized way to specify synthetic causal datasets (Definition 4.1).

Definition 5.1 (Space of Interest)**.** A **Space of Interest (SoI)** is a tuple $\mathcal{S} = \{\mathbb{M}, \mathbb{Q}, \mathbb{D}\}$, where $\mathbb{M}$ is a class of SCMs, $\mathbb{Q}$ a class of causal queries, and $\mathbb{D}$ a class of data.

The mathematical definition of an *SoI* is intentionally open-ended: it specifies the classes of SCMs, queries, and data abstractly, without constraining how these components are parameterized. The concrete parameters exposed in the current CausalProfiler implementation are one instantiation of this definition[2]. As the framework evolves, the parameter list will change and expand. To help readers understand the current implementation, we summarize the main parameter groups below; full descriptions, default values and examples can be found in Appendix C.

Parameters defining the class of SCMs $\mathbb{M}$:

- *Causal structure:* number of variables, expected edge density, proportion of hidden variables, Markovian/semi-Markovian flags, optional predefined graphs.

[2] The current implementation of CausalProfiler supports only $\mathcal{L}_1$ data and ATE, CATE, and Ctf-TE queries.

- *Causal mechanisms:* mechanism family (linear, neural, tabular), discrete cardinalities, custom mechanism arguments, and noise mode (e.g., additive).
- *Noises (exogenous variables):* noise distribution, distribution arguments (e.g., mean), and number of noise regions for discrete variables.

Parameters defining the class of queries $\mathbb{Q}$: query type (e.g., ATE), number of queries per SCM, specific queries, NaN-handling options, and kernel parameters for approximating conditioning in continuous SCMs (kernel type, bandwidth, custom kernels).

Parameters defining the data class $\mathbb{D}$: number of samples generated.

5.2 Sampling Structural Causal Models

Causal Graphs. CausalProfiler first samples a directed acyclic graph over a set of endogenous variables, defining the SCM's causal structure. If specified in the *SoI*, CausalProfiler samples a subset of endogenous variables, $\mathbf{V}_H$, to be treated as unobserved and excluded from the observed dataset. To expose only the visible causal structure to the user, we apply Verma's latent projection algorithm (Verma, 1993) to the full causal graph, which produces an acyclic directed mixed graph.

Mechanisms. Given the causal graph, CausalProfiler assigns each endogenous variable a mechanism based on its parents and an exogenous noise distribution set by the *SoI*. It supports two types of mechanisms. First, **discrete mechanisms**, also called regional discrete mechanisms (see Appendix E.1 for a formal definition) which support binary and categorical treatments, are defined tabularly by associating each element of a partition of the exogenous noise with distinct parents-to-child mappings. This enables controllable stochasticity and complexity, including highly non-linear and non-invertible behavior. The *SoI* also specifies how such mechanisms are sampled (e.g., with rejection-based sampling, see Appendix E.2). Secondly, **continuous mechanisms** are defined using parametric function families—such as neural networks or linear functions—with randomly initialized parameters.

5.3 Sampling Causal Datasets

Data D. Given an SCM $\mathcal{M}^\star$ sampled from the *SoI*, we generate an observational dataset D by sampling i.i.d. data points from the entailed distribution of $\mathcal{M}^\star$ over observed variables. This involves forward-sampling from the structural equations in topological order, using the noise distributions specified for each variable and marginalizing out any latent variables.

Query Q. We first sample endogenous observable variables to serve as treatment, outcome, covariates, and factuals, depending on the query class of the *SoI*. By default, realizations are drawn from a large, separately sampled observational dataset, rather than from the theoretical variable domains. This ensures that queries are well-defined and correspond to realizable variable configurations under the SCM. To support different research goals, *SoIs* can be configured to relax this behavior (e.g., to include NaN queries) to stress-test robustness. For causal discovery, query sampling can be disabled to generate datasets more efficiently given that they already include their ground-truth graph $\mathcal{G}^\star$.

Query ground truth $Q^\star$. Each query is estimated by drawing samples from the (manipulated) ground truth SCM: interventional queries via do-operations (action and prediction), and counterfactual queries via the three-step procedure (Pearl, 2009).

Ground truth causal graph $\mathcal{G}^\star$. As presented in Section 5.2, $\mathcal{G}^\star$ is built as the latent projection of the ground-truth SCM's causal graph over the observed variables.

Ground truth causal assumptions $\mathbf{H}^\star$. Some assumptions are guaranteed directly by the SoI specification (e.g., variable types, cardinalities, presence of hidden variables). To characterize additional assumptions that are not fixed by the SoI, we provide an analysis module that can help quantify them (e.g., linearity via Pearson correlation or monotonicity). A full list of available metrics is provided in Appendix G.

Coverage guarantee. Proposition 5.1 (proof in Appendix J) shows that, with sufficiently expressive discrete mechanisms, CausalProfiler's sampling strategy can theoretically generate any causal dataset within a given *SoI*, guaranteeing $\mathcal{L}_3$-expressivity. In addition, Appendix H provides an analysis exploring the empirical distribution of the sampled datasets.

Proposition 5.1 (Coverage)**.** For a Space of Interest $\mathcal{S} = \{\mathbb{M}, \mathbb{Q}, \mathbb{D}\}$, whose class of Structural Causal Models is a class of Regional Discrete SCMs[1] with the maximum number of noise regions, denoted $\mathbb{M}_{\texttt{RD-SCM},r=R_{\max}}$, any causal dataset $\mathcal{D} = \{Q, Q^\star, D, \mathcal{G}^\star, \mathbf{H}^\star\}$ has a strictly positive probability to be generated.

$$\forall \mathcal{S} = \{\mathbb{M}, \mathbb{Q}, \mathbb{D}\} \; s.t. \; \mathbb{M} \subseteq \mathbb{M}_{\texttt{RD-SCM},r=R_{\max}}, \; P(\mathcal{D}|\mathcal{S}) > 0$$

[1] A formal definition can be found in Appendix E.1.

Benchmark Design. Taken together, these design choices reflect four key properties that are considered essential for rigorous synthetic evaluation in Causal ML (Poinsot et al., 2025): **transparency**, by making all assumptions explicit via the parametrization of the *SoI*, which serves as a declarative specification of the evaluation domain; **repeatability**, through randomized but seed-controlled sampling procedures, ensuring that SCMs and queries can be exactly reproduced across runs; **bias awareness**, supported by the coverage guarantee and the empirical distribution analysis module; and **control over experiments**, by exposing a wide range of configurable parameters in the *SoI* that allow users to tailor the causal dataset generation to their assumptions and research goals.

6 Experiments

6.1 Verification of Benchmark Correctness

To validate the soundness of our benchmark generator, we perform consistency checks across the three levels of the PCH. Using the SCM sampler and query estimator of the CausalProfiler, we test whether sampled SCMs satisfy the Markov condition, do-calculus rules, and the structural counterfactual axioms (Pearl, 2009). We use discrete SCMs to allow exhaustive enumeration of conditioning sets for statistical tests. To ensure robustness, we iterate over a *SoI* parameter grid spanning the number of variables, edge density, cardinalities, and noise regions. See Appendix K for full details and results.

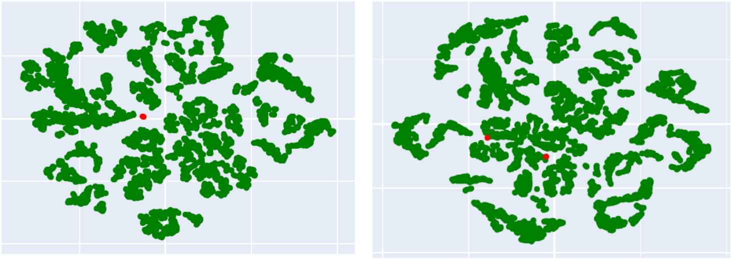
L1: Markov Property Verification. We test whether d-separations in the causal graph imply conditional independencies in the entailed observational distribution of the sampled SCMs. For each SCM, we enumerate d-separated triplets (A, B, C) and test $A \perp B \mid C$ with Pearson's χ^2 test (Pearson, 1900), filtering low-sample strata (Koehler & Larntz, 1980) and correcting for multiple tests (Benjamini & Hochberg, 1995). The Markov property holds in about 95% of the tested cases, with most violations due to finite-sample variability.

L2: Do-Calculus Verification. We test whether the three rules of do-calculus hold empirically. For each rule, we identify variable tuples satisfying its graphical preconditions. We then use the query estimator to generate two interventional datasets corresponding to the rule's left- and right-hand sides. We compare the resulting distributions with Pearson's χ^2 test, filtering low-sample strata (Koehler & Larntz, 1980) and correcting for multiple tests (Benjamini & Hochberg, 1995). About 5.5% of tests fail, mostly due to finite-sample noise.

L3: Structural Counterfactual Axiom Verification. We test whether the axioms of *composition*, *effectiveness*, and *reversibility* hold for sampled SCMs. Since the axioms involve deterministic functional relationships, we count only exact matches of the query estimator. All axioms hold exactly across our samples, confirming the estimator's consistency with structural counterfactual semantics.

6.2 Comparison to existing benchmarks

Comparison. To illustrate CausalProfiler's contribution to SCM diversity for evaluating Causal ML methods, we compare its SCMs (sampled over a *SoI* grid spanning number of variables, edge density, cardinalities, noise regions, and dataset size) with two existing benchmarks: the synthetic SCMs from the Causal Normalizing Flows (CausalNF) work (Javaloy et al., 2023) and the CANCER and EARTHQUAKE models from bnlearn (Scutari, 2019). For interpretable visualization, we apply two-dimensional t-SNE (Maaten & Hinton, 2008) to the computable metrics of the analysis module (Appendix G), with a perplexity set to 30, see Figure 1.



(a) vs. CausalNF SCMs (b) vs. CANCER and EARTHQUAKE

Figure 1: Two-dimensional t-SNE plots of CausalProfiler's SCMs (green) and established benchmarks (red), characterized by metrics from the analysis module.

Findings. Figures 1a and 1b serve complementary purposes. Figure 1a shows that the eleven CausalNF SCMs occupy a very narrow region of the metric space, whereas sampling across an SoI with the CausalProfiler yields much broader diversity. This illustrates the motivation for using a configurable generator rather than relying on a hand-crafted synthetic dataset. Figure 1b intends to illustrate two properties of CausalProfiler: (i) its ability to reproduce datasets whose characteristics resemble well-known datasets (overlap in the embedding), and (ii) the additional diversity that emerges when sampling broadly across an SoI relative to a small set of fixed models. The two bnlearn networks (out of 24 available) were selected because their characteristics (e.g., number of nodes) match the SoI used in this visualization. Further details and results are presented in Appendices H.3 and H.4.

6.3 Method Evaluation using CausalProfiler

We demonstrate the utility of our framework by evaluating several recent causal inference methods across diverse *SoIs*. Our goal is not to exhaustively benchmark each method but to showcase the kinds of structured empirical investigations CausalProfiler enables — especially on exploring robustness and violations of causal assumptions. Accordingly, we keep most *SoI* parameters fixed and vary only one (or a small subset) at a time, so observed differences can be attributed to the parameter under study rather than confounded by simultaneous changes.

For each *SoI*, we evaluate every method using five random seeds, sampling 100 SCMs per seed. Each SCM yields one training set and five queries with ground-truth values, and results are aggregated across SCMs and seeds (see Algorithm 12 in Appendix L). Experiments were run on a single Intel Core i9-14900K machine (24 cores, 32 threads, 96GB RAM), fully parallelized on CPU. Although some methods (e.g., DCM) could benefit from GPU acceleration, none was used here.

Performance is assessed by mean squared error between predicted and true query values, with mean error, standard deviation, runtime, and failure rate (due to numerical issues or exceptions) for each method and *SoI*. We compare Causal Normalizing Flows (CausalNF) (Javaloy et al., 2023), Neural Causal Models (NCM) (Xia et al., 2023), Variational Causal Graph Autoencoder (VACA) (Sánchez-Martin et al., 2022), and Diffusion-based Causal Models (DCM) (Chao et al., 2023).

Additional experiments, extended results, and *SoI* configurations are provided in Appendix L.

6.4 Experiment 1: General Evaluation across Diverse SCMs

To showcase CausalProfiler's flexibility, we evaluate ATE estimates of VACA, CausalNF, DCM, and NCM on continuous-variable SCMs across four *SoIs*: **Linear-Medium**, linear SCMs (15-20 nodes, 1000 samples); **NN-Medium**, neural SCMs with a 2-layer ReLU network (8 hidden units, 15-20 nodes, 1000 samples); **NN-Large**, larger neural SCMs (20-25 nodes, 1000 samples); and **NN-Large-LowData**, identical to NN-Large but with 50 samples. See Table 1 for results.

Findings (Linear-Medium vs. NN-Medium). In the **Linear-Medium** setting, DCM achieves the lowest average error (0.1530), indicating excellent performance, but its error standard deviation is notably high (1.5289), driven by a few extreme outliers (max error 33.98). This suggests DCM

Table 1: Performance summary of CausalNF, DCM, NCM, and VACA on the general experiments.

Space	Method	Mean Error	Std Error	Max Error	Runtime (s)	Fail Rate (%)
Linear-Medium	CausalNF	0.4625	0.8985	9.6079	13790.4	0.00
	DCM	0.1530	1.5289	33.9766	16541.2	0.00
	NCM	0.4618	0.9001	9.6134	7384.7	0.00
	VACA	0.4209	0.6195	2.3807	2734.5	53.40
NN-medium	CausalNF	0.0160	0.0107	0.1209	10732.7	0.00
	DCM	0.0276	0.0114	0.0746	15894.4	0.00
	NCM	0.0111	0.0121	0.1484	7322.8	0.00
	VACA	0.0090	0.0077	0.0479	5759.6	5.00
NN-Large	CausalNF	0.0159	0.0105	0.1535	15114.8	0.00
	DCM	0.0267	0.0100	0.0739	19166.2	0.00
	NCM	0.0101	0.0103	0.1161	9450.6	0.00
	VACA	0.0090	0.0094	0.0535	5690.8	11.60
NN-Large-LowData	CausalNF	0.0359	0.0146	0.1712	22138.2	0.00
	DCM	0.0777	0.0445	0.3701	2412.1	0.00
	NCM	0.0097	0.0107	0.1263	404.7	0.00
	VACA	0.0103	0.0134	0.1043	5217.4	0.00

is effective for most queries but can produce large errors in rare cases—potentially problematic in safety-critical applications matching this *SoI*. VACA performs competitively with lower max error and faster runtime, but suffers a high failure rate (53.4%) due to NaNs. In the **NN-Medium** setting, where the causal mechanisms are small neural networks, DCM's advantage disappears. VACA emerges as the best performer, with the lowest error mean (0.0090) and standard deviation (0.0077), while reducing its failure rate to 5%. Interestingly, DCM becomes the weakest performer in this setting, showing that method rankings are highly sensitive to the underlying functional form of the mechanisms. This underscores the need for practitioners to evaluate methods within the *SoI* most relevant to their application. Lastly, NN SCMs surprisingly yield lower errors than linear ones. A plausible explanation is an inductive-bias match with the evaluated neural methods and the tendency of small randomly initialized NNs to produce relatively smooth, low-frequency functions that are easier to estimate from finite data (Rahaman et al., 2019).

Findings (NN-Large vs. NN-Large-LowData). In this comparison, we investigate the effect of reducing data availability. Comparing **NN-Large** (1000 samples) to **NN-Large-LowData** (50 samples), DCM is strongly affected: its error nearly triples (from 0.0267 to 0.0777) and its IQR expands noticeably. CausalNF also shows greater sensitivity to low-data regimes. In contrast, both VACA and NCM maintain stable performance, with nearly unchanged mean and standard deviation. Notably, VACA achieves a 0% failure rate, with unexpectedly strong robustness under limited data.

Insights. While not intended as a comprehensive benchmark, these experiments illustrate the types of insights enabled by our framework. Across the selected *SoIs*, DCM performs well on average but can produce large outlier errors or become less stable in low-data settings. Conversely, VACA shows promising generalization even with limited data, though it occasionally fails on certain SCMs. These findings are specific to the explored *SoIs* and should not be taken as general conclusions. Rather, they show how our framework enables structured, *SoI*-specific evaluations, helping practitioners assess which methods may be more suitable for their own modeling context.

6.5 Experiment 2: Counterfactual Estimation on Discrete SCMs

This experiment evaluates counterfactual estimation on discrete-variable SCMs as a robustness check, testing CausalNF and DCM—originally designed for continuous settings—motivated by prior work showing that CausalNF can approximate discrete distributions (Javaloy et al., 2023; de Vassimon Manela et al., 2024). We consider three discrete *SoIs*: **Disc-C2-Reject**, with 10-15 nodes, binary variables, and rejection-based mechanism sampling; **Disc-C4-Unbias**, with the same graph size but 4-category variables and unbiased random mechanism sampling; and **Disc-L-C2-Unbias**, with larger graphs (20-30 nodes), binary variables, and unbiased random mechanism sampling (Table 2).

Table 2: Performance summary of CausalNF and DCM on the discrete experiments.

Space	Method	Mean Error	Std Error	Max Error	Runtime	Fail Rate
Disc-C2-Reject	CausalNF	0.0415	0.1116	0.6240	212.8 s	08.08 %
	DCM	0.0424	0.1123	0.6240	4406.2 s	04.28 %
Disc-C4-Unbias	CausalNF	0.0431	0.1270	0.7071	190.7 s	40.68 %
	DCM	0.0411	0.1199	0.7071	3839.4 s	22.60 %
Disc-L-C2-Unbias	CausalNF	NaN	NaN	NaN	0.0 s	100.00 %
	DCM	0.0183	0.0814	0.5000	8192.7 s	11.32 %

Findings. On **Disc-C2-Reject**, both CausalNF and DCM perform well and comparably, with low error means (∼0.04) and low failure rates (8% for CausalNF, 4% for DCM). This suggests that both methods can produce reliable estimates even outside their original assumptions when the functional mechanisms are simple and binary. However, when moving to **Disc-C4-Unbias**, where variables have 4 categories and mechanisms are sampled with unbiased random sampling, the failure rates increase significantly, especially for CausalNF, which fails on over 40% of SCMs (typically with NaN errors). This highlights the sensitivity of some methods to mechanism sampling or variable cardinality, even when mean errors remain similar. To further probe robustness, we scale the graph size in **Disc-L-C2-Unbias** while reverting to binary variables. CausalNF fails on all runs, returning NaNs. DCM has an 11% failure rate, indicating greater resilience in this setting.

Insights. These results underscore the utility of our framework in systematically stress-testing methods beyond their nominal design assumptions. While CausalNF is not built for discrete data, prior work suggested it could work in practice. Our framework can help clarify *when* and *how* it fails: certain function classes and discrete configurations are more likely to cause divergence or failure. DCM appears more robust across these tests, though not immune. Importantly, this evaluation is not meant as a definitive comparison, but as a demonstration of how failure cases can be surfaced and studied in a principled way using the CausalProfiler.

7 Limitations and Future Work

We note that any open-source framework such as CausalProfiler is never a completely finished project, but rather continuously evolving to meet community needs, with new features added as the field advances through contributions to the repository.

Diversify Spaces of Interest. Several directions remain open for extending the supported *SoIs* in CausalProfiler, such as support for scaled and mixed-variable SCMs, sampling interventional training data, more realistic data-generating scenarios (e.g., selection bias or measurement noise), and extensions beyond tabular data to time-series, images, or text. Another promising direction is automating the exploration of *SoIs*—for example, searching for assumption regimes that reveal a method's failure modes—to reduce reliance on manual specification.

Causal Datasets Distribution. While the coverage proposition (Proposition 5.1) guarantees that any causal dataset has a positive probability of being sampled within a given *SoI* with sufficiently expressive discrete mechanisms, it does not characterize the distribution of generated datasets. As presented in Appendix H, certain classes of SCMs remain unlikely to be sampled unless explicitly specified in the *SoI* (e.g., linear SCMs). Hence, when aggregating results, users should bear in mind that causal datasets are not distributed uniformly to avoid misleading interpretations. We strongly recommend users to use the analysis module, presented in Appendix G, to identify the underrepresented attributes, as these vary from one *SoI* specification to another.
Reducing distributional bias is an important future research direction. Achieving a perfectly balanced distribution over all metrics is inherently impossible. For instance, uniform sampling over discrete mechanism functions biases toward non-bijective ones, since bijections are not dense in the function space. Future work may enable finer control over dataset distributions and underrepresented attributes, depending on the guarantees one wishes to enforce. One promising avenue is stratified sampling, which would provide weighted coverage of selected attributes. Currently, controllable *SoI* parameters (e.g., number of nodes) are sampled uniformly, but emergent attributes follow skewed distributions

induced by generation. For controllable *SoI* parameters, stratification could be achieved constructively via weighted sampling over groups of *SoIs*. For emergent properties, approximate stratification may require rejection sampling or, more efficiently, new sampling algorithms that enforce global constraints during generation.

Bridging the simulation-to-real gap. While synthetic evaluation is indispensable (Poinsot et al., 2025), it is insufficient to fully assess method capabilities, as results may not transfer to real-world settings. In CausalProfiler, alignment with real domains currently relies on manually specified *SoIs*, guided by domain expertise or empirical features. A key direction for future work is to develop methods that automatically map real data to sets of *SoIs*, enabling principled semi-synthetic evaluation pipelines where *SoIs* are shaped by empirical evidence rather than fixed assumptions. However, mapping from observational data to *SoIs* is a fundamentally underconstrained problem, and any such inference must be handled with care, given the challenges around identifiability and inductive bias.

8 Conclusion

This work introduces CausalProfiler, a synthetic causal dataset generator for evaluating Causal ML methods across the three levels of the Pearl Causal Hierarchy. At its core is the notion of a *Space of Interest*, which replaces the ad hoc choice of fixed evaluation datasets with a principled specification of the entire evaluation scope, i.e., classes of causal models, queries and data. This shift enables transparent, repeatable, and assumption-aware assessments under diverse causal conditions. After demonstrating that the causal datasets generated by CausalProfiler are correct and can be similar to existing benchmarks while also being considerably more diverse, we show that the performance of state-of-the-art Causal ML methods varies substantially across different *Spaces of Interest*, underscoring the importance of rigorous, distribution-level evaluation. CausalProfiler is not intended to replace real-data studies or targeted evaluations, but to complement them. By enabling systematic exploration, it helps uncover failure modes, expose robustness to violated assumptions, and highlight unexpected strengths that may motivate new research directions. In this way, CausalProfiler marks a first step toward a more complete evaluation ecosystem for Causal ML.

Reproducibility statement

We have taken extensive measures to ensure the reproducibility of our results. The paper specifies fully the steps required to reproduce our experiments, with pseudocode for all algorithms provided in the appendices. All experimental configurations are also documented in the appendices. An anonymized zip archive containing the full code and reproduction instructions is included in the supplementary materials. The codebase reflects the exact setup used in the reported experiments. Upon acceptance, we will publicly release the codebase on GitHub. We note that no external datasets are required to reproduce the experiments. We also specify the hardware used and report runtime metrics, making computational requirements transparent.

Ethics statement

This work introduces CausalProfiler, a synthetic benchmark generator for evaluating Causal ML methods. As a methodological tool rather than an application-facing system, it does not directly raise societal impact concerns to the best of our knowledge. Furthermore, to prevent naive use of CausalProfiler, this work transparently outlines its guarantees and limitations. We also remind readers of the simulation-to-real gap inherent to any synthetic system. To mitigate the risk of inadvertent misuse of CausalProfiler, it is emphasized that evaluation results should not be aggregated and interpreted naively without exploring the distribution of the generated causal datasets.

Finally, we do not release pretrained models or real-world datasets. We provide code that generates fully synthetic data, thereby avoiding issues related to privacy, fairness, and security. The paper involves no human subjects, crowdsourcing, or sensitive data.

REFERENCES

Oriol Abril-Pla, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J Fonnesbeck, Maxim Kochurov, Ravin Kumar, Junpeng Lao, Christian C Luhmann, Osvaldo A Martin, et al. PyMC: a modern, and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9, 2023.

Alejandro Almodóvar, Adrián Javaloy, Juan Parras, Santiago Zazo, and Isabel Valera. DeCaFlow: A deconfounding causal generative model. *arXiv:2503.15114*, 2025.

Susan Athey, Guido W Imbens, Jonas Metzger, and Evan Munro. Using wasserstein generative adversarial networks for the design of monte carlo simulations. *Journal of Econometrics*, 240(2): 105076, 2024.

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, pp. 507–556. 2022.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57(1):289–300, 1995.

Jeroen Berrevoets, Krzysztof Kacprzyk, Zhaozhi Qian, Mihaela van der Schaar, et al. Causal deep learning: Encouraging impact on real-world problems through causality. *Foundations and Trends in Signal Processing*, 18(3):200–309, 2024.

Patrick Chao, Patrick Blöbaum, and Shiva Prasad Kasiviswanathan. Interventional and counterfactual inference with diffusion models. *arXiv:2302.00860*, 2023.

Lu Cheng, Ruocheng Guo, Raha Moraffah, Paras Sheth, K. Selçuk Candan, and Huan Liu. Evaluation methods and measures for causal learning algorithms. *IEEE Transactions on Artificial Intelligence*, 3(6):924–943, 2022.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

Alicia Curth, David Svensson, Jim Weatherall, and Mihaela van der Schaar. Really doing great at estimating CATE? a critical look at ML benchmarking practices in treatment effect estimation. In *Advances in Neural Information Processing Systems*, 2021.

Alicia Curth, Richard W. Peck, Eoin McKinney, James Weatherall, and Mihaela van der Schaar. Using machine learning to individualize treatment effect estimation: Challenges and opportunities. *Clinical Pharmacology & Therapeutics*, 115(4):710–719, 2024.

Daniel de Vassimon Manela, Laura Battaglia, and Robin J. Evans. Marginal causal flows for validation and inference. In *Advances in Neural Information Processing Systems*, 2024.

Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7(1), 2006.

Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34:43–68, 2019.

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.

Timo Freiesleben and Thomas Grote. Beyond generalization: A theory of robustness in machine learning. *Synthese*, 202(4):109, 2023.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Amanda Gentzel, Dan Garant, and David Jensen. The case for evaluating causal models using interventional measures and empirical data. *Advances in Neural Information Processing Systems*, 2019.

Sander Greenland and Babette Brumback. An overview of relations among causal modelling methods. *International Journal of Epidemiology*, 31(5):1030–1037, 2002.

Shantanu Gupta, Cheng Zhang, and Agrin Hilmkil. Learned causal method prediction. *arXiv:2311.03989*, 2023.

Moritz Herrmann, F. Julian D. Lange, Katharina Eggensperger, Giuseppe Casalicchio, Marcel Wever, Matthias Feurer, David Rügamer, Eyke Hüllermeier, Anne-Laure Boulesteix, and Bernd Bischl. Position: Why we must rethink empirical research in machine learning. In *International Conference on Machine Learning*, 2024.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81 (396):945–960, 1986.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2): 251–257, 1991.

Ben Hutchinson, Negar Rostamzadeh, Christina Greer, Katherine Heller, and Vinodkumar Prabhakaran. Evaluation gaps in machine learning practice. In *ACM Conference on Fairness, Accountability, and Transparency*, 2022.

Adrián Javaloy, Pablo Sanchez-Martin, and Isabel Valera. Causal normalizing flows: from theory to practice. In *Advances in Neural Information Processing Systems*, 2023.

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv:2206.15475*, 2022.

Diviyan Kalainathan, Olivier Goudet, and Ritik Dutta. Causal discovery toolbox: Uncovering causal relationships in python. *Journal of Machine Learning Research*, 21(37):1–5, 2020.

Kenneth J. Koehler and Kinley Larntz. An empirical investigation of goodness-of-fit statistics for sparse multinomials. *Journal of the American Statistical Association*, 75(370):336–344, 1980.

Rachel Longjohn, Markelle Kelly, Sameer Singh, and Padhraic Smyth. Benchmark data repositories for better benchmarking. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in Neural Information Processing Systems*, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. Realcause: Realistic causal inference benchmarking. *arXiv:2011.15007*, 2020.

Harsh Parikh, Carlos Varjao, Louise Xu, and Eric Tchetgen Tchetgen. Validating causal inference methods. In *International Conference on Machine Learning*, 2022.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

Anne Helby Petersen. Are you doing better than random guessing? a call for using negative controls when evaluating causal discovery algorithms. *arXiv:2412.10039*, 2024.

Drago Plečko and Elias Bareinboim. Causal fairness analysis. *Foundations and Trends in Machine Learning*, 17(3):1–238, 2024.

Audrey Poinsot, Alessandro Leite, Nicolas Chesneau, Michèle Sébag, and Marc Schoenauer. Learning structural causal models through deep generative models: Methods, guarantees, and challenges. In *International Joint Conference on Artificial Intelligence*, 2024.

Audrey Poinsot, Panayiotis Panayiotou, Alessandro Leite, Nicolas CHESNEAU, Özgür Şimşek, and Marc Schoenauer. Position: Causal machine learning requires rigorous synthetic experiments for broader adoption. In *International Conference on Machine Learning, Position Paper Track*, 2025.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, 2019.

Kara E Rudolph, Nicholas T Williams, Caleb H Miles, Joseph Antonelli, and Ivan Diaz. All models are wrong, but which are useful? comparing parametric and nonparametric estimation of causal effects in finite samples. *Journal of Causal Inference*, 11(1), 2023.

Marco Scutari. Package 'bnlearn'. bayesian network structure learning, parameter learning and inference, r package version, 2019. URL `https://www.bnlearn.com/`.

Rajen D. Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3), 2020.

Amit Sharma and Emre Kiciman. DoWhy: An end-to-end library for causal inference. *arXiv:2011.04216*, 2020.

Yishai Shimoni, Chen Yanover, Ehud Karavani, and Yaara Goldschmnidt. Benchmarking framework for performance-evaluation of causal inference analysis. *arXiv:1802.05046*, 2018.

Beate Sick and Oliver Dürr. Interpretable neural causal models with tram-dags. In *Conference on Causal Learning and Reasoning*, 2025.

Pablo Sánchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Designing variational graph autoencoders for causal queries. *AAAI Conference on Artificial Intelligence*, 2022.

Peter WG Tennant, Eleanor J Murray, Kellyn F Arnold, Laurie Berrie, Matthew P Fox, Sarah C Gadd, Wendy J Harrison, Claire Keeble, Lynsie R Ranker, Johannes Textor, et al. Use of directed acyclic graphs (DAGs) to identify confounders in applied health research: review and recommendations. *International Journal of Epidemiology*, 50(2):620–632, 2021.

Johannes Textor, Benito van der Zander, Mark S Gilthorpe, Maciej Liśkiewicz, and George TH Ellison. Robust causal inference using directed acyclic graphs: the r package 'dagitty'. *International Journal of Epidemiology*, 45(6):1887–1894, 2017.

Jin Tian and Judea Pearl. A general identification condition for causal effects. In *AAAI Conference on Artificial Intelligence*, 2002.

Thomas Sadanand Verma. Graphical aspects of causal models. In *UCLA Cognitive Systems Laboratory, Technical Report (R-191)*, 1993.

Kevin Xia, Yushu Pan, and Elias Bareinboim. Neural causal models for counterfactual identification and estimation. In *International Conference on Learning Representations*, 2023.

Junzhe Zhang and Elias Bareinboim. Fairness in decision-making – the causal explanation formula. *AAAI Conference on Artificial Intelligence*, 2018.

Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In *International Conference on Machine Learning*, 2022.

Qingyang Zhou, Kangjie Lu, and Meng Xu. Causally consistent normalizing flow. *AAAI Conference on Artificial Intelligence*, 2025.

A USE OF LARGE LANGUAGE MODELS (LLMS)

We used LLMs, specifically ChatGPT, as a writing assistant. The model was used only to help with language-related aspects of the paper, including:

- Rephrasing existing content without changing its meaning
- Improving clarity and flow
- Identifying issues such as unclear points, unintended tones, or awkward phrasing.

All scientific contributions originate from the authors, who take full responsibility for the paper.

B ADDITIONAL DEFINITIONS & NOTATIONS

Definition B.1 (Semi-Markovian and Markovian SCMs)**.** An SCM is said to be **semi-Markovian** (Pearl, 2009) if its set of structural equations is acyclic, meaning there exists an ordering of the equations such that for any two functions $f_i, f_j \in \boldsymbol{\mathcal{F}}$, if $f_i < f_j$, then $V_j \notin \boldsymbol{PA}(V_i)$. This condition ensures that the causal dependencies among endogenous variables form a Directed Acyclic Graph.
An SCM is **Markovian** (Pearl, 2009) if the exogenous variables influencing different endogenous variables are mutually independent. Formally, for all distinct $V_i, V_j \in \mathbf{V}$, we have $\mathbf{U}_{V_i} \perp\!\!\!\perp \mathbf{U}_{V_j}$. This implies the absence of latent confounding, allowing the model to be fully described by a DAG with independent noise terms.

Definition B.2 (Causal Graph of a Semi-Markovian SCM)**.** The causal graph of a Semi-Markovian (Bareinboim et al., 2022) SCM is an acyclic directed mixed graph with:

- Directed edge $V_i \to V_j$ if $V_i \in \boldsymbol{PA}(V_j)$
- Bi-directed edge $V_i \leftrightarrow V_j$ if $\mathbf{U}_{V_i} \not\perp\!\!\!\perp \mathbf{U}_{V_j}$

B.1 INTERVENTIONAL QUANTITIES ($\mathcal{L}_2$)

Average Treatment Effect (ATE):

$$\text{ATE}_{T\to Y} = \mathbb{E}[Y|\boldsymbol{do}(T=1)] - \mathbb{E}[Y|\boldsymbol{do}(T=0)]$$

Conditional Average Treatment Effect (CATE):

$$\text{CATE}_{T\to Y}(\mathbf{x}) = \mathbb{E}[Y|\boldsymbol{do}(T=1), \mathbf{X}=\mathbf{x}] - \mathbb{E}[Y|\boldsymbol{do}(T=0), \mathbf{X}=\mathbf{x}]$$

B.2 COUNTERFACTUAL QUANTITIES ($\mathcal{L}_3$)

A counterfactual query such as $P(Y_{\boldsymbol{do}(T=t)}|\mathbf{V}_F = \mathbf{v}_F)$ is computed by abduction (conditioning on factual data), action (intervening), and prediction (computing the outcome) (Pearl, 2009).

Counterfactual Total Effect (Ctf-TE):

$$\text{Ctf-TE}_{T\to Y}(y, t, c, \mathbf{v}_F) = P(y_{\boldsymbol{do}(T=t)}|\mathbf{V}_F = \mathbf{v}_F) - P(y_{\boldsymbol{do}(T=c)}|\mathbf{V}_F = \mathbf{v}_F)$$

Originally, (Zhang & Bareinboim, 2018) defined counterfactual direct, indirect, and spurious effects by conditioning on the factual realization of one variable. Later, (Plečko & Bareinboim, 2024) generalized this to allow the factual evidence to be any subset $\mathbf{V}_F$ of endogenous variables, enabling more granular and flexible counterfactual analyses.

C Space of Interest

C.1 Configurable Parameters of a Space of Interest

Each Space of Interest is defined by a set of parameters that control the *SCM space*, the causal queries of interest (*Query space*), and the dataset used for estimation (*Data space*). Table 3 provides an overview of all configurable parameters in a Space of Interest instance, along with their default values. Some parameters are only relevant under specific conditions—for instance, kernel parameters are used only with continuous variables (e.g., when evaluating conditional expectations), function sampling strategies apply exclusively to discrete mechanisms, noise regions apply only for discrete SCMs, and noise mode is ignored for tabular mechanisms (noise is already embedded in the table). Note that one can use symbolic expressions involving `N` (the number of nodes) and `V` (the cardinality of a variable) to define parameters that depend on sampled values. For example, the expected number of edges can be set as `0.5 * N`, or the number of noise regions in a discrete SCM can be set to `V`.

Table 3: Parameters defining a Space of Interest instance and their default values. The double lines in the table conceptually separate the SCM space, Query space, and Data space.

Category	**Parameter**	**Default Value**
SCM structure	Number of endogenous variables	[5, 15]
	Variable dimensionality	[1, 1]
	Expected number of edges (required)	—
	Proportion of hidden variables	0.0
	Markovian boolean flag	True
	Semi-Markovian boolean flag	False
	Predefined causal graph	—
Mechanisms	Mechanism family (e.g., Linear, NN, Tabular)	Linear
	Mechanism arguments (used to define custom NN/tabular mechanisms)	—
	Endogenous variable cardinality (for discrete variables only)	2
	Variable type	Continuous
	Discrete function sampling (for discrete variables only)	Sample Rejection
	Noise mode	Additive
Noise	Noise distribution	Uniform
	Noise distribution arguments	[-1, 1]
	Number of noise regions (for discrete variables only)	N
Query	Number of queries per sample	1
	Query type	ATE
	Specific query (overrides random query sampling)	—
	Whether to allow queries that evaluate to NaN	False
	Whether to disable query sampling (e.g., for causal discovery)	False
Kernel	Kernel type	Gaussian
	Kernel bandwidth	0.1
	Custom kernel function	—
Data	Number of samples in the set of observed data	1000

Parameter descriptions and typical values

We briefly summarize the role of each parameter and the typical values it can take. Unless otherwise stated, scalar parameters may be given as fixed values, ranges (e.g., `(a, b)`), or simple expressions in `N` (number of nodes) and `V` (variable cardinality).

SCM structure.

- **Number of endogenous variables.** Range for the number of nodes in each sampled graph (e.g., `[5, 15]`). A value is drawn from this range for each SCM.
- **Variable dimensionality.** Range for the dimensionality of each variable (typically `[1, 1]` in our experiments, but higher-dimensional variables are supported).

- **Expected number of edges (required).** Controls graph density via the expected total number of edges. Can be a fixed integer, a range, or an expression such as `0.5 * N` or `log(N)`.
- **Proportion of hidden variables.** Fraction of endogenous variables that are hidden in the returned graph, data, and queries (a float in $[0, 1]$); `0.0` means no hidden variables.
- **Markovian / Semi-Markovian flags.** Boolean flags specifying whether the SCM is Markovian (no latent confounders) or semi-Markovian (allows latent confounders). These flags are mutually exclusive.
- **Predefined causal graph.** Fixed graph to be used for all SCMs. If unset, graphs are sampled according to the structural parameters above.

Mechanisms.

- **Mechanism family.** Choice of functional form for the structural mechanisms (e.g., linear, neural network, tabular), given by an enum.
- **Mechanism arguments.** Optional hyperparameters passed to the chosen mechanism family (e.g., hidden-layer sizes for neural networks, or explicit tables for tabular mechanisms).
- **Endogenous variable cardinality.** Cardinality (or range of cardinalities) for discrete variables (e.g., `2` or `(2, 4)`). Ignored when `variable type` is continuous.
- **Variable type.** Whether variables are continuous or discrete. This determines which mechanism and noise options are applicable.
- **Discrete function sampling.** Strategy for sampling discrete mechanisms (e.g., sample-rejection, enumeration, or random sampling). More information about these strategies in Appendix appendix E.2.
- **Noise mode.** How noise enters the structural equations (e.g., additive or multiplicative). This is ignored for tabular mechanisms, where stochasticity is already encoded in the table.

Noise.

- **Noise distribution.** Distribution from which exogenous noise variables are drawn (e.g., uniform).
- **Noise distribution arguments.** Parameters of the noise distribution (e.g., `[-1, 1]` for a uniform distribution on $[-1, 1]$).
- **Number of noise regions.** Used to specify the number of noise regions in mechanisms. The more the number of noise regions, the more random / stochastic the mechanism is. Setting to `1` yields deterministic mechanisms.

Query space.

- **Number of queries per sample.** Number of causal queries generated for each SCM.
- **Query type.** Type of causal query to sample (e.g., ATE, CATE, or Ctf-TE), specified via an enum.
- **Specific query.** Optional string specifying a fixed query to evaluate. If provided, this overrides random query sampling.
- **Allow NaN queries.** Whether to include queries whose numerical estimates evaluate to NaN (e.g., due to lack of support). By default, such queries are excluded.
- **Disable query sampling.** If set to `True`, no queries are sampled or evaluated (useful for causal discovery tasks where only data and graphs are needed).

Kernel weighting (continuous conditioning only).

- **Kernel type.** Choice of kernel used to approximate conditioning for continuous variables (e.g., Gaussian, epsilon).

- **Kernel bandwidth.** Bandwidth parameter controlling the smoothness of the kernel weighting (and acting as an epsilon threshold when using an epsilon kernel).
- **Custom kernel function.** Optional user-specified (in Python) kernel function.

Data space.

- **Number of samples in the set of observed data.** Size of the dataset generated for each SCM.

C.2 Guidelines for defining a Space of Interest

This section presents general guidelines on how researchers and practitioners could define the Spaces of interest depending on the analysis they want to carry out.

Testing a new method without a predefined application. Begin by evaluating the method in settings where its assumptions hold. If an assumption can be enforced directly through SoI parameters (e.g., no hidden variables, linear mechanisms), fix those parameters accordingly. Otherwise, sample from a broader SoI and use the assumption-analysis module to retain only SCMs satisfying the assumption.

Next, assess robustness by gradually introducing assumption violations. Assumptions that can be varied explicitly (e.g., increasing the proportion of hidden variables) should be adjusted directly through the SoI. For assumptions that cannot be controlled parametrically, sample broadly and filter using the assumption-analysis module. The module can also quantify the *degree* of violation, enabling sensitivity analyses.

Comparing multiple methods without a predefined application. Follow the same two-stage structure. First evaluate all methods in SoIs where their assumptions are jointly satisfied (verification). Then introduce controlled assumption violations to study comparative robustness. This yields a principled, assumption-aware comparison rather than a collection of isolated tests.[3]

Evaluating methods for a specific application or use case. Fix all SoI parameters that are known from domain expertise (e.g., variable types, expected graph sparsity, presence of latent confounding). Then vary the remaining uncertain parameters to span the plausible causal conditions for the application. This produces a well-defined set of SCMs consistent with the use case, enabling structured, domain-grounded evaluation. We now illustrate this with a concrete example.

C.2.1 Example 1: Price elasticity

A company's analytical marketing team wishes to estimate the price elasticity of one of its products. The team has access to three years of sales data and price history, as well as competitors' prices and inflation trends. The team knows that calculating price elasticity involves determining the ATE of price on sales for various price values. In addition, the team has also constructed a causal graph corresponding to the decision-making process used to set the product price and its effect on sales, as shown in Figure 2. In fact, the price is set based on competitors' prices, inflation (because production costs are highly correlated with it), and a set of other factors for which they do not have historical data to include in the modeling. These factors are also assumed to be used by competitors.

The team wants to calculate discrete elasticity using non-parametric DoubleML methods (Chernozhukov et al., 2018). However, they do not know which method is more suitable for their setting.Hence, the team decides to use CausalProfiler to perform their own comparison and define the following set of SoI parameters:

- **SCM structure parameters**: The team decides to use the option of using a predefined causal graph corresponding to the one in Figure 2.
- **Mechanisms parameters**:
  - Variable type: Continuous, as all the variables in this use case are continuous.
  - Mechanism family: Neural Networks, as no assumption is made about the functional form of the causal mechanisms.

[3] Our experiments in Sections 6.4 and 6.5 illustrate the types of analyses enabled by CausalProfiler but are not intended as full comparative evaluations.

  - Given the previously made choices, the other parameters have no influence on the generation.
- **Noise parameters**:
  - Noise distribution: all the available noise distributions are considered, as no assumption is taken about the form of the distribution.
  - Noise distribution arguments are the default ones, as neither the mean nor the scale of the noise should drastically affect the generation, as we are using randomly initialized Neural Networks as causal mechanisms.
- **Query parameters**:
  - The team decides to define a set of specific queries rather than randomly sampling them, as they are interested in a single pair of treatment and outcome variables.
- **Kernel parameters**: Default Kernel parameters.
- **Data parameter**: The number of samples is varied between the number of observations they have for the past year and for the three past years. Indeed, the team would ideally measure the price elasticity over the past year to have the most recent measurement, but is also ready to include older data (maximum three years old) to provide more observations to the model if it drastically changes its accuracy.

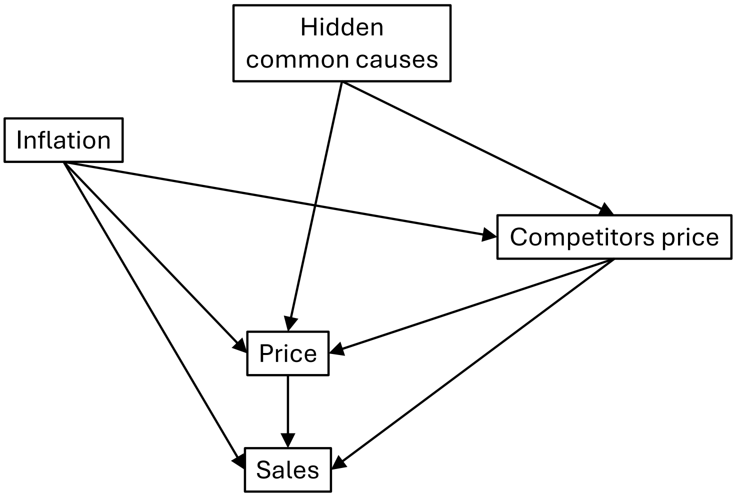


Figure 2: Causal graph of the price elasticity example.

D Causal Graph Sampling

We first generate a random Directed Acyclic Graph (DAG) that specifies causal relations between variables. This structure is then extended by designating a subset of variables as hidden/unobserved, enabling the creation of both Markovian and semi-Markovian SCMs depending on the *SoI* spec. We separate these two steps in separate algorithms for clarity (Algorithm 2 uses Algorithm 1).

First, Algorithm 1 samples a DAG over a unique type of variables, not yet distinguishing between observable and unobservable variables. To do so, the list of nodes is defined as a list of integers imposed to be the topological order of the DAG (line 1). Then, for each node (line 4), its number of parents is sampled from a Binomial law of parameters $i - 1$ and p_{edge} with i the rank of the node in the topological order (line 5). The actual parents are sampled from the set of nodes having a smaller topological rank (line 6) which guarantees that the generated graph is a DAG.

Second, from the generated DAG, Algorithm 2 simply creates the two sets of observables and unobservable variables by sampling $p_h.|\mathbf{V}|$ unobservable node among the total set of nodes (line 3).

Algorithm 1 Generate a Random DAG with Expected Degree

```
Inputs: number of nodes $N$, expected degree $d$
$V \leftarrow \{1, \ldots, N\}$
$E \leftarrow \{\}$
$p_{edge} \leftarrow \frac{2d}{N-1}$
for $i \in [1, N]$ do
    $N_{\boldsymbol{PA}(i)} \sim B(i-1, p_{edge})$
    $\boldsymbol{PA}(i) \leftarrow N_{\boldsymbol{PA}(i)}$ nodes sampled without replacement from $V$
    $E \leftarrow E \cup \{j \to i \mid j \in \boldsymbol{PA}(i)\}$
end for
Output: $\mathcal{G} = \{V, E\}$
```

Algorithm 2 Generate a DAG with Observed and Hidden Variables

```
Inputs: number of nodes $N$, expected degree $d$, proportion of hidden variables $p_h$
$\mathcal{G} = (V, E) \leftarrow DAG\_sampling(N, d)$ (see Algorithm 1)
$N_h \sim B(N, p_h)$
$V_h \leftarrow N_h$ nodes sampled without replacement from $V$
$V_o \leftarrow V \backslash V_h$
Output: $\mathcal{G} = \{V = V_oV_h, E\}$
```

Because some variables in the DAG are unobserved, we expose only the observed structure to the user in the form of an acyclic directed mixed graph. To obtain this, we apply Verma's latent projection algorithm to the causal graph of each sampled regional discrete SCM (see Algorithm 3). If a method requires the true SCM, including the hidden confounders, that can be accessed as well.

Algorithm 3 Projection Algorithm (Verma, 1993)

```
Input: an acyclic directed mixed graph $\mathcal{G} = \{\mathbf{V_O}, \mathbf{V_H}, \mathbf{E}\}$, with $\mathbf{V_O}$ the set of observed variables, $\mathbf{V_H}$ the set of hidden variables and $\mathbf{E}$ the mixed edges
$\mathbf{E}' \leftarrow \{\}$
for $A, B \in \mathbf{V_O}$ do
    if there is a directed path $A \to \ldots \to B$ in $\mathcal{G}$ with all intermediate nodes belonging to $\mathbf{V_H}$ then
        $\mathbf{E}' \leftarrow \mathbf{E}' \cup \{A \to B\}$
    end if
    if there is a collider-free path $A \leftarrow \ldots \to B$ in $\mathcal{G}$ with all intermediate nodes belonging to $\mathbf{V_H}$ then
        $\mathbf{E}' \leftarrow \mathbf{E}' \cup \{A \leftrightarrow B\}$
    end if
end for
$\mathbf{G}' \leftarrow \{\mathbf{V_O}, E'\}$
Output: $\mathbf{G}'$ the latent projection of $\mathbf{G}$ over $\mathbf{V_O}$
```

E Sampling Discrete SCMs

E.1 Regional Discrete SCMs

Regarding discrete SCMs, we sample discrete Markovian SCMs which we refer to as **Regional discrete SCMs** as presented in definition E.1 below.

Definition E.1. **Regional discrete SCM**

A **regional discrete SCM** is a markovian SCM $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$ where:

- $\mathbf{V} = \{V_1, ..., V_d\}$ the set of finite discrete endogenous variables is divided into two sets $\mathbf{V}_o$ and $\mathbf{V}_h$ respectively representing the set of observed and hidden variables such that $\mathbf{V} = \mathbf{V}_o \cup \mathbf{V}_h$ and $\mathbf{V}_o \cap \mathbf{V}_h = \emptyset$
- $\mathbf{U} = \{U_1, ..., U_d\}$ the set of mutually independent continuous exogenous variables is such that $\forall i \in [1, d],\ U_{V_i} = U_i$
- $\boldsymbol{\mathcal{F}}$ the structural equations are regional discrete mechanisms as defined in Definition E.2

The class of regional discrete SCMs is denoted $\mathbb{M}_{\texttt{RD-SCM}}$.

Definition E.2. **Regional discrete mechanism**

Given $\mathbf{I}_V = \{I_V^r\}_{r\in[1,R]}$ a partition of R parts of Ω_{U_V} and $m_V = \{m_V^r : \Omega_{\boldsymbol{PA}(V)} \mapsto \Omega_V\}_{r\in[1,R]}$ a set of R distinct mappings from $\Omega_{\boldsymbol{PA}(V)}$ to Ω_V, the **regional discrete mechanism** of an endogenous variables V is a function $f_V : \Omega_{\boldsymbol{PA}(V)}, \Omega_{U_V} \mapsto \Omega_V$ such that:

$$f_V(\mathbf{pa}(V),\ u_V) = m_r(\boldsymbol{PA}(V) \mapsto V) \text{ when } u_V \in I_V^r$$

I_V^r and m_r are called the r^{th} noise region and mapping of the regional discrete mechanism f_V.

Remark on Ω_{U_V} and R: In the definition of a regional discrete mechanism (Definition E.2), no constraints are imposed on Ω_{U_V}. However, if Ω_{U_V} is discrete, then $|\Omega_{U_V}| \geq R$ is required to form a partition of R elements of Ω_{U_V}. Consequently, in order to be able to constitute such a partition for any finite R, we decided to consider continuous exogenous variables in the definition of a regional discrete SCM (Definition E.1). In addition, since the m_V^r mappings are considered distinct and there are exactly $|\Omega_V|^{|\Omega_{\boldsymbol{PA}(V)}|}$ different mappings from V to $\boldsymbol{PA}(V)$, $R \leq |\Omega_V|^{|\Omega_{\boldsymbol{PA}(V)}|}$ is required.

Even if regional discrete SCMs are Markovian, the fact that they contains two types of endogenous variables (i.e., observed and unobserved by the user) enables the representation of complex situations where not all variables are observable. This induces the presence of potential hidden confounders from the user's perspective. As a result, the causal sufficiency assumption is no longer always respected. In our parametric definition of a *SoI*, this phenomenon is controlled by the parameter specifying the proportion of unobserved variables among the endogenous variables. Thus, if this parameter is set to 0, the *SoI*'s class of SCMs is included in the class of causally sufficient discrete SCMs.

The complexity of discrete mechanisms can be controlled by the number of noise regions R. Indeed, as the number of noise regions increases, so does the complexity of the causal mechanism, in the sense that it becomes a mixture of a larger number of mappings. The distribution of a variable given its parents is, hence, more stochastic. As a result, the user-defined class of regional discrete SCMs can be very broad. This provides an additional degree of complexity to make our synthetic causal datasets less trivial.

The class of regional discrete SCMs got inspired by the class of Regional Canonical Models by (Xia et al., 2023) and the class of canonical SCMs by (Zhang et al., 2022). We decided to define our own class rather than using one of these two classes for two reasons. First, canonical SCMs are very expensive to sample particularly because of the presence of confounded components. Second, even if Regional Canonical Models are designed to be less expensive because their expressivity can be regulated via the number of noise regions to consider, they lose some interesting properties such as the non overlapping of the noise regions which is crucial to favor a strong dependence between the user choice of the number of noise regions and the complexity of the generated mechanisms. Moreover, Regional Canonical Models still rely on confounded components which is the major source of complexity at the sampling stage. Hence, we defined the class of Regional discrete SCMs to not have to deal with confounded components at the sampling stage (instead we rely on a projection algorithm after sampling, see Appendix D) and to regulate mechanisms expressivity through the use of non-overlapping noise regions.

E.2 Discrete Mechanism Sampling strategies

We use *regional discrete mechanisms* (Definition E.2), which define tabular mappings from parent variables to a target variable, conditioned on regions of the exogenous noise space. By default, each region induces a distinct mapping, enabling both stochasticity and high functional expressivity.

To generate these mechanisms, we support three sampling strategies described below. All methods define a partition of the exogenous noise domain Ω_U into R regions, and assign a parent-to-child mapping to each region. Let C be the cardinality of the variables, and $\Omega_{\text{Pa}(V)}$ the space of parent configurations for variable V.

Controlling complexity. The number of possible mappings from parent configurations to output values grows as $|\Omega_V|^{|\Omega_{\text{Pa}(V)}|}$. To keep simulations tractable, users can control the number of noise regions R. When R is small, sampling provides diverse but lightweight mechanisms. When R approaches the total number of mappings, full enumeration becomes feasible but computationally expensive.

We now describe the three supported sampling strategies.

Exhaustive partition

This strategy enumerates all possible mappings from parent configurations to output values and assigns each one to a distinct noise region ($R = |\Omega_V|^{|\Omega_{\text{Pa}(V)}|}$), ensuring complete coverage of the function space. This method guarantees maximal functional diversity across regions and can serve as a stress test for generalization under highly non-linear mechanisms. This is the only strategy where the number of noise regions is not decided by the user but rather set to the maximum. The exhaustive partition sampling strategy is the one to use if one wants the coverage guarantee (Proposition 5.1) to apply.

Sample rejection

This strategy samples parent-to-output mappings uniformly at random, rejecting duplicates to ensure that each region corresponds to a distinct function. As mappings are sampled with replacement, rejection may require several attempts when R approaches the number of possible mappings.

We provide below, in Algorithm 4, a pseudocode version of this strategy. The algorithm proceeds as follows. For each endogenous variable V (line 2) a regional discrete mechanism is created. To do so, the domain of V is first initialized with a list of integers corresponding of the cardinality specified in the *SoI* (line 3). Then, if the number of noise regions R specified in the *SoI* is larger than the maximum number of noise regions, the maximum number of noise regions is used to generate the regional discrete mechanism (lines 4-5). The partition of the noise regions is built as consecutive intervals of random size resulting from the ordering of $R - 1$ sampled realizations of the uniform exogenous distribution (lines 6 to 9 and 13). Finally, for each noise region r (line 12), mappings m_V^r are sampled till one mapping not already used for other noise regions is sampled (lines 15 to 18). This is why this algorithm is denoted as the "sample rejection" approach. One can note that there are two sources of randomness in this algorithm: the size of the noise regions and the sampled mappings whenever the number of noise regions is not maximal.

Algorithm 4 Generating regional discrete mechanisms with sample rejection

Inputs: set of endogenous variables $\mathbf{V}$ of cardinality C, causal graph $\mathcal{G}$, Ω_U domain of exogenous variables, number of noise regions R

```
$\boldsymbol{\mathcal{F}} \leftarrow \{\}$
for $V \in \mathbf{V}$ do
    $\Omega_V \leftarrow \{1, \dots, C\}$
    $\Omega_{\boldsymbol{PA}_{\mathcal{G}}(V)} \leftarrow \{1, \dots, C\}^{|\boldsymbol{PA}_{\mathcal{G}}(V)|}$
    $R \leftarrow \min(R, |\Omega_V|^{|\Omega_{\boldsymbol{PA}(V)}|})$
    $l_{\min} \leftarrow \inf(\Omega_U)$
    $l_{\max} \leftarrow \sup(\Omega_U)$
    $\mathbf{L} = \{l_i \sim \mathcal{U}[l_{\min}, l_{\max}] \mid i \in \ [1, R-1]\} \cup \{l_{\min}, l_{\max}\}$
    Sort $\mathbf{L}$ in ascending order
    $f_V \leftarrow \{\}$
    $m_V \leftarrow \{\}$
    for $r \in [1, R]$ do
        $I_V^r \leftarrow [\mathbf{L}_r, \mathbf{L}_{r+1}[$ with $\mathbf{L}_r$ the $r^{th}$ element of $\mathbf{L}$
        $m_V^r \leftarrow \{\}$
        while $m_V^r = \{\}$ or $m_V^r \in m_V$ do
            $m_V^r \leftarrow |\Omega_{\boldsymbol{PA}(V)}|$ elements sampled with replacement from $\Omega_V$
        end while
        $m_V \leftarrow m_V \cup m_V^r$
        $f_V \leftarrow f_V \cup \{m_V^r; I_V^r\}$
    end for
    $\boldsymbol{\mathcal{F}} \leftarrow \boldsymbol{\mathcal{F}} \cup f_V$
end for
Output: $\boldsymbol{\mathcal{F}}$
```

Unbiased random assignment

In this strategy, each noise region is assigned a mapping sampled independently and without enforcing uniqueness. As a result, multiple regions may correspond to the same function from parent configurations to outputs.

For example, suppose a variable has one binary parent taking values in $\{0, 1\}$, and the output variable takes values in $\{0, 1, 2\}$. One randomly sampled mapping might assign output 0 to parent value 0, and output 2 to parent value 1. Since mappings are sampled independently for each region, this same function $(0 \to 0, 1 \to 2)$ may appear in multiple regions by chance.

This approach reflects scenarios where mechanisms are drawn independently from a distribution over functions, without enforcing any requirements on uniqueness or coverage. As a result, the effective variability in the entire system may be lower compared to other strategies, but the sampling is a lot more computationally efficient.

F Query Sampling and Estimation

In this work, we consider the following types of queries: Average Treatment Effect (ATE), Conditional Average Treatment Effect (CATE) and Ctf-TE. Their definitions can be found in Appendix B. All the queries can be defined for sets of covariates and factuals belonging to the set of endogenous variables. In other words, we do not implement multi-interventions, but we consider conditioning and observing factuals on several variables. Finally, the values taken by these variables (e.g., treatment and control values for ATE) must belong to their definition domain. The only parameter that controls the queries class is the type of queries chosen by the user (i.e., ATE, CATE and Ctf-TE). Thus, the class of considered queries can be defined as follows:

$$\mathcal{Q}_{\text{ATE}} = \{\text{ATE}_{T \to Y}(t, c) \mid T, Y \subseteq \mathbf{V} \text{ and } t, c \in \Omega_T\}$$

$$\mathcal{Q}_{\text{CATE}} = \{\text{CATE}_{T \to Y | \mathbf{X}}(t, c, \mathbf{x}) \mid T, Y \subseteq \mathbf{V},\ \mathbf{X} \subseteq \mathbf{V} \backslash \{T, Y\} \text{ and } t, c \in \Omega_T,\ \mathbf{x} \in \Omega_{\mathbf{X}}\}$$

$$\mathcal{Q}_{\text{Ctf-TE}} = \{\text{Ctf-TE}_{T \to Y}(y, t, c, \boldsymbol{v}_F) \mid T, Y, \boldsymbol{V}_F \subseteq \mathbf{V} \text{ and } t, c \in \Omega_T,\ y \in \Omega_Y,\ \boldsymbol{v}_F \in \Omega_{\boldsymbol{V}_F}\}$$

Formally speaking, we have not integrated the causal graph as a causal query but rather as a hypothesis or prior knowledge. Indeed, except for causal discovery tasks, the causal graph is most often assumed to be known (or at least some information derived from the graph, such as the constitution of a valid adjustment set, or a valid causal ordering). Nevertheless, one can use our random causal dataset generator to evaluate causal discovery or causal representation learning methods. To do so, one just needs to retrieve the causal graph from the causal dataset directly instead of using a query.

Finally, a user can also implement a specific query and use it to generate synthetic causal datasets. To do this, the user has to use the Query class in our code base.

F.1 Query Sampling

As the values taken by varaibles in the queries have to belong to their definition domain, we draw realizations from a large, separately sampled observational dataset. Indeed, given the randomness of the causal mechanisms, we cannot know in advance the domain over which the SCMs are defined. Even when variable cardinalities are fixed, the sampled mechanisms may be non-surjective, making certain values impossible to observe. For this reason, we approximate the domain of definition through data sampling, ensuring that queries are computed only for realizable variable configurations. Moreover, since the dataset given to the user is smaller to the one we use for query sampling and estimation, it is possible that queries use values outside of the observational dataset or that they are non-identifiable. Explicitly enabling queries to be outside the observed dataset can be useful for studying generalization—especially in settings where the support is known, such as linear SCMs. However, we let for future work the devlopement of a user-configurable option in *SoIs*, for instance, allowing users to define a custom domain for the query variables.

The following algorithms detail the procedures for sampling ATE, CATE, and Ctf-TE queries. In these algorithms, given a dataset D, a variable X and a realization x of X, we use the notation $D_{|X}$ (resp. $D_{|X=x}$) to represent the dataset D restricted to the variable X (resp. restricted to the samples whose X realization equals x). In addition, $B(n, p)$ denotes the Binomial law of parameters n and p.

Algorithm 5 Generating sets of observed data

```
Inputs: causal graph $\mathcal{G}$, causal mechanisms $\boldsymbol{\mathcal{F}}$, distribution of the exogenous variables $P(\mathbf{U})$, dataset size $N$

$D \leftarrow \{\}$
$D_o \leftarrow \{\}$
$\{\mathbf{u}_1, \dots, \mathbf{u}_N\} \sim P(\mathbf{U})$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    $\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\} \leftarrow D_{|\boldsymbol{PA}(V)}$
    $\{u_{V_1}, \dots, u_{V_N}\} \leftarrow D_{|\mathbf{U}_V}$
    $\{v_1, \dots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \dots, u_{V_N}\})$
    $D \leftarrow D \cup \{v_1, \dots, v_N\}$
    if $V \in \mathbf{V}_o$ then
        $D_o \leftarrow D_o \cup \{v_1, \dots, v_N\}$
    end if
end for
Output: $D_o$
```

Algorithm 6 Generating ATE queries

```
Inputs: set of observable endogenous variables $\mathbf{V}_o$, training set $D$
$T \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$Y \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$t \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
$c \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
Output: $Q_{ATE} = \{T, Y, t, c\}$
```

Algorithm 7 Generating CATE queries

```
Inputs: set of observable endogenous variables $\mathbf{V}_o$, training set $D$
$T \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$Y \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$d_{\mathbf{X}} \leftarrow$ an integer randomly sampled from $[1, \dots, |\mathbf{V}_o| - 2]$
$\mathbf{X} \leftarrow d_{\mathbf{X}}$ variables randomly sampled from $\mathbf{V}_o \backslash \{T, Y\}$
$t \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
$c \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
$\mathbf{x} \leftarrow$ one realization of $\mathbf{X}$ randomly sampled from $D_{|\mathbf{X}}$
Output: $Q_{CATE} = \{T, Y, \mathbf{X}, t, c, \mathbf{x}\}$
```

Algorithm 8 Generating Ctf-TE queries

```
Inputs: set of observable endogenous variables $\mathbf{V}_o$, training set $D$
$T \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$Y \leftarrow$ one variable randomly sampled from $\mathbf{V}_o$
$d_{\mathbf{V}_F} \leftarrow$ an integer randomly samples from $[1, \dots, |\mathbf{V}_o|]$
$\mathbf{V}_F \leftarrow d_{\mathbf{V}_F}$ variables randomly sampled from $\mathbf{V}_o$
$t \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
$c \leftarrow$ one realization of $T$ randomly sampled from $D_{|T}$
$\mathbf{v}_F \leftarrow$ one realization of $\mathbf{V}_F$ randomly sampled from $D_{|\mathbf{V}_F}$
Output: $Q_{CTF-TE} = \{T, Y, \mathbf{V}_F, t, c, \mathbf{v}_F\}$
```

F.2 SCM-Based Query Estimation

Each query is evaluated by modifying the SCM, sampling the exogenous variables, and computing expectations over the outcomes. In practice, we simulate interventions and counterfactuals by directly manipulating structural equations and conditioning on sampled variables. Our implementation supports efficient batch estimation using the same random seeds for reproducibility.

Queries that yield `NaN` estimates can optionally be rejected and resampled, depending on the *SoI* settings. `NaN` estimates appear if the corresponding sampled query is undefined (e.g., conditioning on a zero-probability event). However, to evaluate the ability of some models to identify if the query is undefined instead of trying to answer it, `NaN` estimates can be interesting to keep. This is why we decided to let users choose this option through a parameter of the *SoI*.

The following algorithms detail the procedures for estimating ATE, CATE, and Ctf-TE queries.

Algorithm 9 Estimating ATE queries

Inputs: ATE query to estimate $Q = \{T, Y, t, c\}$, causal graph $\mathcal{G}$, causal mechanisms $\boldsymbol{\mathcal{F}}$, distribution of the exogenous variables $P(\mathbf{U})$, number of samples to draw for estimation N

```
$\{\mathbf{u}_1, \dots, \mathbf{u}_N\} \sim P(\mathbf{U})$
$D_t \leftarrow \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \dots, v_N\} \leftarrow \{t, \dots, t\}$
    else
        $\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\} \leftarrow D_{t|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \dots, u_{V_N}\} \leftarrow D_{t|\mathbf{U}_V}$
        $\{v_1, \dots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \dots, u_{V_N}\})$
    end if
    $D_t \leftarrow D_t \cup \{v_1, \dots, v_N\}$
end for
$D_c \leftarrow \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \dots, v_N\} \leftarrow \{c, \dots, c\}$
    else
        $\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\} \leftarrow D_{c|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \dots, u_{V_N}\} \leftarrow D_{c|\mathbf{U}_V}$
        $\{v_1, \dots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \dots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \dots, u_{V_N}\})$
    end if
    $D_c \leftarrow D_c \cup \{v_1, \dots, v_N\}$
end for
$Q^\star \leftarrow \text{avg}(D_{t|Y}) - \text{avg}(D_{c|Y})$
```

Output: $Q^\star$

Algorithm 10 Estimating CATE queries

Inputs: CATE query to estimate $Q = \{T, Y, \mathbf{X}, t, c, \mathbf{x}\}$, causal graph $\mathcal{G}$, causal mechanisms $\boldsymbol{\mathcal{F}}$, distribution of the exogenous variables $P(\mathbf{U})$, number of samples to draw for estimation N

```
$\{\mathbf{u}_1, \ldots, \mathbf{u}_N\} \sim P(\mathbf{U})$
$D_t \leftarrow \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \ldots, v_N\} \leftarrow \{t, \ldots, t\}$
    else
        $\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\} \leftarrow D_{t|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \ldots, u_{V_N}\} \leftarrow D_{t|\mathbf{U}_V}$
        $\{v_1, \ldots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \ldots, u_{V_N}\})$
    end if
    $D_t \leftarrow D_t \cup \{v_1, \ldots, v_N\}$
end for
$D_c \leftarrow \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \ldots, v_N\} \leftarrow \{c, \ldots, c\}$
    else
        $\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\} \leftarrow D_{c|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \ldots, u_{V_N}\} \leftarrow D_{c|\mathbf{U}_V}$
        $\{v_1, \ldots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \ldots, u_{V_N}\})$
    end if
    $D_c \leftarrow D_c \cup \{v_1, \ldots, v_N\}$
end for
$D_t \leftarrow D_{t|\mathbf{X}=\mathbf{x}}$
$D_c \leftarrow D_{c|\mathbf{X}=\mathbf{x}}$
$Q^\star \leftarrow \text{avg}(D_{t|Y}) - \text{avg}(D_{c|Y})$
```

Output: $Q^\star$

Algorithm 11 Estimating Ctf-TE queries

Inputs: Ctf-TE query to estimate $Q = \{T, Y, \mathbf{V}_F, t, c, \mathbf{v}_F\}$, causal graph $\mathcal{G}$, causal mechanisms $\boldsymbol{\mathcal{F}}$, distribution of the exogenous variables $P(\mathbf{U})$, number of samples to draw for estimation N

```
$\{\mathbf{u}_1, \ldots, \mathbf{u}_N\} \sim P(\mathbf{U})$
$D_{\mathbf{U}_{\mathbf{v}_F}} \leftarrow \{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    $\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\} \leftarrow D_{\mathbf{U}_{\mathbf{v}_F}|\boldsymbol{PA}(V)}$
    $\{u_{V_1}, \ldots, u_{V_N}\} \leftarrow D_{\mathbf{U}_{\mathbf{v}_F}|\mathbf{U}_V}$
    $\{v_1, \ldots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \ldots, u_{V_N}\})$
    $D_{\mathbf{U}_{\mathbf{v}_F}} \leftarrow D_{\mathbf{U}_{\mathbf{v}_F}} \cup \{v_1, \ldots, v_N\}$
end for
$D_{\mathbf{U}_{\mathbf{v}_F}} \leftarrow D_{\mathbf{U}_{\mathbf{v}_F}|\mathbf{V}_F=\mathbf{v}_F}$
$M \leftarrow |D_{\mathbf{U}_{\mathbf{v}_F}}|$
$\{\mathbf{u}_1, \ldots, \mathbf{u}_M\} \leftarrow D_{\mathbf{U}_{\mathbf{v}_F}|\mathbf{U}}$
$D_t \leftarrow \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \ldots, v_N\} \leftarrow \{t, \ldots, t\}$
    else
        $\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\} \leftarrow D_{t|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \ldots, u_{V_N}\} \leftarrow D_{t|\mathbf{U}_V}$
        $\{v_1, \ldots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \ldots, u_{V_N}\})$
    end if
    $D_t \leftarrow D_t \cup \{v_1, \ldots, v_N\}$
end for
$D_c \leftarrow \{\mathbf{u}_1, \ldots, \mathbf{u}_M\}$
for $V \in \mathbf{V}$ following a causal order given by $\mathcal{G}$ do
    if $V = T$ then
        $\{v_1, \ldots, v_N\} \leftarrow \{c, \ldots, c\}$
    else
        $\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\} \leftarrow D_{c|\boldsymbol{PA}(V)}$
        $\{u_{V_1}, \ldots, u_{V_N}\} \leftarrow D_{c|\mathbf{U}_V}$
        $\{v_1, \ldots, v_N\} \leftarrow f_V(\{\mathbf{pa}(V)_1, \ldots, \mathbf{pa}(V)_N\}, \{u_{V_1}, \ldots, u_{V_N}\})$
    end if
    $D_c \leftarrow D_c \cup \{v_1, \ldots, v_N\}$
end for
$Q^\star \leftarrow \text{avg}(D_{t|Y}) - \text{avg}(D_{c|Y})$
```

Output: $Q^\star$

G Analysis module's metrics

In order to analyze the characteristics of the sampled SCMs we implemented the following metrics. Let us imagine we sampled an SCM $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$ with $\mathbf{V} = (\mathbf{V}_o, \mathbf{V}_h)$ and whose causal graph is denoted $\mathcal{G}$. The projection of $\mathcal{G}$ over the observable variables $\mathbf{V}_o$ is denoted $\mathcal{G}_{\mathbf{V}_o}$.

Analysis of the causal graph $\mathcal{G}$:

- Average in-degree: $\bar{d}_{in} = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} |\boldsymbol{PA}(V)|$
- Variance of in-degree: $\text{var}(d_{in}) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} (|\boldsymbol{PA}(V)| - \bar{d}_{in})^2$
- Average number of ancestors: $\overline{|An(V)|} = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} |An(V)|$ where $An(V)$ denotes the set of ancestors of V
- Variance of number of ancestors: $\text{var}(|An(V)|) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} (|An(V)| - \overline{|An(V)|})^2$
- Average number of descendants: $\overline{|De(V)|} = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} |De(V)|$ where $De(V)$ denotes the set of descendants of V
- Variance of number of descendants: $\text{var}(|De(V)|) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} (|De(V)| - \overline{|De(V)|})^2$
- Average length of causal paths: $\overline{L} = \frac{1}{|\mathbf{p}_\mathcal{G}|} \sum_{p \in \mathbf{p}_\mathcal{G}} |p|$ where $\mathbf{p}_\mathcal{G}$ denotes the set of directed paths in $\mathcal{G}$
- Variance length of causal paths: $\text{var}(L) = \frac{1}{|\mathbf{p}_\mathcal{G}|} \sum_{p \in \mathbf{p}_\mathcal{G}} (|p| - \overline{L})^2$
- Maximum length of causal paths: $L_{\max} = \max_{p \in \mathbf{p}_\mathcal{G}} |p|$

Analysis of the projected causal graph $\mathcal{G}_{\mathbf{V}_o}$:

- Average number of siblings[4]: $\overline{|Si(V)|} = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} |Si(V)|$ where $Si(V)$ denotes the set of siblings of V
- Variance of number of siblings: $\text{var}(|Si(V)|) = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} (|Si(V)| - \overline{|Si(V)|})^2$
- Number of maximal confounded components (c-comps)[5]: $|\mathbf{C}|$ where $\mathbf{C}$ denotes the set of maximal c-comps in $\mathcal{G}_{\mathbf{V}_o}$
- Average size of maximal c-comps: $\overline{|\mathbf{C}|} = \frac{1}{|\mathbf{C}|} \sum_{C \in \mathbf{C}} |C|$
- Variance of the size of maximal c-comps: $\text{var}(|\mathbf{C}|) = \frac{1}{|\mathbf{C}|} \sum_{C \in \mathbf{C}} (|C| - \overline{|\mathbf{C}|})^2$

Analysis of the observational distribution $P_\mathcal{M}(\mathbf{V_o})$:

- Minimum probability of the joint distribution: $p_{\mathbf{V}_o, \min} = \min_{\mathbf{v}_o \in \Omega_{\mathbf{V}_o}} P_\mathcal{M}(\mathbf{V}_o = \mathbf{v}_o)$
- Proportion of events with a null probability: $p_0 = \frac{1}{|\Omega_{\mathbf{V}_o}|} \sum_{\mathbf{v}_o \in \Omega_{\mathbf{V}_o}} \mathbf{1}_{P_\mathcal{M}(\mathbf{V_o} = \mathbf{v}_o) = 0}$ where $\mathbf{1}_{\_}$ denotes the indicator function
- Minimum probability of the marginal distributions:
$$p_{\min} = \min_{V \in \mathbf{V}_o} \min_{v \in \Omega_V} P_\mathcal{M}(V = v)$$
- Average minimum probability of the marginal distributions:
$$\bar{p}_{\min} = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} \frac{1}{|\Omega_V|} \min_{v \in \Omega_V} P_\mathcal{M}(V = v)$$

[4] Two variables are considered siblings if they are linked by a bi-directed edge.

[5] We use (Tian & Pearl, 2002) definition of (maximal) confounded components.

- Variance of the minimum probability of the marginal distributions:

$$\text{var}(p_{\min}) = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} (\min_{v \in \Omega_V} P_{\mathcal{M}}(V = v) - \bar{p}_{\min})^2$$

- Distance (L_1) of the joint distributions to the uniform one:

$$d(P_{\mathcal{M}}; \mathcal{U}) = \sum_{\mathbf{v}_o \in \Omega_{\mathbf{V}_o}} |P_{\mathcal{M}}(\mathbf{V}_o = \mathbf{v}_o) - \frac{1}{|\Omega_{\mathbf{V}_o}|}|$$

- Average distance (L_1) of the marginal distributions to the uniform one:

$$\overline{d(P_{\mathcal{M}}; \mathcal{U})} = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} \sum_{v \in \Omega_V} |P_{\mathcal{M}}(V = v) - \frac{1}{|\Omega_V|}|$$

- Variance of the distance (L_1) of the marginal distributions to the uniform one:

$$\text{var}(d(P_{\mathcal{M}}; \mathcal{U})) = \frac{1}{|\mathbf{V}_o|} \sum_{V \in \mathbf{V}_o} \left(\sum_{v \in \Omega_V} |P_{\mathcal{M}}(V = v) - \frac{1}{|\Omega_V|}| - \overline{d(P_{\mathcal{M}}; \mathcal{U})} \right)^2$$

- Entropy of the joint distribution: $\text{H}(P_{\mathcal{M}}(\mathbf{V}))$

All the above-mentioned probabilities are computed from a set of 1M samples drawn from the SCM $\mathcal{M}$.

Let us note that $p_{\min}$ enables the user to check if the strong positivity assumption holds. If $p_{\mathbf{V}_o,\min} > 0$, then strong positivity is respected. In addition, if strong positivity does not hold, $p_{\mathbf{V}_o,\min}$ and p_0 indicate the extent to which the assumption is not met – the higher the metrics, the less the hypothesis is respected. On the other hand, $p_{\min}$ indicates whether the weak positivity assumption holds. If $p_{\min} > 0$, then weak positivity is respected. Finally, $d(P_{\mathcal{M}}; \mathcal{U})$, $\overline{d(P_{\mathcal{M}}; \mathcal{U})}$ and $\text{var}(d(P_{\mathcal{M}}; \mathcal{U}))$ enables the user to assess to which extent the observational distribution is imbalanced.

Analysis of the causal mechanisms $\boldsymbol{\mathcal{F}}$:

- Average Pearson's correlation between the parent-child pairs[6]:

$$\bar{\rho}_P = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} \frac{1}{|\boldsymbol{PA}(V) \cup U_V|} \sum_{V_j \in \boldsymbol{PA}(V) \cup U_V} \rho_P(V, V_j)$$

- Variance of Pearson's correlation between the parent-child pairs:

$$\text{var}(\rho_P) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} \frac{1}{|\boldsymbol{PA}(V) \cup U_V|} \sum_{V_j \in \boldsymbol{PA}(V) \cup U_V} (\rho_P(V, V_j) - \bar{\rho}_P)$$

- Average Spearman's correlation between the parent-child pairs[3]

$$\bar{\rho}_S = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} \frac{1}{|\boldsymbol{PA}(V) \cup U_V|} \sum_{V_j \in \boldsymbol{PA}(V) \cup U_V} \rho_S(V, V_j)$$

- Variance of Spearman's correlation between the parent-child pairs:

$$\text{var}(\rho_S) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} \frac{1}{|\boldsymbol{PA}(V) \cup U_V|} \sum_{V_j \in \boldsymbol{PA}(V) \cup U_V} (\rho_S(V, V_j) - \bar{\rho}_S)$$

- Average conditional entropy of a variable given its parents:

$$\overline{\text{H}} = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} \text{H}(V | \boldsymbol{PA}(V))$$

[6] ρ_P and ρ_S respectively denote the Pearson's and Spearman's correlation

- Variance of conditional entropy of a variable given its parents:

$$\mathrm{var}(\mathrm{H}) = \frac{1}{|\mathbf{V}|} \sum_{V \in \mathbf{V}} (\mathrm{H}(V|\boldsymbol{PA}(V)) - \overline{\mathrm{H}})^2$$

In order to be able to use person correlations, spearman correlations, and conditional entropy as indicators of degrees of linearity, monotonicity, and stochasticity of causal mechanisms, we do not derive these quantities from samples drawn from the entailed distribution. Instead, for each variable, we create a dataset resulting from the application of its causal mechanism to the cartesian product of the values taken by its endogenous and exogenous parents[7]. In other words, we analyze the mechanisms' images of their input space. This allows us to analyze each mechanism independently of the others.

Thus, $\bar{\rho}_P$ and $\mathrm{var}(\rho_P)$ can be interpreted as the average degree of linearity of causal mechanisms and their variance. Furthermore, $\bar{\rho}_S$ and $\mathrm{var}(\rho_S)$ can be interpreted as the average degree of monotonicity of causal mechanisms and their variance. Finally, $\overline{\mathrm{H}}$ and $\mathrm{var}(\mathrm{H})$ can be interpreted as the average level of stochasticity of causal mechanisms and its variance.

H Analysis of the empirical distribution of the generated SCMs

As we do not provide the user with an expression of the distribution of the sampled regional discrete SCMs, we need to investigate if some SCMs classes are over/underrepresented. This analysis is important to identify the potential biases CausalProfiler might create in order to take them into account when evaluating Causal ML methods. Indeed, as our goal is to provide a tool for rigorous empirical evaluation of causal methods, we need to be transparent on the limitations of our generator so that researchers and practitioners can interpret the results of their methods with full knowledge of the potential biases coming from CausalProfiler.

H.1 Experiment

To visualize the distribution of the SCMs generated, we analyze the distribution of the metrics of the analysis module characterizing the SCMs. For each SCM sampled, all the implemented metrics (see Appendix G) are computed.

The studied SCMs are sampled from the SoIs defined by the cartesian product of the following parameters:

- **Number of endogenous variables**: $\{3, 4, 5\}$
- **Expected edge probability**: $\{0.2, 0.4, 0.6, 0.8\}$
- **Proportion of unobserved endogenous variables**: $\{0, 0.1, 0.2, 0.3\}$
- **Number of noise regions**: $\{2, 5, 10, 20, 50\}$
- **Cardinality of endogenous variables**: $\{2, 3, 4, 7\}$
- **Distribution of exogenous variables**: set to $\mathcal{U}[0, 1]$

For each *SoI* 10 SCMs are sampled, making a total of 9600 SCMs studied. Let us mention that we sample more SCMs than for verification (Section 6.1 for two reasons. First, it enables us to have a better approximation of the SCMs distribution. Second, the computation of all the assumptions and characteristics metrics is, in fact, less computationally expensive than computing all the independence tests that were required for verification.

[7] For continuous SCMs, we first discretize the variables' domains of definition and then build the cartesian product.

H.2 Results

The first conclusion, based on Figures 3 to 7, is that the generated SCMs do indeed belong to the specified SoIs and that their characteristics are consistent with the latter.

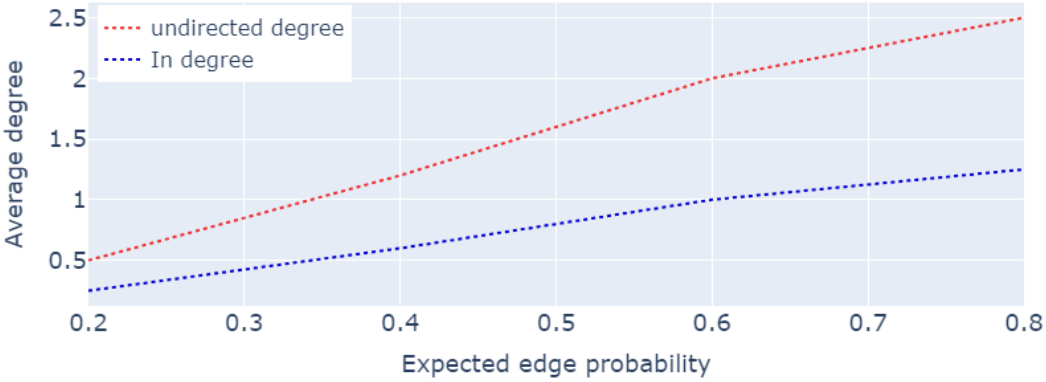


Figure 3: Average degree of the causal graphs for the generated SCMs depending on the expected edge probability. Observation: The average degree corresponds on average to the degree of the generated causal graphs.

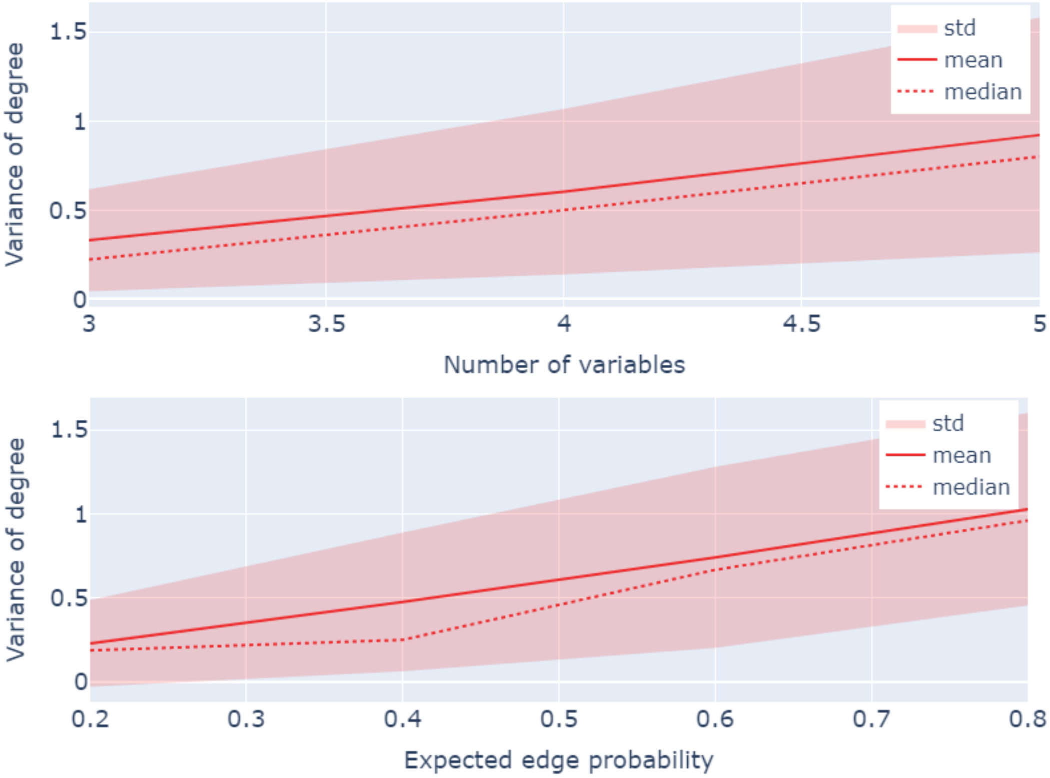


Figure 4: Variance of the causal graphs' degree of the generated SCMs depending on the number of variables and the expected edge probability. Observation: The variance of the degree increases with the size of the graph and its density.

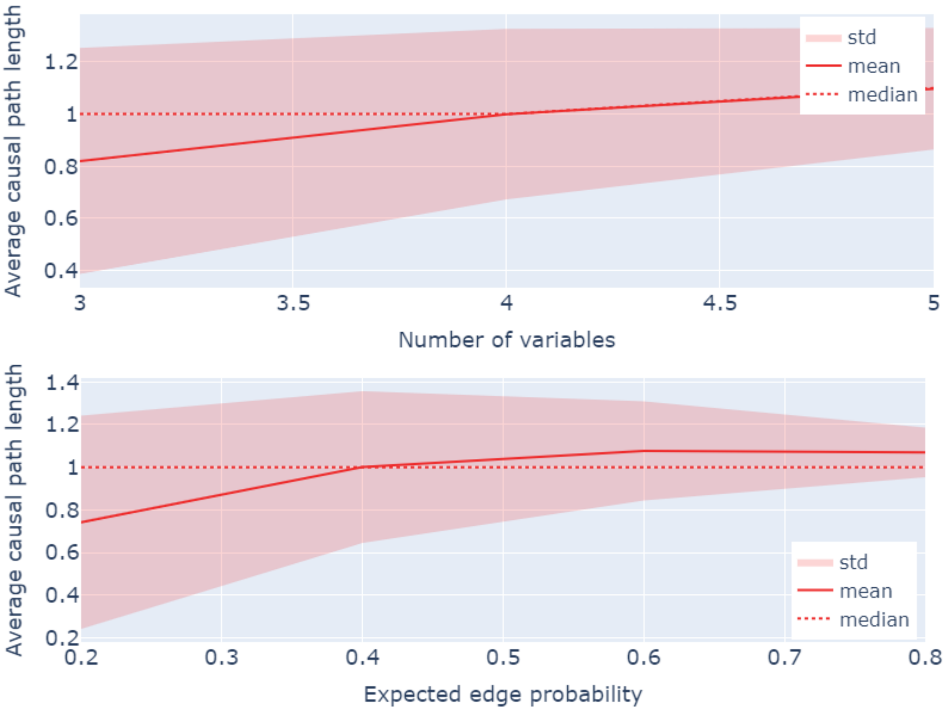


Figure 5: Average causal paths length of the causal graphs of the generated SCMs depending on the number of variables and the expected edge probability. Observation: The length of causal paths increases with the size of the causal graph and its density.

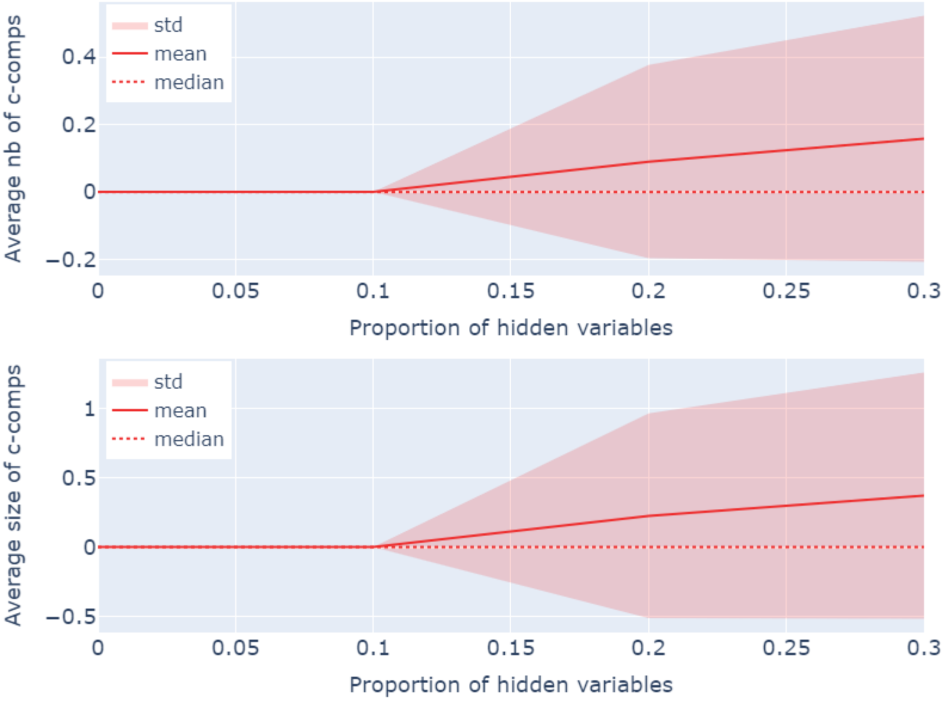


Figure 6: Average number and size of maximally confounded components in the projected causal graphs of the generated SCMs depending on the number of unobserved variables. Observation: The number and size of confounded components increase with the proportion of unobserved variables.

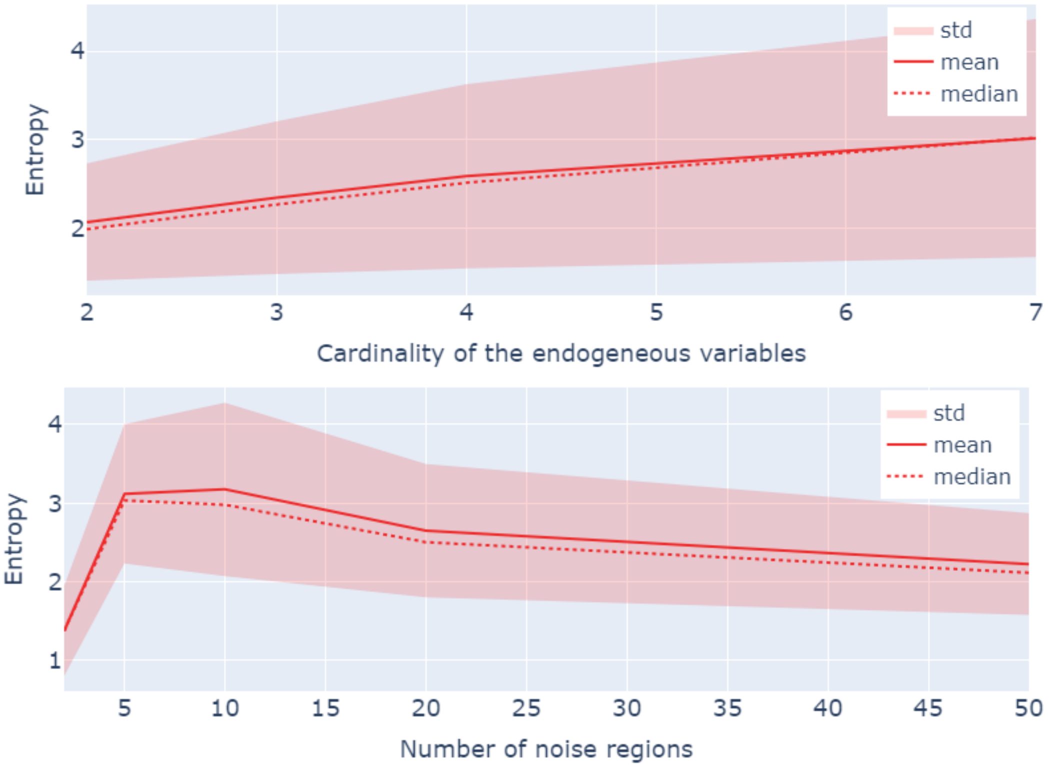


Figure 7: Average conditional entropy of a variable given its parents in the generated SCMs depending on the variables' cardinality and the number of noise regions. Observation: The stochasticity of causal mechanisms increases with the cardinality of endogenous and exogenous variables.

Table 4: Percentage of SCMs with confounded components depending on their proportion of unobserved endogenous variables.

	Number of maximally confounded components		
Unobserved endo. variables (%)	0	1	>1
0	100	0	0
10	100	0	0
20	90.9	9.1	0
30	83.3	16.7	0

Table 5: Percentage of SCMs with confounded components of different sizes depending on their proportion of unobserved endogenous variables. The size 1 of confounded components is not referenced, as if a confounded component is not empty, it is at least composed of two variables.

	Avg. size of maximally confounded components				
Unobserved endo. variables (%)	0	2	3	4	>4
0	100	0	0	0	0
10	100	0	0	0	0
20	90.9	5.7	2.3	1.0	0
30	83.3	10.8	4.9	1.0	0

In addition, a number of findings about the distribution of the sampled SCMs can also be drawn. First, the number and size of confounded components often equals zero (see also Tables 4 and 5). As highly confounded SCMs are rare, we recommend that users sample SCMs with a large enough number of variables and edge probability, if they want to consider graphs containing hidden confounders. For instance, we recommend at least 10 variables with a 50% edge probability to have a large proportion of graphs with at least one confounded component when setting the proportion of hidden endogenous variables to 30%.

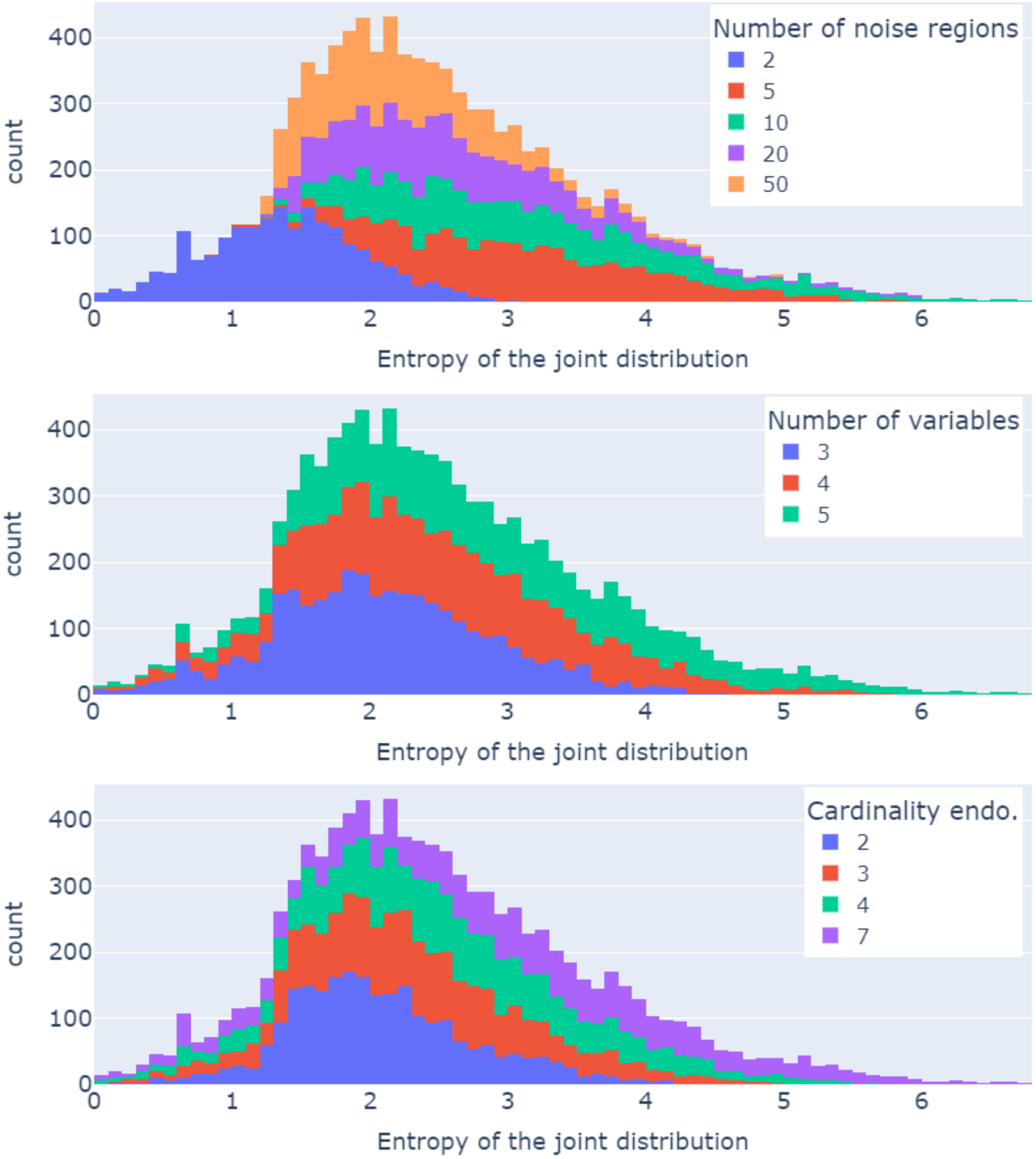


Figure 8: Stacked histograms of the stochasticity level (measured through the entropy of the $\mathcal{L}_1$ joint distribution) of the sampled SCMs depending on the number of noise regions, the number of variables, and their cardinality. Mean, standard deviation, and skewness of the distributions can be found in Tables 6 to 8.

Second, analyzing the stochasticity level (measured through the entropy of the $\mathcal{L}_1$ joint distribution, see Appendix G) of the generated SCMs, one can see that the latter can be controlled in part by the parameters of the *SoI*. Indeed, increasing the number of endogenous variables and their cardinality tends to increase the level of stochasticity, see Figure 8 and Tables 6 and 7. This behavior is expected as the discrete mechanisms are randomly sampled with an almost null probability of being deterministic (i.e., the probability of sampling a noise region with an empty support is almost null).

Table 6: Mean, standard deviation, and skewness of the distribution of stochasticity level (measured through the entropy of the $\mathcal{L}_1$ joint distribution) over the sampled SCMs depending on their number of endogenous variables. The distribution is displayed in Figure 8.

	Entropy of the joint distribution		
Number of endogenous variables	Mean	Std	Skewness
3	2.09	0.77	0.19
4	2.54	1.03	0.28
5	2.88	1.21	0.46

Table 7: Mean, standard deviation, and skewness of the distribution of stochasticity level (measured through the entropy of the $\mathcal{L}_1$ joint distribution) over the sampled SCMs depending on the cardinality of their endogenous variables. The distribution is displayed in Figure 8.

	Entropy of the joint distribution		
Cardinality	Mean	Std	Skewness
2	2.06	0.67	0.50
3	2.34	0.84	0.36
4	2.57	1.02	0.19
7	3.03	1.35	0.05

Table 8: Mean, standard deviation, and skewness of the distribution of stochasticity level (measured through the entropy of the $\mathcal{L}_1$ joint distribution) over the sampled SCMs depending on their number of noise regions. The distribution is displayed in Figure 8.

	Entropy of the joint distribution		
Number of noise regions	Mean	Std	Skewness
2	1.35	0.56	0.08
5	3.12	0.87	0.37
10	3.15	1.10	0.74
20	2.65	0.86	0.88
50	2.24	0.65	0.84

In addition, increasing the number of noise regions and the number of variables tends to increase the asymmetry of the distribution, see Figure 8 and Tables 6 and 8. This illustrates the fact that the number of degrees of freedom is increasing, and that it is therefore possible to generate increasingly stochastic mechanisms, although their probability of being sampled remains low. On the contrary, increasing the cardinality of the endogenous variables seems to reduce the asymmetry of the distribution, which may seem surprising. In reality, the distribution flattens out at higher stochasticity levels, making it more symmetrical. Indeed, both the mean and the standard deviation increase.

This analysis also reveals a surprising result: The number of noise regions does not seem to increase the level of stochasticity, cf. Figure 8 and Table 8. Theoretically, the more noise regions, the higher the number of mappings defining a causal mechanism. By complementing this mixture, we could expect to obtain a higher level of stochasticity. Further analysis is therefore required here to clarify the effect of the noise region parameter on stochasticity.

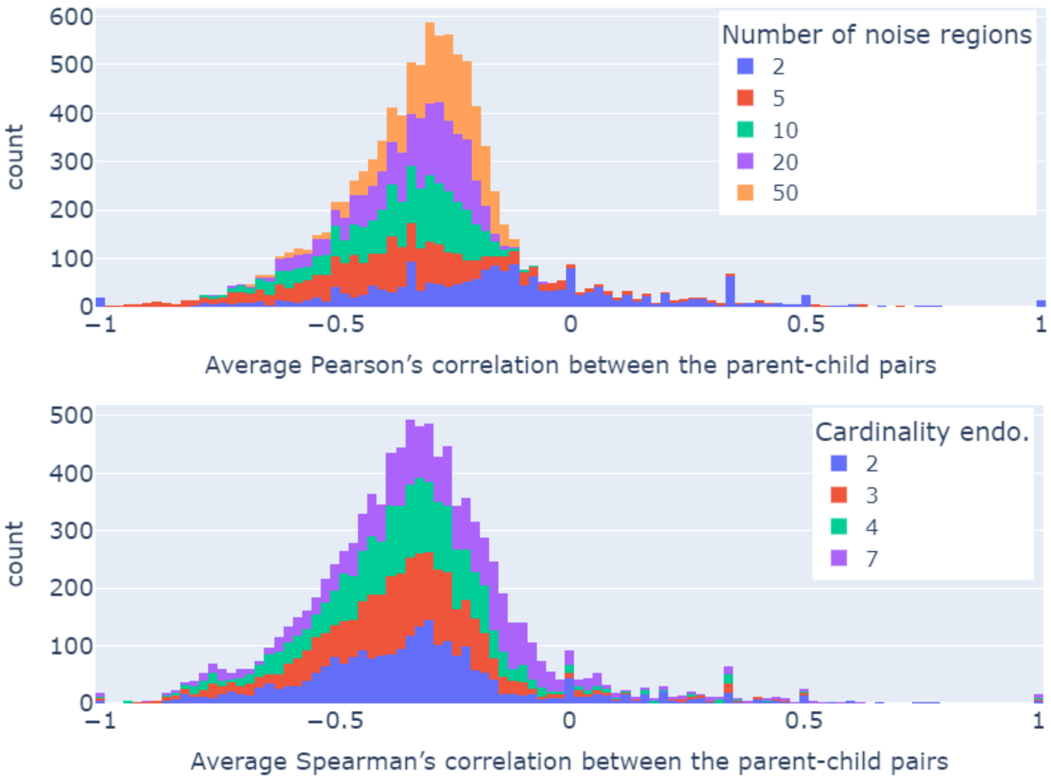


Figure 9: Stacked histograms of the average Pearson's and Spearman's correlation between the parent-child pairs of the generated SCMs. Mean, standard deviation, and skewness of the distributions can be found in Tables 9 and 10.

Table 9: Mean, standard deviation, and skewness of the distribution of the average Pearson's correlation between the parent-child pairs of the generated SCMs depending on their number of noise regions. The distribution is displayed in Figure 9.

	Pearson's correlation		
Number of noise regions	Mean	Std	Skewness
2	-0.15	0.30	0.54
5	-0.38	0.22	0.60
10	-0.36	0.13	-0.73
20	-0.34	0.11	-0.68
50	-0.30	0.10	-0.80

Table 10: Mean, standard deviation, and skewness of the distribution of the average Spearman's correlation between the parent-child pairs of the generated SCMs depending on the cardinality of their endogenous variables. The distribution is displayed in Figure 9.

	Pearson's correlation		
Cardinality	Mean	Std	Skewness
2	-0.32	0.25	1.10
3	-0.36	0.20	0.97
4	-0.35	0.19	1.09
7	-0.27	0.22	0.57

Third, the analysis of the levels of linearity and monotonicity (measured using Pearson and Spearman correlations) reveals that the sampled causal mechanisms are mostly neither linear nor monotonic, see Figure 9. Even if this result is to be expected, as the regional discrete mechanisms are discrete mappings without any notion of ordering, the fact that all the distributions are constituted of one peak on the negative side instead of two peaks, symmetric with respect to 0 is surprising. Hence, more investigation remains to be done to understand if our sampling algorithm tends to favor the generation of monotonically decreasing mechanisms.

One can also notice from Tables 9 and 10 that neither the cardinality of the endogenous variables nor the number of noise regions seems to affect the mean of the distributions, which is close to 0.35. In particular, the cardinality seems to have no effect on the distribution, while increasing the number of noise regions seems to increase the asymmetry of the distribution towards more linear mechanisms and decrease the standard deviation. Hence, we warn the users that choosing a high number of noise regions, hoping to be very diverse when generating mechanisms, might create the opposite effect over some metrics, as the distributions of Spearman's and Pearson's correlations seem to narrow down in this analysis.

Table 11: Percentage of SCMs respecting the strong positivity assumption depending on the number of endogenous variables.

	Avg. min. proba. of the joint distribution	
Number of variables	0	>0
3	88.6	11.4
4	95.6	4.4
5	97.0	3.0

Table 12: Percentage of SCMs respecting the strong positivity assumption depending on the cardinality of the endogenous variables.

	Avg. min. proba. of the joint distribution	
Cardinality	0	>0
2	95.7	4.3
3	97.5	2.5
4	95.5	4.5
7	89.8	10.2

Finally, Tables 11 and 12 illustrate that the assumption of strong positivity is rarely respected for all kinds of SCMs, whereas weak positivity is respected for all the sampled SCMs. More precisely, strong positivity hold on average in 6% of the generated datasets. This figure should be interpreted as a conservative lower bound. Indeed, our check uses finite samples, while strong positivity is defined in the infinite-sample regime: we reported a violation whenever any realization had an empirical frequency of 0 in 10,000 samples. In addition, there does not seem to be a correlation between the cardinality of the endogenous variables and the validation of the positivity assumption. It seems to mainly depend on the number of variables, which makes sense as the number of possible observations increases exponentially with the number of variables. Failure to respect the strong positivity assumption is a direct consequence of working with finite data, where infinitesimal probabilities are rounded to zero.

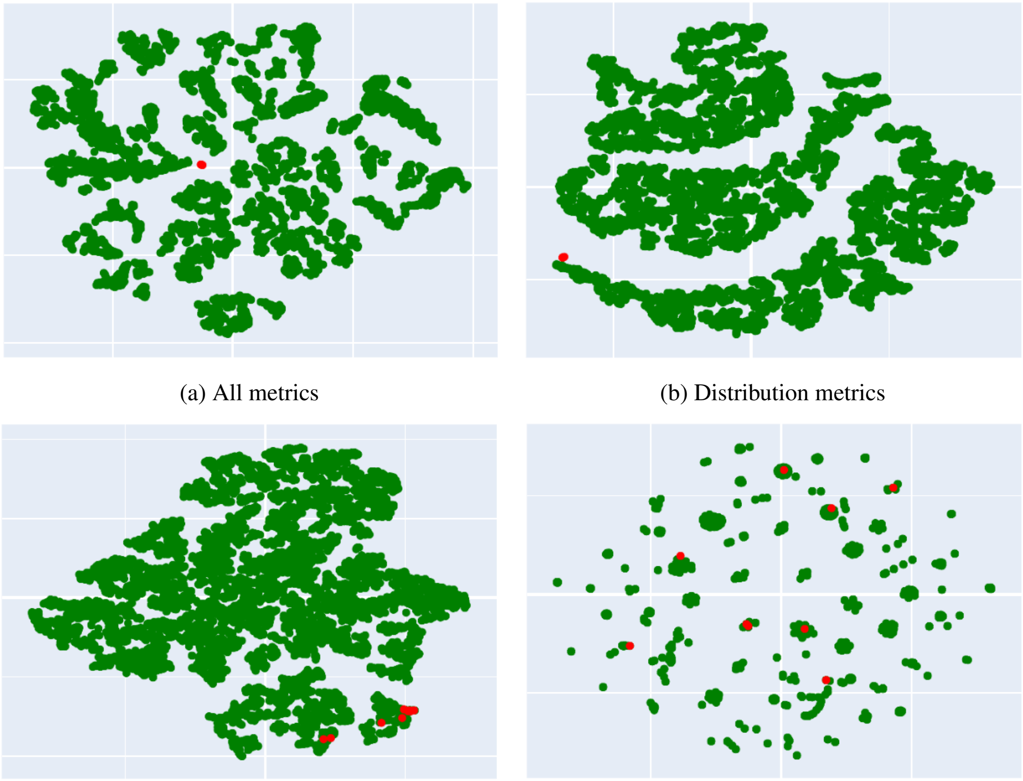
We therefore recommend that in order to evaluate Causal ML methods taking the strong positivity assumption, users use our analysis module to classify the sampled SCMs into two groups, depending on their compliance with the strong positivity assumption or not, and analyze them separately. This isolates the performance analysis within the theoretical validity framework of the method, and the analysis of its robustness to the violation of this assumption.

As a result, the generated SCMs belong mainly to the non-identifiable domain of Causal ML methods, as positivity is poorly respected. Users must, therefore, be careful in their interpretations when evaluating methods, as identifiable SCMs are much less represented than non-identifiable ones. We recommend starting the evaluation on small SoIs close to the identifiable domain, before progressively increasing the complexity of the causal datasets generated.

Let us highlight that this study was only carried out on regional discrete SCMs. We reserve for future work its extension to continuous SCMs.

H.3 Comparision to CausalNF synthetic SCMs used for evaluation

To illustrate the contribution in SCMs diversity that CausalProfiler can give to practitioners wishing to evaluate Causal ML methods, we compare the SCMs sampled in the previous Section with those used in the CausalNF work (Javaloy et al., 2023) for evaluation. We decided to first focus on the CausalNF synthetic SCMs because they have been reused by other papers (Sick & Dürr, 2025; Zhou et al., 2025) to evaluate new methods as if they were classical synthetic benchmarks for counterfactual evaluation.



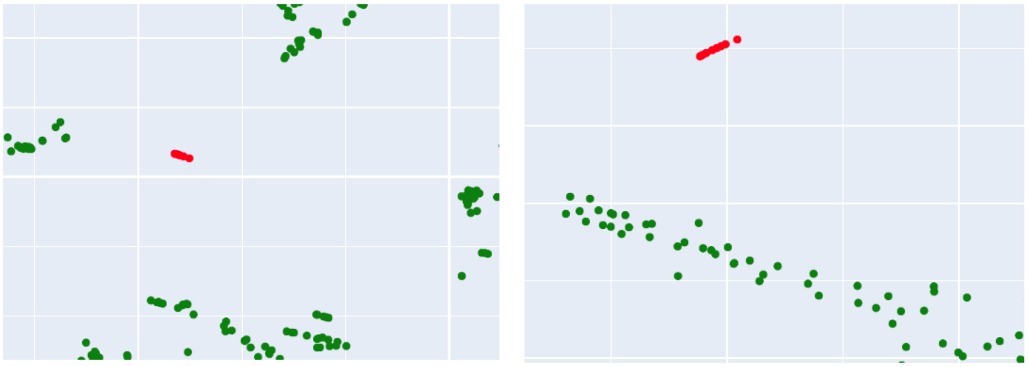
(a) All metrics

(b) Distribution metrics

(c) Mechanism metrics

(d) Graph metrics

Figure 10: Two-dimensional t-SNE plots representing our sampled SCMs (green) and the synthetic SCMs used for evaluation of CausalNF (red). The latter, less numerous, have been plotted in the foreground to highlight their distribution in relation to our SCMs. The SCMs are described using characterization metrics from the analysis module. (a) t-SNE plot using all metrics (b) t-SNE plot using distribution metrics only (c) t-SNE plot using mechanism metrics only (d) t-SNE plot using graph metrics only.



(a) All metrics zoomed ×9

(b) Distribution metrics zoomed ×17

Figure 11: t-SNE plots of Figures 10a and 10b zommed on CausalNF synthetic SCMs.

For this comparison, we reimplemented the synthetic SCMs of CausalNF using CausalProfiler, and applied all the metrics of the analysis module (cf. Appendix G). In this way, the CausalNF SCMs were processed in the same way as our SCMs. We then used these metrics to compare the two groups of SCMs. For the sake of having a fair comparison, not penalizing the fact that some assumptions were taken by the authors, we removed some metrics from the analysis: the hidden confounders and positivity metrics. Indeed, all CausalNF SCMs satisfy the causal sufficiency and strong positivity hypotheses, whereas, as presented in Appendix H.2, our SCMs do not by design. Finally, in order to obtain an easily interpretable visual result, we applied a two-dimensional t-SNE projection (Maaten & Hinton, 2008) to all these metrics and subgroups of metrics (Figure 10). Each t-SNE has been applied here with a perplexity of 30.

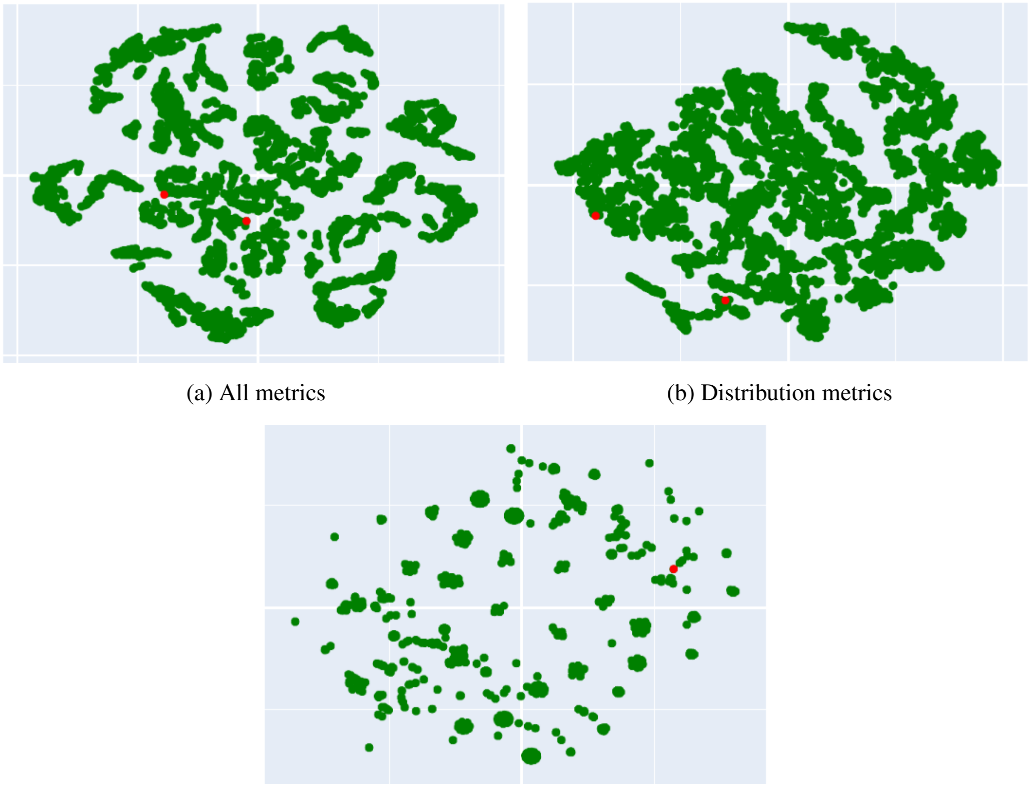
It can be seen that our SCMs are more diverse than those of CausalNF. Regarding graph metrics, it seems that CausalNF already has good diversity. The fact that we have greater support could mainly stem from the fact that we sampled a large number of SCMs. On the other hand, regarding distributions and mechanisms metrics, the increase in diversity is clear: The CausalNF SCMs are so similar compared to the total diversity that the dimension reduction projected them onto a confined space, cf. Figure 11.

As a result, we can conclude that CausalProfiler can enable practitioners to evaluate Causal ML methods on a more diverse set of SCMs and naturally derive more conclusions.

H.4 Comparision to bnlearn semi-sythetic graphical causal models

This section also illustrates the contribution CausalProfiler makes to SCMs' diversity by comparing them with other causal models used in the literature: CANCER and EARTHQUAKE from bnlearn (Scutari, 2019). Unlike the synthetic and continuous SCMs from CausalNF, CANCER and EARTHQUAKE are discrete causal graph models. The following analysis, therefore, enriches the conclusions of the previous section.

CANCER and EARTHQUAKE were compared to the SCMs sampled by CausalProfiler in the same way as in the previous section: a two-dimensional t-SNE projection is applied to the metrics from the analysis module. The only difference here is that the mechanisms metrics cannot be computed on CANCER and EARTHQUAKE, as they are not proper SCMs, but graphical causal models. For the sake of having a fair comparison, we also excluded the hidden confounders metrics (as both bnlearn graphs are DAGs) but kept the positivity metrics for this analysis.



(a) All metrics

(b) Distribution metrics

(c) Graph metrics

Figure 12: Two-dimensional t-SNE plots representing our sampled SCMs (green) and the semi-synthetic graphical causal models CANCER and EARTHQUAKE from bnlearn (red). The SCMs are described using characterization metrics from the analysis module. (a) t-SNE plot using all metrics (b) t-SNE plot using distribution metrics only (c) t-SNE plot using graph metrics only. Mechanism metrics cannot be used as bnlearn models do not model mechanisms but rather distributions.

The results, presented in Figure 12, show that the two bnlearn datasets are not confined to a small region of the two-dimensional space. Instead, they fall within the bottom left region of the t-SNE plot, overlapping with some of our generated SCMs. Hence, the conclusion of this analysis is similar to the previous one: CausalProfiler can generate SCMs producing similar causal datasets to existing ones while also generating more diverse sets of SCMs.

Why do we compare CausalProfiler SCMs to CausalNF synthetic SCMs and bnlearn datasets instead of datasets like IHDP, Twins, Syntren, or ACIC2016? Our comparison focuses on the diversity of underlying SCMs, which requires access to the full structural model (graph, mechanisms, and exogenous noise). These benchmarks do not expose their underlying SCMs needed to compute SCM-level metrics used in our analysis.

Further, for this analysis we require datasets whose characteristics match those of the studied SoIs, in particular datasets with a small number of variables (3-5). This is why we include CausalNF and bnlearn networks, and exclude IHDP (Hill, 2011), Twins (Louizos et al., 2017), Syntren (den Bulcke et al., 2006), and ACIC2016 (Dorie et al., 2019), which contain substantially more variables. One might argue that we could simply sample higher-dimensional SCMs from CausalProfiler. While this is possible, computing the full set of assumption-analysis metrics (Appendix G) becomes computationally expensive as dimensionality and graph density increase; for example, Markov property checks and pairwise independence tests scale poorly with the number of variables. As a result, performing a detailed comparison with higher-dimensional datasets is not very tractable.

I VISUAL OVERVIEW OF CAUSALPROFILER'S SAMPLING STRATEGY

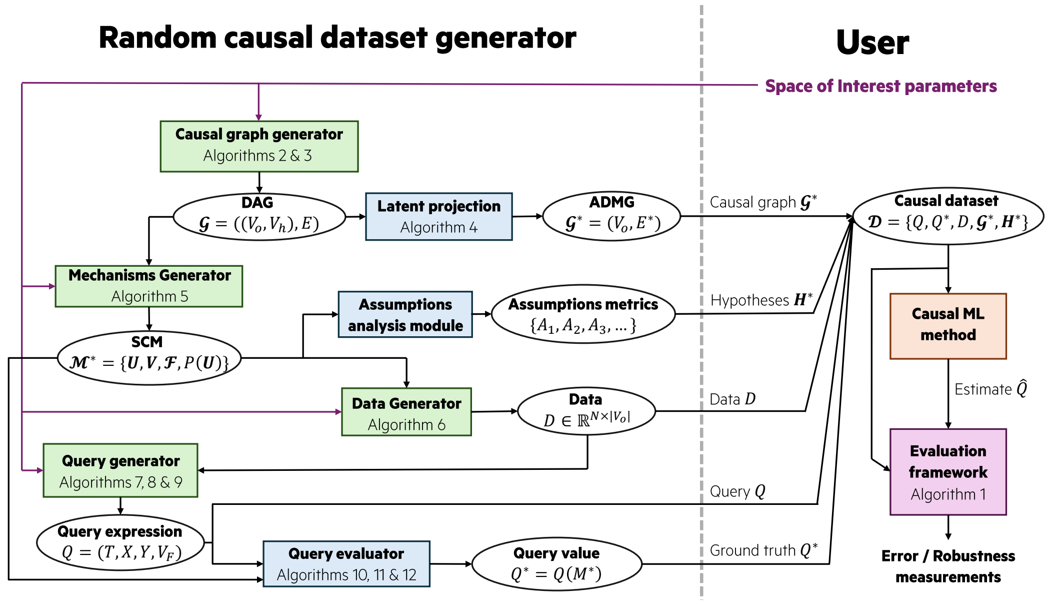

Random causal dataset generator
User
Space of Interest parameters
Causal graph generator
Algorithms 2 & 3
DAG
$\mathcal{G} = ((V_o, V_h), E)$
Latent projection
Algorithm 4
ADMG
$\mathcal{G}^* = (V_o, E^*)$
Causal graph $\mathcal{G}^*$
Causal dataset
$\mathcal{D} = \{Q, Q^*, D, \mathcal{G}^*, \boldsymbol{H}^*\}$
Mechanisms Generator
Algorithm 5
Assumptions analysis module
Assumptions metrics
$\{A_1, A_2, A_3, \ldots\}$
Hypotheses $\boldsymbol{H}^*$
Causal ML method
SCM
$\mathcal{M}^* = \{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\mathcal{F}}, P(\boldsymbol{U})\}$
Estimate $\hat{Q}$
Data Generator
Algorithm 6
Data
$D \in \mathbb{R}^{N \times |V_o|}$
Data D
Query generator
Algorithms 7, 8 & 9
Evaluation framework
Algorithm 1
Query Q
Query expression
$Q = (T, X, Y, V_F)$
Query evaluator
Algorithms 10, 11 & 12
Query value
$Q^* = Q(M^*)$
Ground truth Q^*
Error / Robustness measurements

Figure 13: CausalProfiler structure. The left-hand side of the figure represents the code structure of the causal dataset generator. The right-hand side represents the user code. It illustrates how CausalProfiler can be used to evaluate a Causal ML method.

J PROOF OF PROPOSITION 5.1 (COVERAGE)

This section presents the proof of Proposition 5.1 stating that: For a Space of Interest $\mathcal{S} = \{\mathbb{M}, \mathbb{Q}, \mathbb{D}\}$, whose class of SCMs is a class of Regional Discrete SCMs with the maximum number of noise regions, any causal dataset $\mathcal{D} = \{Q, Q^\star, D, \mathcal{G}^\star, \mathbf{H}^\star\}$ has a strictly positive probability to be generated.

Firstly, let us note that:

- Stating that any query Q can have any ground truth value $Q^\star$ given $\mathcal{S}$ is equivalent to saying that the class of considered SCMs, i.e., the class of Regional Discrete SCMs with the maximum number of noise regions, is $\mathcal{L}_3$-expressive with regards to the class of Markovian discrete SCMs (i.e., any $\mathcal{L}_3$-distribution of the class of Markovian discrete SCMs can be expressed with a Regional Discrete SCM).
- As the set of hypotheses $\mathbf{H}^\star$ can contain at most $\mathcal{L}_3$ conditions, if the class of considered SCMs is $\mathcal{L}_3$-expressive, then any set of hypotheses $\mathbf{H}^\star$ can be represented.
- If the class of considered SCMs is $\mathcal{L}_3$-expressive, then it is also $\mathcal{L}_1$-expressive, hence, D can be sampled from any distribution

As a result, our proof consists of showing that $P(Q, \mathcal{G}^\star|\mathcal{S}) > 0$ and that the class of Regional Discrete SCMs with the maximum number of noise regions, denoted $\mathbb{M}_{\text{RD}-\text{SCM}, r=R_{\max}}$, is $\mathcal{L}_3$-expressive with regards to the class of Markovian discrete SCMs given an *SoI* $\mathcal{S}$ and a causal graph $\mathcal{G}$.

Let us consider a *SoI* $\mathcal{S} = \{\mathbb{M}, \mathbb{Q}, \mathbb{D}\}$ with $\mathbb{M} \subseteq \mathbb{M}_{\text{RD}-\text{SCM}, r=R_{\max}}$.

Proving $P(\mathcal{G}^\star|\mathcal{S}) > 0$**:**

$\mathcal{G}^\star$ is built through Algorithm 2 as the latent projection of a DAG $\mathcal{G} = \{(\mathbf{V}_H, \mathbf{V}_O), E\}$ over $\mathbf{V}_O$ where $\mathcal{G}$ is sampled using Algorithm 1. As a result, following the steps of Algorithms 1 and 2:

$$\begin{aligned}
P(\mathcal{G}^\star|\mathcal{S}) &= P(\{(\mathbf{V}_H, \mathbf{V}_O), E\}|\mathcal{S}) \\
&= P(E|\mathbf{V})P(\mathbf{V}_H, \mathbf{V}_O|\mathcal{S}) && \text{Edges are sampled independently of the observability of the variables} \\
&= P(E|\mathbf{V})P(\mathbf{V}_H, \mathbf{V}_O|\,|\mathbf{V}|)P(|\mathbf{V}|) && |\mathbf{V}| \text{ and } p_h \text{ are the only parameters influencing the observability of the variables} \\
&= P(E|\mathbf{V})P(\mathbf{V}_H, \mathbf{V}_O|\,|\mathbf{V}|)\frac{1}{N_{\max} - N_{\min}} && |\mathbf{V}| \sim \mathcal{U}[N_{\min}, N_{\max}] \\
&= P(E|\mathbf{V})\frac{|\mathbf{V}_H|!}{|\mathbf{V}|!}\frac{1}{N_{\max} - N_{\min}} && \mathbf{V}_H \subseteq \mathbf{V} \text{ sampled without replacement} \\
&= \frac{|\mathbf{V}_H|!}{|\mathbf{V}|!(N_{\max} - N_{\min})}P(E|\mathbf{V})
\end{aligned}$$

As $E = \{V_k \to V_i \,|\, V_k \in \boldsymbol{PA}(V_i), \forall V_i \in \mathbf{V}\}$ and the edges are sampled along the causal order $[1, N]$ with probability p_{edge}:

$$\begin{aligned}
P(\mathcal{G}^\star|\mathcal{S}) &= \frac{|\mathbf{V}_H|!}{|\mathbf{V}|!(N_{\max} - N_{\min})}\prod_{i=1}^{N} P(\{V_k \to V_i \,|\, V_k \in \boldsymbol{PA}(V_i)\}) \\
&= \frac{|\mathbf{V}_H|!}{|\mathbf{V}|!(N_{\max} - N_{\min})}\prod_{i=1}^{N} {p_{edge}}^{|\boldsymbol{PA}(V_i)|}(1 - p_{edge})^{i-1-|\boldsymbol{PA}(V_i)|}
\end{aligned}$$

Let us note that $p_{edge} = 0 \implies |\boldsymbol{PA}(V_i)| = 0$ and $p_{edge} = 1 \implies |\boldsymbol{PA}(V_i)| = i - 1$. As a result, $P(\mathcal{G}^\star|\mathcal{S}) > 0$.

Proving that $\mathbb{M}_{\texttt{RD-SCM},r=R_{\max}}$ is $\mathcal{L}_3$-expressive with regards to the class of Markovian discrete SCMs: Regional discrete SCMs are, by construction, Markovian Canonical SCMs (Zhang et al., 2022). Furthermore, if the number of noise regions is chosen to be large enough (typically set to its maximum value), any Markovian Canonical SCM can be represented using a Regional Discrete SCM[8]. Thus, applying Zhang et al. (2022) Theorem 2.4, we can assert that: for an arbitrary Markovian discrete SCM, there exists a Regional Discrete SCM such that they both have the same causal graph and the same $\mathcal{L}_3$-distribution. Consequently, the class of Regional Discrete SCMs is $\mathcal{L}_3$-expressive with respect to the class of Markovian discrete SCMs given the causal graph $\mathcal{G}$. Moreover, $P(\mathcal{G}) > 0$ for all $\mathcal{G}$ because $\prod_{i=1}^{N} {p_{edge}}^{|\boldsymbol{PA}(V_i)|}(1 - p_{edge})^{i-1-|\boldsymbol{PA}(V_i)|} > 0$ (cf. previous paragraph). Thus, more generally, the class of Regional Discrete SCMs sampled by our CausalProfiler is $\mathcal{L}_3$-expressive with respect to the class of Markovian SCMs.

Proving $P(Q|\mathcal{G}^\star, \mathcal{S}) > 0$: Q is sampled given $\mathbb{Q}, D$ and $\mathcal{G}^\star$. Even though we currently only implement queries sampling for the classes $\mathcal{Q}_{\text{ATE}}$, $\mathcal{Q}_{\text{CATE}}$ and $\mathcal{Q}_{\text{Ctf-TE}}$ (cf. Appendix F and Algorithms 6, 7 and 8), we can generalize our proof to any other query class (e.g., CDE, NDE). We simply assume that these classes translate the set of constraints on the variables under consideration (e.g., conditioning variables have to be distinct from treatment variables or any other graphical constraints that can be checked with $\mathcal{G}^\star$) and express the probabilistic causal formula to be estimated. Once such a query class $\mathbb{Q}$ is defined, our method randomly samples variables from $\mathbf{V}_O$ in accordance with $\mathbb{Q}$ constraints and by sampling realizations from D. We showed in the previous paragraph that $\mathbb{M}_{\texttt{RD-SCM},r=R_{\max}}$ is $\mathcal{L}_3$-expressive implying that it is $\mathcal{L}_1$-expressive too. So, any realization can be present in D. As a result, for a given query class $\mathbb{Q}$, any Q can be generated. Hence, $P(Q|\mathcal{G}^\star, \mathcal{S}) > 0$.

[8] The distinction between $\mathbf{V}_O$ and $\mathbf{V}_H$ is of no importance for $\mathcal{L}_3$-expressiveness.$\mathbf{V}_O$ and $\mathbf{V}_H$ are only used to determine what will be visible to the user as benchmark.

Proving Proposition 5.1 by combining previous results: We proved that $\mathbb{M}_{\texttt{RD-SCM},r=R_{\max}}$ is $\mathcal{L}_3$-expressive, hence any training set D, ground truth query $Q^\star$ and set of hypotheses $\mathbf{H}^\star$ can be generated given an *SoI* $\mathcal{S}$, a causal graph $\mathcal{G}$ and a causal query Q. In addition, $P(Q, \mathcal{G}^\star|\mathcal{S}) = P(Q|\mathcal{G}^\star, \mathcal{S})P(\mathcal{G}^\star|\mathcal{S})$ and we also prove that $P(Q|\mathcal{G}^\star, \mathcal{S}) > 0$ and $P(\mathcal{G}^\star|\mathcal{S}) > 0$. Hence, $P(Q, \mathcal{G}^\star|\mathcal{S}) > 0$. As a result, any causal dataset $\mathcal{D}$ has a strictly positive probability to be generated.

Remark on continuous SCMs. The universal approximation theorem (Hornik, 1991) states that NNs (with non-polynomial activation functions) are dense in the space of continuous functions, meaning that any continuous function can be approximated by a sequence of NNs converging to this function. However, this does not guarantee that they strictly cover the space of continuous functions. In particular, whenever the number of layers and neurons is finite, one can always build a continuous function too complex to be represented with this finite number of parameters. Hence, Proposition 5.1 cannot be extended to any class of continuous SCMs. However, it could potentially be adapted not to ask for strict coverage but rather density. We leave this question for future work.

K Verification Results

We design and run verification experiments targeting each level of the PCH.

All following experiments are done on discrete SCMs to reduce approximations. Indeed, distributions over continuous variables can only be approximated (e.g., using kernel methods) while discrete ones can be computed exactly. In addition, the experiments rely on conditional independence testing, which has been proven to be particularly difficult to use with continuous variables. Indeed, (Shah & Peters, 2020) proved that no conditional independence test with a continuous conditioning variable can have both a valid significance level and power.

K.1 $\mathcal{L}_1$ verification

Consistency with $\mathcal{L}_1$ level of the PCH is tested through the verification that the Markov property holds on randomly sampled regional discrete SCMs. Below is a description of the experimental design choices made and the associated results.

K.1.1 Experiment

For a given SCM $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$, we check that the Markov property is satisfied by assessing whether there is a statistically significant amount of d-separations not leading to conditional independence in the entailed distribution.
To do so, we first enumerate the list of sets of variables $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ in $\mathbf{V}$ corresponding to d-separations in $\mathcal{M}$'s causal graph $\mathcal{G}_\mathcal{M}$, ie $\mathbf{A} \perp\!\!\!\perp_{\mathcal{G}_\mathcal{M}} \mathbf{B}|\mathbf{C}$. Second, for each d-separated set $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, we test whether $\mathbf{A} \perp\!\!\!\perp_{P_\mathcal{M}} \mathbf{B}|\mathbf{C}$ by sampling 50k data points from the entailed distribution $P_\mathcal{M}$.

In practice, enumerating all the d-separations can be very costly. Moreover, as the set of variables $\mathbf{C}$ increases, it becomes increasingly complicated to robustly test the conditional independence $\mathbf{A} \perp\!\!\!\perp_{P_\mathcal{M}} \mathbf{B}|\mathbf{C}$. Indeed, as the cardinality of $\mathbf{C}$ increases, so does the number of combinations of values for which to test independence between variables $\mathbf{A}$ and $\mathbf{B}$. Running the statistical test becomes costly, and the data volume required for robust independence test results increases exponentially. This is why we limit ourselves to listing the d-separated sets $(A, B, \mathbf{C})$ such that $A \in \mathbf{V}$, $B \in \mathbf{V} \backslash A$, and $C \in \mathbf{V} \cup \mathbf{V}^2 \cup \mathbf{V}^3$ by enumerating all the possible $(A, B, \mathbf{C})$ tuples, and testing whether they are d-separated in $\mathcal{G}_\mathcal{M}$.

As the sampled SCMs are regional discrete, the conditional independence $A \perp\!\!\!\perp_{P_\mathcal{M}} B|\mathbf{C}$ can be tested with Pearson's χ^2 independence tests (Pearson, 1900). More precisely, A and B are considered independent conditionally to $\mathbf{C}$ if for all values $\mathbf{c}$ of $\mathbf{C}$, the H_0 hypothesis "A and B are independent" is not rejected. Since Pearson's χ^2 test is based on the assumption that the number of samples is large, we decide to skip tests where the Koehler criterion (Koehler & Larntz, 1980) is not met. Based on empirical analyses, this criterion indicates whether the χ^2 test is reliable depending on the number of samples considered. In addition, as we conduct tests for each observed value c, we need to control for the expected proportion of false positives (represented by the Type I error of the test). To do so, we apply the Benjamini-Hochberg correction (Benjamini & Hochberg, 1995).

For each *SoI*, defined by the Cartesian product of the following parameters, we sample 5 SCMs:

- **Number of endogenous variables**: $\{4, 5, 6\}$
- **Expected edge probability**: $\{0.1, 0.4\}$
- **Proportion of unobserved endogenous variables**: set to 0 because the Markov property only hold for Markovian SCMs
- **Number of noise regions**: $\{5, 10\}$
- **Cardinality of endogenous variables**: $\{2, 3, 10\}$
- **Distribution of exogenous variables**: set to $\mathcal{U}[0, 1]$
- **Number of data points**: 50000

K.1.2 RESULTS

Table 13: Conditional independence tests based on χ^2 independence tests to assess compliance of sampled SCMs with the Markov property. Results are expressed as a percentage of the total of each test type for each conditioning set size. The number of tests is also shown in brackets.

Conditioning set size	$A \perp\!\!\!\perp_{P_{\mathcal{M}}} B \mid \mathbf{C}$ **tests**				χ^2 **independence tests**			
	Total	Pass	Fail	Skip	Total	Pass	Fail	Skip
$\lvert\mathbf{C}\rvert = 1$	100 (2 391)	91.76 (2 194)	4.94 (118)	3.3 (79)	100 (9 130)	85.4 (7 797)	1.43 (131)	13.17 (1 202)
$\lvert\mathbf{C}\rvert = 2$	100 (2 986)	91.16 (2 722)	5.63 (168)	3.22 (96)	100 (53 040)	45.2 (23 976)	0.33 (177)	54.46 (28 887)
$\lvert\mathbf{C}\rvert = 3$	100 (1 693)	91.08 (1 542)	5.67 (96)	3.25 (55)	100 (145 320)	18.49 (26 874)	0.07 (106)	81.43 (118 340)
TOTAL	100 (7 070)	91.34 (6 458)	5.40 (382)	3.25 (230)	100 (207 490)	28.26 (58 647)	0.2 (414)	71.54 (148 429)

The experimental results are summarized in Table 13, where it can be seen that 5.4% of the conditional independence tests failed. Despite the use of the Koehler criterion and Benjamini-Hochberg correction, some tests can still be rejected due to the random nature of finite data sampling, which can produce slight artificial correlations in the data. Moreover, on closer inspection, the majority of the failed tests (at least 350 out of 382)[9] are unsuccessful because of a single failed χ^2 independence test. This reinforces our previous argument about the random nature of finite data sampling.

One can also notice that the number of skipped χ^2 independence tests increases with the size of the conditioning set. Such behavior is to be expected, since the number of realizations of the conditioning set increases exponentially with its cardinality, while the number of observations sampled to perform the independence tests remains constant. As a result, there are fewer and fewer observations available to perform each χ^2 test. In contrast, the number of fully skipped conditional independence tests remains constant. This means that the χ^2 skipped tests are relatively homogeneously distributed across all the conditional independence tests.

Someone might argue that the number of sampled observations should simply be automatically computed to verify the Koehler criterion. However, in general, such a calculation is complicated, if not impossible, to automate, as causal mechanisms are randomly sampled. As a result, all kinds of observational distributions can be induced with potentially very low probability realizations, for which the Koehler criterion could never be validated because the number of data to be sampled would be too large.

[9] Indeed, there is a total of 414 χ^2 tests that failed corresponding to 382 failed conditional independence tests. It mean that, at most 32(=414-382) conditional independence tests can have more than one failed χ^2 independence test.

To conclude, these results are sufficient to conclude that the Markov property is empirically verified by the sampled SCMs.

K.2 $\mathcal{L}_2$ VERIFICATION

Consistency with $\mathcal{L}_2$ level of the PCH is tested through the verification that the Do-calculus rules hold on randomly sampled regional discrete SCMs. Below is a description of the experimental design choices made (Appendix K.2.1) and the associated results (Appendix K.2.2).

K.2.1 EXPERIMENT

Definition K.1. Do-Calculus rules (Pearl, 2009)
Given an SCM $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$ whose causal graph $\mathcal{G}$ is a DAG, and disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, and $\mathbf{W}$ of $\mathbf{V}$, the rules of the **Do-Calculus** are defined as follows:

1. **Insertion/deletion of observation**: if $\mathbf{Y}$ and $\mathbf{Z}$ are d-separated by $\mathbf{X} \cup \mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{X}}}$, then $P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \mathbf{W}, \mathbf{Z}) = P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \mathbf{W})$
2. **Action/observation exchange**: if $\mathbf{Y}$ and $\mathbf{Z}$ are d-separated by $\mathbf{X} \cup \mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{X}},\underline{\mathbf{Z}}}$, then $P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \boldsymbol{do}(\mathbf{Z} = \mathbf{z}), \mathbf{W}) = P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \mathbf{Z}, \mathbf{W})$
3. **Insertion/deletion of action**: if $\mathbf{Y}$ and $\mathbf{Z}$ are d-separated by $\mathbf{X} \cup \mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{X}},\overline{\mathbf{Z}(\mathbf{W})}}$, then $P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \boldsymbol{do}(\mathbf{Z} = \mathbf{z}), \mathbf{W}) = P(\mathbf{Y}|\boldsymbol{do}(\mathbf{X} = \mathbf{x}), \mathbf{W})$

where $\mathcal{G}_{\overline{\mathbf{X}}}$ (resp. $\mathcal{G}_{\underline{\mathbf{X}}}$) represents the graph $\mathcal{G}$ where the incoming edges in (resp. outgoing edges from) $\mathbf{X}$ have been removed and $\mathbf{Z}(\mathbf{W})$ is the subset of nodes in $\mathbf{Z}$ that are not ancestors of any node in $\mathbf{W}$ in $\mathcal{G}_{\overline{\mathbf{X}}}$

For a given SCM, we check each rule by first enumerating the sets of d-separated variables of interest. Second, for each d-separated set, we test whether the distributions are statistically significantly similar by sampling 50k data points from the intervened SCMs and testing whether they are drawn from the same distribution.

For the same computational cost reasons as for $\mathcal{L}_1$ verification, we consider only univariate sets of variables X, Y, Z, and W. In addition, the studied SCMs are sampled from the same *SoIs* as defined in the $\mathcal{L}_1$-verification experiment (Appendix K.1.1). Finally, to assess whether two conditional distributions are identical, we used Pearson's χ^2 goodness of fit tests (Pearson, 1900). As done in Section K.1, we also use the Koehler criterion (Koehler & Larntz, 1980) and the Benjamini-Hochberg correction (Benjamini & Hochberg, 1995).

For each *SoI*, defined by the Cartesian product of the following parameters, we sample 2 SCMs:

- **Number of endogenous variables**: $\{4, 5, 6\}$
- **Expected edge probability**: $\{0.1, 0.4\}$
- **Proportion of unobserved endogenous variables**: set to 0 because the Markov property only hold for Markovian SCMs
- **Number of noise regions**: $\{5, 100\}$
- **Cardinality of endogenous variables**: $\{2, 5\}$
- **Distribution of exogenous variables**: set to $\mathcal{U}[0, 1]$
- **Number of data points**: 50000

Compared to the previous experiment (Appendix K.1.1), we reduce the number of sampled SCMs because comparing distributions two by two is more computationally expensive than conditional independence tests.

Table 14: Conditional independence tests based on χ^2 goodness of fit tests to assess compliance of sampled SCMs with the Do-Calculus rules. Results are expressed as a percentage of the total of each test type for each conditioning set size. The number of tests is also shown in brackets.

	Cond. goodness of fit				**χ^2 goodness of fit**			
Do-Calculus Rule	Total	Pass	Fail	Skip	Total	Pass	Fail	Skip
Rule 1 Insertion/deletion of observation	100 (3 378)	96.15 (3 248)	3.85 (130)	0 (0)	100 (171 092)	88.84 (152 004)	0.1 (172)	11.06 (18 916)
Rule 2 Action/observation exchange	100 (5 065)	94.04 (4 763)	5.96 (302)	0 (0)	100 (259 509)	83.84 (217 578)	0.09 (241)	16.06 (41 690)
Rule 3 Insertion/deletion of action	100 (5 169)	93.75 (4 846)	6.25 (323)	0 (0)	100 (282 184)	89.21 (251 731)	0.06 (157)	10.74 (30 296)
TOTAL	100 (13 612)	94.45 (12 857)	5.55 (755)	0 (0)	100 (712 785)	87.17 (621 313)	0.08 (570)	12.75 (90 902)

K.2.2 RESULTS

The experimental results are summarized in Table 14 where it can be seen that they are very similar to the $\mathcal{L}_1$ verification ones: roughly 6% of the conditional goodness of fit tests were not validated, some tests are rejected due to the random nature of finite data sampling but the majority them (at least 570 out of 755) are unsuccessful because of a single failed χ^2 goodness of fit test.

One can also notice that the percentage of skipped χ^2 goodness of fit tests is similar for rules 1 and 3 but increases by roughly 50% for rule 2. Such behavior is to be expected as rule 2 is the only rule to have conditioning sets of size 3 on both sides of the equality. However, the number of skipped tests remains low, with a maximum of 16%.

As a result, we estimate that these results are sufficient to conclude that the Do-calculus rules are respected by the sampled SCMs.

K.3 $\mathcal{L}_3$ VERIFICATION

Consistency with $\mathcal{L}_3$ level of the PCH is tested through the verification that the axiomatic characterization of structural counterfactuals holds on randomly sampled regional discrete SCMs. Below is a description of the experimental design choices made (Appendix K.3.1) and the associated results (Appendix K.3.2).

Definition K.2. Axiomatic characterization of structural counterfactuals (Pearl, 2009) Given an SCM $\mathcal{M} := \{\mathbf{V}, \mathbf{U}, \boldsymbol{\mathcal{F}}, P(\mathbf{U})\}$ whose causal graph $\mathcal{G}$ is a DAG, the **axioms of structural counterfactuals** are defined as follows:

1. **Composition**: For any sets of endogenous variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{W}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$, if $\mathbf{W}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}) = \mathbf{w}$ then $\mathbf{Y}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(\mathbf{W}=\mathbf{w})}(\mathbf{u}) = \mathbf{Y}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u})$
2. **Effectiveness**: For any disjoint sets of endogenous variables $\mathbf{X}$, and $\mathbf{W}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$, $\mathbf{X}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(\mathbf{W}=\mathbf{w})}(\mathbf{u}) = \mathbf{x}$
3. **Reversibility**: For any two distinct variables Y and W and any sets of other variables $\mathbf{X}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$, if $Y_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(W=w)}(\mathbf{u}) = y$ and $W_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(Y=y)}(\mathbf{u}) = w$ then $Y_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}) = y$

Note that we do not write $P(\mathbf{W}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}|\mathbf{U})$ but rather $\mathbf{W}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u})$ as it is a deterministic expression. Indeed, if $\mathbf{U}$ is fixed, there is no stochastically anymore, so we no longer need to reason in distributions but rather in functional forms.

K.3.1 EXPERIMENT

For a given SCM, using Definition K.1 notations, we check that:

1. The **Composition** axiom is satisfied by assessing whether $\mathbf{W}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}) = \mathbf{w}$ implies $\mathbf{Y}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(\mathbf{W}=\mathbf{w})}(\mathbf{u}) = \mathbf{Y}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u})$ for any sets of endogenous variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{W}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$
2. The **Effectiveness** axiom is satisfied by assessing whether $\mathbf{X}_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(\mathbf{W}=\mathbf{w})}(\mathbf{u}) = \mathbf{x}$ for any sets of endogenous variables $\mathbf{X}$, and $\mathbf{W}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$
3. The **Reversibility** axiom is satisfied by assessing whether $Y_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(W=w)}(\mathbf{u}) = y$ and $W_{\boldsymbol{do}(\mathbf{X}=\mathbf{x}),\boldsymbol{do}(Y=y)}(\mathbf{u}) = w$ implies $Y_{\boldsymbol{do}(\mathbf{X}=\mathbf{x})}(\mathbf{u}) = y$ for any two (distinct) variables Y and W and any sets of variables $\mathbf{X}$ in $\mathbf{V}$ and any realization $\mathbf{u}$ of $\mathbf{U}$

For each *SoI*, defined by the Cartesian product of the following parameters, we sample 5 SCMs:

- **Number of endogenous variables**: $\{3, 5, 10\}$
- **Expected edge probability**: $\{0.1, 0.5, 0.7\}$
- **Proportion of unobserved endogenous variables**: set to 0 because the Markov property only hold for Markovian SCMs
- **Number of noise regions**: $\{3, 5, 10\}$
- **Cardinality of endogenous variables**: $\{2, 5, 7\}$
- **Distribution of exogenous variables**: set to $\mathcal{U}[0, 1]$
- **Number of data points**: 50000

For each SCM, instead of enumerating all the possible four sets of variables $\mathbf{X}, \mathbf{Y}$ and $\mathbf{W}$, we sample a partition of three elements of a randomly sampled subset of $\mathbf{V}$ of a size randomly picked in $[3, |\mathbf{V}|]$. This sampling strategy enables us to make sure the three sets are disjoint and of randomly varying size. In addition, for each four sets, we sample 50k realizations of $\mathbf{U}$.

Let us note that the axioms now correspond to exact realizations and not equal probabilities. As a result, we expect no failure as no approximation is made in this experiment.

K.3.2 Results

As expected, all the tested equalities are verified in our experiments. We can, therefore, consider that the SCMs created by our generator allows the estimation of any structural counterfactual queries.

L Extended Experimental Results

This appendix complements Section 6 with extended setup details and results. We first provide further details for Experiments 1 and 2, inlcuding the Algorithm 12 describing our evaluation protocol. We then include an additional experiment on ATE estimation under hidden confounding, and then an evaluation of runtime scalability on larger graphs.

L.1 Experiment 1: Additional Information

Table 15 details the *SoI* used in our experiments, Table 16 reports extended performance metrics complementing Table 1, and Figure 14 shows box plots of ATE estimation errors.

Parameters not explicitly listed for a given *SoI* are set to their default values as per the benchmark configuration. Neural Networks for our experiments have two 8-neuron layers and use ReLU activation. Unless otherwise specified, we use 1000 samples per SCM in our experiments. This value was chosen as a stable default for these *SoIs* after testing several dataset sizes. More precisely, after testing the stability of the methods we evaluate (i.e., CausalNF, DCM, NCM, VACA) over the following dataset sizes, 50, 100, 200, 1000, and 2000, we found that 1000 samples was the smallest dataset size not drastically degrading the performance of the methods. This is why we decided to take this value as default for our experiments. We only vary it explicitly when studying the effect of limited data (e.g., in NN-Large-LowData).

Algorithm 12 Evaluation process for causal machine learning methods

```
Input: List of Spaces of Interest $SoIs$, list of seeds $seeds$ number of examples per SCM $num\_examples$
Initialize: $method \leftarrow$ CausalMLMethod()
for each $SoI$ in $SoIs$ do
    for each $seed$ in $seeds$ do
        setGlobalSeed(seed)
        for each $examples$ in $num\_examples$ do
            Generate samples, queries, and targets from the profiler
            Get estimates using the $method$ on the generated samples and queries
            Calculate (and store) error by comparing estimates with targets
        end for
        Compute performance statistics for seed
    end for
    Compute performance statistics for SoI
end for
Output: Final summary with evaluation results
```

Table 15: Specification of each *SoI* used in the general experiments. N denotes the sampled number of nodes.

Name	`Linear-Medium`
# Nodes	15-20
Mechanism	Linear
Expected Edges	$2 \times N$
Variable Type	Continuous
Samples	1000
Query Type	ATE
Seeds	[10, 11, 12, 13, 14]

Name	`NN-Medium`
# Nodes	15-20
Mechanism	NN
Expected Edges	$2 \times N$
Variable Type	Continuous
Samples	1000
Query Type	ATE
Seeds	[10, 11, 12, 13, 14]

Name	`NN-Large`
# Nodes	20-25
Mechanism	NN
Expected Edges	$2 \times N$
Variable Type	Continuous
Samples	1000
Query Type	ATE
Seeds	[10, 11, 12, 13, 14]

Name	`NN-Large-LowData`
# Nodes	20-25
Mechanism	NN
Expected Edges	$2 \times N$
Variable Type	Continuous
Samples	50
Query Type	ATE
Seeds	[10, 11, 12, 13, 14]

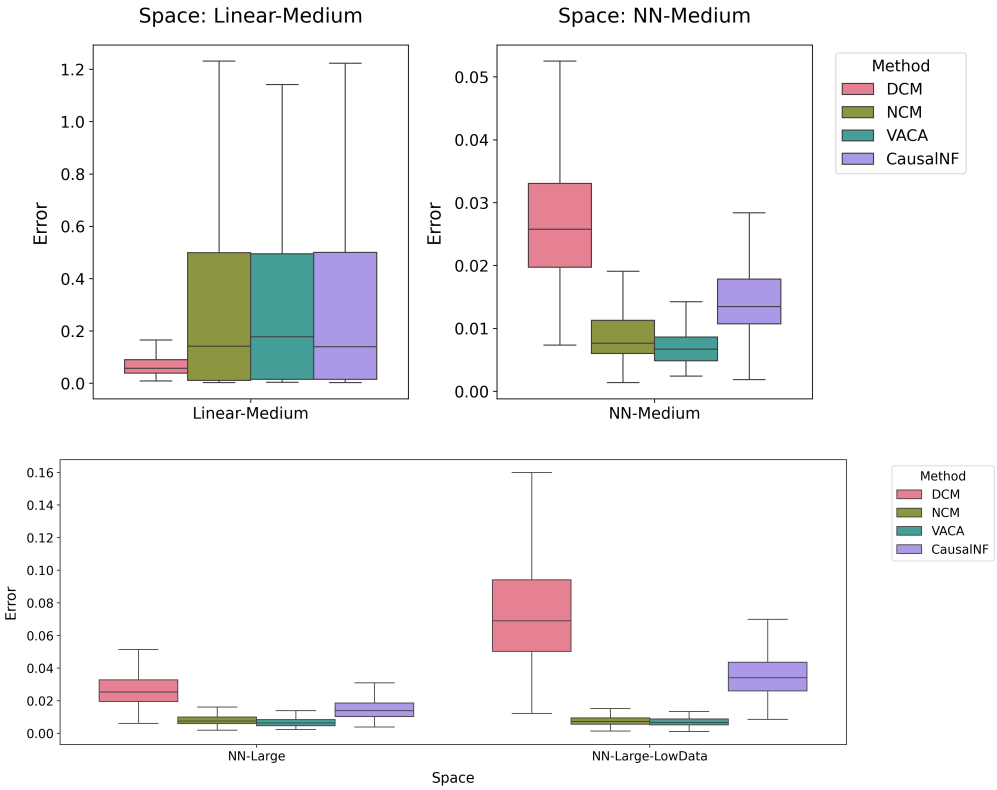


Figure 14: Box plots showing ATE estimation errors across different *SoIs*

Table 16: Additional performance metrics of CausalNF, DCM, NCM, and VACA on the general experiments.

Space	Method	Min Error	Total Fail	Runtime Mean	Runtime Std
Linear-Medium	CausalNF	0.0024	0	27.58 s	18.33 s
	DCM	0.0086	0	33.08 s	9.71 s
	NCM	0.0024	0	14.77 s	1.42 s
	VACA	0.0038	1335	11.69 s	4.54 s
NN-Medium	CausalNF	0.0019	0	21.47 s	19.52 s
	DCM	0.0073	0	31.79 s	10.62 s
	NCM	0.0014	0	14.65 s	1.43 s
	VACA	0.0024	125	12.13 s	4.41 s
NN-Large	CausalNF	0.0038	0	30.23 s	25.33 s
	DCM	0.0060	0	38.33 s	14.02 s
	NCM	0.0018	0	18.90 s	1.38 s
	VACA	0.0023	290	12.88 s	4.31 s
NN-Large-LowData	CausalNF	0.0086	0	44.28 s	17.10 s
	DCM	0.0121	0	4.82 s	1.34 s
	NCM	0.0013	0	0.81 s	0.11 s
	VACA	0.0010	0	10.43 s	4.59 s

L.2 Experiment 2: Additional Information

We provide more details about the *SoI* used in our experiments in Table 17 and present extended performance metrics in Table 18, complementing those already shown in Table 2. Parameters not explicitly listed for a given *SoI* are set to their default values as per the benchmark configuration.

Table 17: Specification of the Spaces of Interest used for evaluating discrete SCMs with Ctf-TE queries. N denotes the sampled number of nodes.

Name	`Disc-C2-Reject`
# Nodes	10–15
# Categories	2
Mechanism	Tabular
Sampling Strategy	Rejection
Edges	N
Samples	500
Query Type	Ctf-TE
Seeds	[1, 2, 3, 4, 5]

Name	`Disc-C4-Unbias`
# Nodes	10–15
# Categories	4
Mechanism	Tabular
Sampling Strategy	Random
Edges	N
Samples	500
Query Type	Ctf-TE
Seeds	[1, 2, 3, 4, 5]

Name	`Disc-L-C2-Unbias`
# Nodes	20–30
# Categories	2
Mechanism	Tabular
Sampling Strategy	Random
Edges	N
Samples	500
Query Type	Ctf-TE
Seeds	[1, 2, 3, 4, 5]

Table 18: Additional performance metrics of CausalNF and DCM on the discrete experiments.

Space	Method	Min Error	Total Fail	Runtime Mean	Runtime Std
Disc-C2-Reject	CausalNF	0.0000	202	0.46 s	0.04 s
	DCM	0.0000	107	8.81 s	3.55 s
Disc-C4-Unbias	CausalNF	0.0000	1017	0.42 s	0.03 s
	DCM	0.0000	565	7.68 s	3.43 s
Disc-L-C2-Unbias	CausalNF	NaN	2500	0 s	0 s
	DCM	0.0000	283	16.39 s	6.42 s

L.3 Experiment 3: ATE Estimation under Hidden Confounding

In this experiment, we demonstrate how our framework can be used to evaluate methods in the presence of latent confounders — a common challenge in real-world causal inference. A key goal here is not only to confirm theoretical limitations but to investigate how quickly and severely performance degrades when assumptions are violated. While theory can tell us whether identification holds, it is often agnostic to the *degree* of failure. See Table 20 for a summary of results, Table 21 for a few additional performance metrics, and Figure 15 for a boxplot of ATE estimation errors over the different *SoI*.

We focus on two linear SCM settings:

- **Linear-No-Hidden:** Linear SCMs with 10-15 nodes and full observability (no hidden confounders), using 1000 data points per SCM.
- **Linear-60-Hidden:** Same setup as above, but with 60% of the variables unobserved (hidden).

We provide more details about the *SoI* used in our experiments in Table 19. Parameters not explicitly listed for a given *SoI* are set to their default values as per the benchmark configuration.

Table 19: Specification of the *SoIs* used to evaluate performance under hidden confounding. N denotes the sampled number of nodes.

Name	`Linear-No-Hidden`
# Nodes	10-15
Mechanism	Linear
Expected Edges	$2 \times N$
Variable Type	Continuous
Prop. Hidden Nodes	0%
Samples	1000
Query Type	ATE
Seeds	[42, 43, 44, 45, 46]

Name	`Linear-60-Hidden`
# Nodes	10-15
Mechanism	Linear
Expected Edges	$2 \times N$
Variable Type	Continuous
Prop. Hidden Nodes	60%
Samples	1000
Query Type	ATE
Seeds	[42, 43, 44, 45, 46]

Setup. We evaluate three methods: CausalNF (Javaloy et al., 2023), DCM (Chao et al., 2023), and DeCaFlow (Almodóvar et al., 2025). The first two methods assume causal sufficiency, and therefore cannot, in theory, handle hidden confounding. DeCaFlow, in contrast, is explicitly designed for this setting but requires access to the full causal graph (including hidden variables) and does not run when all variables are observed. Thus, we include it only in the hidden confounding *SoI*.

Results (Linear-No-Hidden). As expected, both CausalNF and DCM perform well when all variables are observed. DCM achieves lower mean error (0.0845) and standard deviation (0.1515), with a maximum error of 2.89. The upper whisker of DCM's box plot lies below the median of CausalNF, indicating consistent superior performance. These results serve as a reference point for comparison when introducing hidden variables.

Results (Linear-60-Hidden). With 60% of variables hidden, method performance degrades significantly. DeCaFlow performs reliably, with an error mean of 0.3405 and low variance. In contrast, CausalNF—despite a box plot that visually appears well-behaved—has a massive error mean of 2.67×10^{12} and a maximum error exceeding 10^{15}. This is due to a small subset of SCMs producing extremely large errors (14 with error > 1000), illustrating that, when assumptions are violated, error can become arbitrarily large. While DCM does not show such instability on this particular sample, its theoretical limitations under hidden confounding still hold — the expectation is that if we evaluate over enough SCMs we will eventually also get arbitrarily large errors due to the violation of the causal sufficiency assumption.

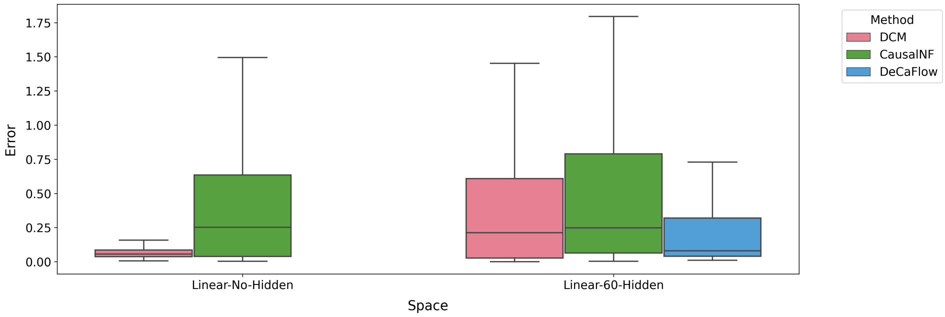


Figure 15: Box plots of ATE estimation errors in the presence and absence of hidden confounding. Each box shows the interquartile range and median, with whiskers extending to 1.5× IQR. CausalNF and DCM are shown for both *SoIs*; DeCaFlow is shown only for the hidden setting.

Table 20: Performance summary of CausalNF, DCM, and DeCaFlow on the hidden confounder experiments.

Space	Method	Mean Error	Std Error	Max Error	Runtime (s)
`Linear-No-Hidden`	CausalNF	0.5538	0.9866	14.2495	8570.0
	DCM	0.0845	0.1515	2.8954	12144.6
`Linear-60-Hidden`	CausalNF	2.667e+12	5.497e+13	1.225e+15	293.2
	DCM	0.5584	1.2122	17.2049	4187.6
	DeCaFlow	0.3405	0.6799	5.9435	2264.0

Table 21: Additional performance metrics of CausalNF, DCM, and DeCaFlow on the hidden confounder experiments.

Space	Method	Min Error	Total Fail	Runtime Mean	Runtime Std
Linear-No-Hidden	CausalNF	0.0036	0	17.14 s	10.61 s
	DCM	0.0068	0	24.29 s	7.64 s
Linear-60-Hidden	CausalNF	0.0029	0	0.59 s	0.02 s
	DCM	0.0000	0	8.38 s	3.45 s
	DeCaFlow	0.0108	0	4.53 s	1.27 s

L.4 Time and Space Complexity of Experiment 1

We provide a time and space complexity analysis based on a setting consistent with the experimental setup of Experiment 1. Assume a continuous SCM where each mechanism is modeled as a 2-layer neural network (with hidden size 8), the dimensionality of each variable is 1, the expected number of edges scales linearly with the number of variables, and the queries are ATE. More details on the exact SoI can be found in Appendix L.1.

Parameters: V = Number of variables, E = Expected number of edges per variable, N = Number of samples, Q = Number of queries.

Time Complexity.

- **Graph generation:** $O(V^2)$
- **NN initialization (per variable):** $O(E)$
- **NN inference (per variable, per sample):** $O(E)$
- **Sample generation:**
  - For each of the N samples, we:
    - sample noise
    - run a topological sort (once)
    - evaluate each of the V variables via a forward pass through a neural network with on average E inputs, so the cost per sample is $O(V \cdot E)$
  - Hence, in total $O(N \cdot V \cdot E)$.
- **Query generation and evaluation:** $O(Q \cdot N \cdot V \cdot E)$
- **Overall dominant term (worst case):** $O(Q \cdot N \cdot V \cdot E)$

This scaling is intuitive: each query requires N samples, where each sample involves computing all V variables, and each variable depends on approximately E parents through a neural network. Note that in this setting the $O(V^2)$ graph-generation term is always dominated, since $VE = \Theta(V^2)$.

In practice, we get constant-time speedups using vectorized operations and batch processing, e.g., we do not have a loop over samples but process them by batch.

Space Complexity.

- **Graph structure:** $O(V + E)$
- **NN parameters:** $O(V \cdot E)$
- **Sample storage:** $O(N \cdot V)$
- **Query outputs:** $O(Q)$, working memory to compute a single query: $O(Q \cdot V \cdot N)$
- **Total:** $O(V \cdot E + N \cdot V + Q)$

L.5 Runtime Scalability on Larger Graphs

We additionally evaluate the scalability of CausalProfiler with respect to the number of variables. Batch processing and vectorized operations enable efficient dataset generation even for graphs with hundreds of variables. Table 22 reports the average generation time (over 5 runs) for producing $10{,}000$ samples and 50 queries (each estimated using $10{,}000$ additional datapoints), using the same CPU hardware described in Section 6.3.

Table 22: Average runtime (seconds) of CausalProfiler for generating datasets across increasing numbers of variables. Each value is the mean over 5 runs with standard deviation in parentheses.

Num Variables	Mean Time (s)	Std Dev (s)
10	0.19	0.01
50	0.89	0.03
100	1.81	0.03
500	9.61	0.11
1000	19.24	0.21

For completeness, Table 23 reports the runtime of each evaluated method in Experiment 1 (Section 6.4) on the *NN-Large SoI* as the number of nodes increases (with the expected number of edges fixed to N, the number of nodes). While some methods scale better than others, dataset generation with the CausalProfiler remains efficient.

Table 23: Runtime scaling of causal inference methods (in seconds). Each entry reports mean and standard deviation across runs.

Node Range	CausalNF	DCM	NCM	VACA
30–40	(1, 0.6)	(24, 8.4)	(12, 1.2)	(11, 4.6)
50–70	(2, 0.4)	(43, 12.8)	(22, 2.6)	(12, 4.8)
70–90	(3, 0.3)	(53, 18.0)	(29, 2.4)	(12, 4.7)
90–110	(4, 0.4)	(60, 23.0)	(36, 2.5)	(9, 2.5)

The apparent reduction in average VACA runtime is explained by its increasing failure rate. All other methods exhibit a 0% failure rate.