Graph Distance Based on Cause-Effect Estimands with Latents

Zhufeng Li **Niki Kilbertus**
Technical University of Munich
Helmholtz Munich
Munich Center for Machine Learning (MCML)

Abstract

Causal discovery aims to recover graphs that represent causal relations among given variables from observations, and new methods are constantly being proposed. Increasingly, the community raises questions about how much progress is made, because properly evaluating discovered graphs remains notoriously difficult, particularly under latent confounding. We propose a graph distance measure for acyclic directed mixed graphs (ADMGs) based on the downstream task of cause-effect estimation under unobserved confounding. Our approach uses identification via fixing and a symbolic verifier to quantify how graph differences distort cause-effect estimands for different treatment-outcome pairs. We analyze the behavior of the measure under different graph perturbations and demonstrate how it complements existing distance metrics.

1 Introduction

Motivation. Uncovering causal relationships from observational data, often called causal discovery or structure learning, is a longstanding goal within causal inference. In particular, causal discovery is a well-defined specific task within the structural causal model (SCM) framework popularized by Pearl [Pearl, 2000]. There, the assumed data generating process, the SCM, entails an observational distribution as well as a directed acyclic graph (DAG) over the observed variables. The task is then (i) to decide whether the graph, or which parts of it, can be uniquely identified given the observational distribution (this is called *identifiability*), and (ii) to devise algorithms to actually estimate a graph from finite observations. Both parts have received considerable attention from the scientific community [Heinze-Deml et al., 2018, Glymour et al., 2019, Zanga et al., 2022, Vowels et al., 2022]. While many methods are being proposed—leveraging automatic differentiation (e.g., Zheng et al., 2018), variational inference (e.g., Annadani et al., 2021), large scale pre-training (e.g., Lorch et al., 2022), or supporting causal discovery with large language models—a common dictum in the community is that "causal discovery is still awaiting its ImageNet moment."

Two key missing aspects in this comparison to computer vision are (i) generally accepted benchmark datasets and (ii) meaningful metrics to measure success [Hardt, 2025]. Regarding (i), currently, each proposed method typically comes with its own evaluation on simulated data. These are mostly synthetic and sometimes supplemented with isolated semi-synthetic case studies or evaluations on virtually the only "gold standard" real-world dataset the community has agreed on, a protein-signaling network published 20 years ago [Sachs et al., 2005]. Moreover, there are growing concerns about how meaningful and generalizable reported results for causal discovery are when they are based on simulated graphs and data [Reisach et al., 2021]. In recognition of these shortcomings, researchers have, for example, built platforms to benchmark methods (e.g., `https://causeme.uv.es`, Rios et al., 2025), compiled candidate benchmark datasets based on real data [Chevalley et al., 2025], and

Code: `https://github.com/zhufengli/fixing_identification_distance`

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

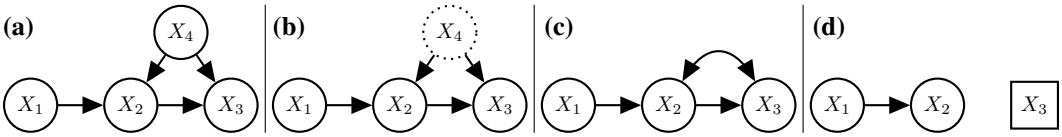

Figure 1: Example graphs: **(a)** A DAG without latents. **(b)** A DAG with latent variable X_4. **(c)** An ADMG, which is the latent projection of the graph in (b). **(d)** The CADMG resulting from fixing node X_3 in (c).

developed standardized procedures [Geffner et al., 2022] or improved techniques [Herman et al., 2025] for generating synthetic data. There are even physical machines with supposedly known causal structure from which measurements can be taken for this purpose [Gamella et al., 2024].

As for the evaluation metrics, one typically evaluates a distance or similarity measure on pairs of graphs, the true one and the estimated one. Among the most intuitive and widely used metrics is the Structural Hamming Distance (SHD) [Tsamardinos et al., 2006, Perrier et al., 2008], which simply counts the discrepancies between two graphs in terms of their edge set. While there is also ambiguity in how the SHD is applied exactly, it is unclear whether it captures "relevant" differences. This motivates measuring graph similarity by how much their implications differ on downstream tasks. While extracting a causal graph can sometimes be an end goal in itself, we argue that mostly one is ultimately interested in the strength of causal connections, or the outcomes of hypothetical experiments. These are captured by cause-effect estimation—arguably the most relevant downstream task of causal discovery [Gentzel et al., 2019].

The accurate estimation of causal effects from observational data is critical for reliable decision-making across various fields such as economics [Athey and Imbens, 2017, Imbens and Wooldridge, 2009], healthcare [Badri et al., 2009, Sanchez et al., 2022], medicine [Castro et al., 2020], and social sciences [Dehejia and Wahba, 1999, Sewell and Shah, 1968]. Cause-effect estimation typically relies on strong assumptions about the data generating process, which in the SCM framework are naturally expressed as DAGs. Different assumed DAGs yield different cause-effect estimands.

In this work, we introduce the *Fixing Identification Distance (FID)*, a distance defined on acyclic directed mixed graphs (ADMGs) based on how much graphs with latent variables agree in terms of their implied cause-effect estimands. That is, FID compares graphs through the fixing-based identifying functionals they entail for causal queries $p(y \mid do(t))$. Importantly, this comparison is graph-only: it does not require choosing a parametric model or a particular estimator for causal effects. We defer a discussion of a distribution-dependent numerical evaluation and its disadvantages for a graph-only distance to Appendix D.3. This avoids conflating structural errors with estimator variance, and isolates discrepancies that are intrinsic to the assumed causal graph. Accommodating ADMGs (not just DAGs) is relevant because incorrect or incomplete specification of latent structures can severely bias cause-effect estimation, potentially resulting in misguided interventions.

We highlight that our distance measure is not only useful to evaluate causal discovery algorithms in settings where the true causal graph is known, but perhaps even more so to assess the sensitivity of downstream cause-effect estimation to different possible graphs more broadly. Many algorithms for cause-effect estimation require a causal graph as input. Since causal graphs are generally not known, the literature often alludes to either causal discovery algorithms or domain experts to provide these graphs. Both are error-prone, and different algorithms/experts may give different answers. Understanding these differences helps researchers evaluate the robustness of causal conclusions to structural assumptions and supports transparent reporting of causal inference results.

Related Work. A related, but orthogonal body of work focuses on evaluating causal discovery algorithms when the ground truth graph is not or only partially known [Viinikka et al., 2018, Eigenmann et al., 2020, Peyrard and West, 2021, Wildberger et al., 2023]. These methods are based on empirical validation in that they rely on data to evaluate candidate graphs.

Instead, our graph distance is graphical, i.e., it takes as input only the graphs and no data. Among purely graph-based distance measures, the Structural Hamming Distance (SHD) and variants thereof [Tsamardinos et al., 2006, Perrier et al., 2008, de Jongh and Druzdzel, 2009] are most common and simply count how many pairs of nodes have different connections. The Structural Intervention Distance (SID) [Peters and Bühlmann, 2015] essentially counts the number of intervention distributions that would be falsely inferred under a candidate graph with respect to a reference graph. Improving upon and extending SID, Henckel et al. [2024] propose the Adjustment Identification Distance

(AID), measuring how many pairs of variables the two graphs would yield different cause-effect estimands derived from a given adjustment strategy. Their AID applies not only to DAGs, but also to completed partially directed acyclic graphs (CPDAGs), which represent Markov equivalence classes of DAGs. Most recently, Wahl and Runge [2025] proposed a Separation-based Distance (SD) that quantifies whether a separating set between two variables in one graph remains a separating set between the same variables in the other graph. This separation-based measure applies specifically to DAGs and maximal ancestral graphs (MAGs) [Richardson and Spirtes, 2002], but can in principle be extended to other types of graphs with corresponding separation notions.

While FID addresses the same goal as these task-aligned distances, it is not limited to the fully observed setting and allows for unobserved confounding. This fundamentally changes the identification object: in ADMGs, causal effects generally do not reduce to adjustment, but may admit many different identifying representations. We thus introduce the first task-aligned (cause-effect estimation) graph distance for graphs with latent variables.

2 Background

In this work, we assume basic familiarity with structural causal models [Pearl, 2000, Peters et al., 2017], including the notions of interventions, Markov equivalence classes, CPDAGs, and valid adjustment sets [Shpitser et al., 2012]. We make use of nested Markov models, especially how causal effects in DAGs with latents can be identified via fixing within the corresponding ADMG after projecting out the latents [Richardson et al., 2023]. We give a short introduction to identification via fixing, and refer to Richardson et al. [2023] for more details. Additional graph-theoretic notation and background are presented in Appendix A.

The "gadjid" framework. Henckel et al. [2024] propose the Adjustment Identification Distance (AID) for comparing two (CP)DAGs through their agreement on adjustment-based causal estimands. Their construction is organized around two algorithmic components.

Identifier $\mathcal{I}$ (identification strategy). Fix a particular adjustment strategy (e.g., parent-, ancestor-, or optimal-adjustment). Given a candidate graph H and an ordered pair of nodes (T, Y), the identifier either (i) returns an adjustment set (and thereby an adjustment-based estimand for $p(y \mid do(t))$), or (ii) reports that the strategy does not identify the effect.

Verifier $\mathcal{V}$. Given a reference graph G and the identifier's output for (T, Y), the verifier checks purely graphically whether the proposed adjustment set is valid in G (i.e., whether the corresponding adjustment functional identifies $p(y \mid do(t))$ under G).

AID then aggregates these binary per-pair verification outcomes over a user-chosen collection of ordered pairs (T, Y), yielding a distance that counts how often the candidate-graph-derived adjustment estimand fails to remain valid under the reference graph.

Identification with latents via fixing. While valid adjustment sets are fully characterized in the (fully observed) DAG setting via complete graphical criteria [Shpitser et al., 2012, Perković et al., 2018] with various unique choices among them, such as parent [Pearl, 2000], ancestral/minimal [Textor and Liskiewicz, 2012, Textor et al., 2016], and optimal adjustment sets [Henckel et al., 2022], allowing for unobserved confounding presents a major challenge. Tian and Pearl [2002] established a sound and complete algorithm to identify causal effects based on a DAG with latent variables commonly referred to as the "ID algorithm". However, the ID algorithm only provides a single possible identifying formula among potentially many (when the effect is identifiable). The formula returned depends on operational details such as implementation specifics and node ordering in the input. Hence, the ID algorithm alone does not suffice as a verifier: it cannot check whether a given expression remains a valid estimand in a different graph with latents (it just provides one, potentially different, valid expression). In their seminal work on nested Markov models, Richardson et al. [2023] introduced a powerful framework for identifying causal effects via "fixing operations" in the corresponding ADMGs. This method allows for a systematic enumeration of valid expressions for a given causal effect in the presence of latent variables. Intuitively, fixing a node is like "removing" its influence and the influence of its latent confounders to isolate a sub-problem where the effect is easier to see. Some terminology is required.

We denote by $G(\mathbf{V}, \mathbf{L})$ a DAG with observed nodes $\mathbf{V}$ (solid circle nodes) and latent/unobserved nodes $\mathbf{L}$ (dashed/dotted circle nodes). The *latent projection* of a DAG with latents $G(\mathbf{V}, \mathbf{L})$ is

an acyclic directed mixed graph with nodes $\mathbf{V}$ that can contain directed ($\rightarrow$) and bidirected ($\leftrightarrow$) edges—we denote it by $G(\mathbf{V})$. It is constructed from $G(\mathbf{V}, \mathbf{L})$ by including a directed edge $V_i \rightarrow V_j$ whenever there is a directed path from V_i to V_j in $G(\mathbf{V}, \mathbf{L})$ where all non-endpoints are in $\mathbf{L}$; and we include a bidirected edge $V_i \leftrightarrow V_j$ whenever there is an undirected path between V_i and V_j in $G(\mathbf{V}, \mathbf{L})$ where all non-endpoints are non-colliders in $\mathbf{L}$ and has arrowheads at both endpoints. Effectively, whenever there is a latent confounding structure between V_i and V_j, we denote this by a bidirected arrow, whereas direct influences (also mediated via latents) are still encoded via directed edges. This gives rise to semi-Markovian models, where unobserved confounding is encoded only pair-wise via bidirected edges. A *conditional acyclic directed mixed graph* (CADMG) is an ADMG $G(\mathbf{V}, \mathbf{W})$ over two disjoint sets of nodes $\mathbf{V}$, which we call *random*, and $\mathbf{W}$, which we call *fixed* (solid square nodes) such that no node in $\mathbf{W}$ has an arrowhead pointing towards it (via directed or bidirected edges). Intuitively, we think of fixed nodes as ones that have been intervened upon with a fixed, but arbitrary value, explaining why they are freed from all influences (no arrowheads towards them). Figure 1 provides examples of different graph types.

A *district* in a CADMG G is a maximal bidirected-connected component and represents groups of variables affected by common latent confounding. We write $\mathcal{D}(G)$ for the set of districts of G. We say a node $V \in \mathbf{V}$ is *fixable* in a CADMG $G(\mathbf{V}, \mathbf{W})$ if none of its descendants is in the same district as V. For a fixable node r in a CADMG $G(\mathbf{V}, \mathbf{W})$, we define the fixing operation [1]

$$\phi_r(G(\mathbf{V}, \mathbf{W})) := G^*(\mathbf{V} \backslash \{r\}, \mathbf{W} \cup \{r\})$$

where $G^*(\mathbf{V} \setminus \{r\}, \mathbf{W} \cup \{r\})$ is a CADMG with the subset of edges in $G(\mathbf{V}, \mathbf{W})$ that have no arrowheads at r. A *kernel over* $\mathbf{V}$ *indexed by* $\mathbf{W}$ is a non-negative function $q_{\mathbf{V}}(\cdot \mid \mathbf{W}) : \mathbb{R}^{|\mathbf{V}|} \rightarrow \mathbb{R}_{\geq 0}$ with $\int q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}} \mid \mathbf{x}_{\mathbf{W}})\, d\mathbf{x}_{\mathbf{V}} = 1$ for all $\mathbf{x}_{\mathbf{W}} \in \mathbb{R}^{|\mathbf{W}|}$. If a kernel $q_{\mathbf{V}}$ is Markov with respect to a CADMG $G(\mathbf{V}, \mathbf{W})$ and r is a fixable node, we define *the fixing operation with respect to a CADMG* $G(\mathbf{V}, \mathbf{W})$ via

$$\phi_r(q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}} \mid \mathbf{x}_{\mathbf{W}}); G) = \frac{q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}} \mid \mathbf{x}_{\mathbf{W}})}{q_{\mathbf{V}}(x_r \mid \mathbf{x}_{\mathrm{nd}_r^G})}\,,$$

where nd_r^G are the non-descendants of r in G. If r has no children in G, we have

$$\phi_r(q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}}|\mathbf{x}_{\mathbf{W}}); G) = \int q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}}|\mathbf{x}_{\mathbf{W}}) dx_r = q_{\mathbf{V}}(\mathbf{x}_{\mathbf{V}\backslash\{r\}}|\mathbf{x}_{\mathbf{W}})\,.$$

That is, fixing a node without children allows us to simply marginalize it out, whereas fixing a node with descendants means dividing by the kernel of that node conditioned on its non-descendants.

Richardson et al. [2023] show that fixing operations for different nodes commute for both graphs and kernels (under the nested Markov assumption) as long as the nodes remain fixable sequentially. This extends to entire *valid fixing sequences*, i.e., sequences of nodes $\mathbf{w} = (w_1, \ldots, w_k) \subseteq \mathbf{V}$ where the nodes remain sequentially fixable in the corresponding CADMGs starting from $G(\mathbf{V}, \mathbf{W})$. We then write $\phi_{\mathbf{w}}(G) := \phi_{w_k} \circ \cdots \circ \phi_{w_1}(G)$ and $\phi_{\mathbf{w}}(q_{\mathbf{V}}; G) := (\phi_{w_k} \circ \cdots \circ \phi_{w_1})(q_{\mathbf{V}}; G)$, where the final result is invariant to the order of the nodes in $\mathbf{w}$ as long as they remain a valid fixing sequence. This invariance is the key to obtaining a list of valid expressions for a causal effect, as one may obtain different expressions for different valid fixing sequences, which are guaranteed to yield the same outcome.

Next, a CADMG $G(\mathbf{V}, \mathbf{W})$ is *reachable from an ADMG* $G^*(\mathbf{V} \cup \mathbf{W})$ if there exists a valid fixing sequence $\mathbf{w}$ for $\mathbf{W}$ such that $G = \phi_{\mathbf{w}}(G^*)$ and we then also call $\mathbf{V}$ *reachable in* G^*. Finally, a set of nodes $\mathbf{S} \subseteq \mathbf{V}$ is called *intrinsic* in an ADMG $G(\mathbf{V})$ if it is a district in any reachable graph derived from $G(\mathbf{V})$—the set of intrinsic sets is $\mathrm{In}(G(\mathbf{V}))$.

We now formulate a sound and complete strategy to identify causal effects in latent variables models.

Theorem 1 (Richardson et al., 2023)**.** *For an observational distribution $p(\mathbf{x}_{\mathbf{V}})$ from a DAG model with latents $G(\mathbf{V} \cup \mathbf{L})$ and its latent projection $G(\mathbf{V})$, $T, Y \in \mathbf{V}$ different nodes, let $\mathbf{Y}^* = \mathrm{an}_Y^{G(\mathbf{V})_{\mathbf{V}\backslash\{T\}}} \cup \{Y\}$, where $\mathrm{an}_Y^{G(\mathbf{V})_{\mathbf{V}\backslash\{T\}}}$ are the ancestors of Y in the graph induced by $\mathbf{V} \setminus \{T\}$. Then if $\mathcal{D}(G(\mathbf{V})_{\mathbf{Y}^*}) \subseteq \mathrm{In}(G(\mathbf{V}))$,*

$$p(y \mid do(t)) = \int \prod_{\mathbf{D} \in \mathcal{D}(G(\mathbf{V})_{\mathbf{Y}^*})} \phi_{\mathbf{V}\backslash\mathbf{D}}(p(\mathbf{x}_{\mathbf{V}}); G)\, d\mathbf{x}_{\mathbf{Y}^*\backslash\{Y\}}\,. \tag{1}$$

[1] We follow the convention of overloading the fixing operator ϕ to operate on both kernels and graphs (CADMGs).

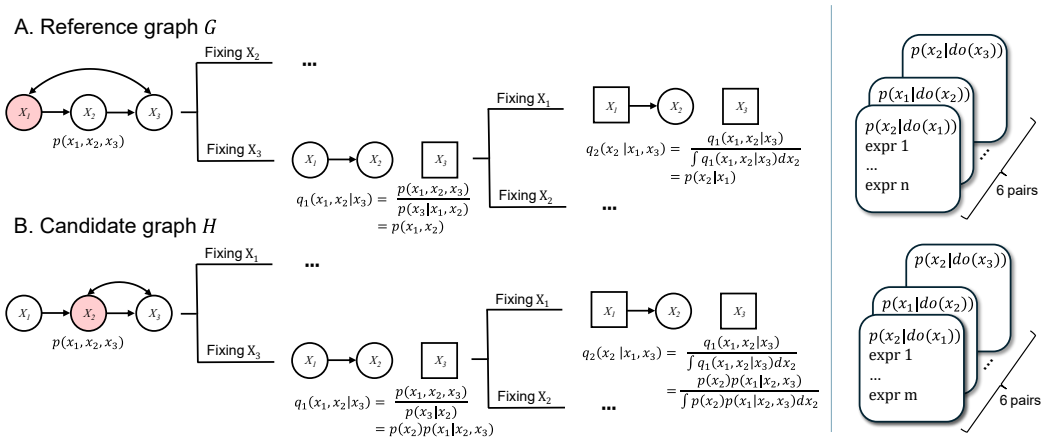

Figure 2: **Left:** Given graphs G and H and a fixed pair of nodes (x_1, x_2), identification via fixing may branch according to different valid fixing sequences and lead to different symbolic expressions for $p(x_2 \mid do(x_1))$ (e.g., `expr 1`, ..., `expr n`), where also the number of expressions can differ. The red shaded nodes indicate non-fixable variables so not all possible node sequences are valid fixing sequences. **Right:** For a set of tuples of distinct nodes, we obtain a list of symbolic expressions for the corresponding causal effect w.r.t. the two graphs. Intuitively, our FID distance then measures the aggregate/averaged overlap in the expressions for each tuple across the two graphs.

If not (i.e., there exists $\mathbf{D} \in \mathcal{D}(G(\mathbf{V})_{\mathbf{Y}^*})$ that is not intrinsic in $G(\mathbf{V})$), then $p(y \mid do(t))$ is not identifiable in $G(\mathbf{V} \cup \mathbf{L})$.

3 Methodology

We study the problem of measuring the discrepancy (or distance) between two causal graphs with hidden confounders represented as ADMGs G and H in a way that reflects how they (dis)agree on downstream cause-effect estimation in the sense of whether they produce equivalent sets of formulas for $p(y \mid do(t))$ for different pairs $(T, Y) \in \mathbf{V} \times \mathbf{V}$. The graph G could be a known ground truth graph while H was proposed by a causal discovery algorithm, or they could both be provided by different domain experts. We decompose the computation of FID into identifier, canonicalizer, and verifier to separate (i) how candidate estimands are generated and (ii) how they are compared across graphs in the latent-variable setting. While conceptually aligned with AID [Henckel et al., 2024], the objects generated and compared are sets of fixing-based estimands in ADMGs.

Figure 2 provides an overview of our approach. At the core, we exploit that in Equation (1), for each district in the graph induced by $\mathbf{Y}^*$, we have a factor $\phi_{\mathbf{V}\setminus\mathbf{D}}(\ldots)$, where we fix all nodes except for $\mathbf{D}$. That means, for each factor, we may have multiple valid fixing sequences. Hence, any combination of valid fixing sequences for each of the factors results in a valid identification formula, which are all guaranteed to be equal (under the graphical assumptions) albeit potentially being different expressions. Our proposed identification strategy $\mathcal{I}$ leverages these degrees of freedom to generate an exhaustive list of identifying formulas for all causal effects of interest via fixing.

Definition 1 (fixing identification $\mathcal{I}$)**.** *In the setting of Theorem 1, for distinct $T, Y \in \mathbf{V}$, we define $\mathcal{I}(G, T, Y)$ to be $\emptyset$ if $p(y \mid do(t))$ is not identifiable in G, $\{p(y)\}$ if Y is not a descendant of T, and otherwise the set of identifying expressions from Equation* (1) *for all combinations of valid fixing sequences of $\mathbf{V} \setminus \mathbf{D}$ across the districts $\mathbf{D} \in \mathcal{D}(G(\mathbf{V})_{\mathbf{Y}^*})$.*

In fully observed graphs, many identification procedures for $p(y \mid do(t))$ can be summarized by a single adjustment-based functional [Pearl, 1995, Perković et al., 2018]. In ADMGs, identification generally does not reduce to selecting a covariate set: fixing may branch over valid fixing sequences and yield multiple syntactically distinct but graph-valid identifying expressions, and the number of such expressions may differ across graphs and queries. To see why this matters, consider two graphs G and H and a fixed query (T, Y). Let E_G and E_H be the deduplicated sets of canonicalized identifying expressions. A "single-estimand" comparison that only inspects one selected pair (e_G, e_H) is fundamentally ambiguous in two ways. First, even if $e_G = e_H$, so a shared expression is selected,

this does not imply equality of the full sets of identifying expressions. Second, $e_G \neq e_H$ may simply be because we are selecting among multiple valid expressions. So any individual comparison does not determine the overlap and/or differences between G and H and is driven by incidental algorithmic choices. For these reasons, using all identifying expressions is the principled way to obtain a stable, graph-induced object for comparison.

Accordingly, our verifier $\mathcal{V}$ operates directly on collections of symbolic identifying expressions and determines their overlap by checking equivalence between expressions involving nested conditioning, products and ratios of conditional densities, and marginalization over subsets of variables; see Figure 2. Due to the combinatorial growth of expressions, a first step is to eliminate redundancy, i.e., expressions that are "trivially equal" in that they can be transformed into one another by rules of probability without leveraging graph-induced structure. That is, we do not simplify according to conditional independencies implied by a graph via the nested Markov property, as in practice, when comparing graphs G, H, we cannot know w.r.t. which (if any) of the two graphs the distribution is nested Markov. One effective approach is to reduce each expression in isolation to a canonical "simplified" form. Before comparison, fixing-generated expressions are written in a common symbolic language by recursively replacing fixing and marginalization operations by their quotient and sum/integral forms. We then define a deterministic canonicalization map $\mathcal{C}$ by applying a fixed set of graph-free rewrite rules.

Definition 2 (Canonicalizer $\mathcal{C}$)**.** *Fix a total order $\prec$ over variables. Let $\mathcal{E}$ denote the class of symbolic probability expressions produced by the identifier $\mathcal{I}$ after recursively writing fixing-generated kernels in terms of sums/integrals, products, quotients, and probability terms. Assume strictly positive densities, product spaces for variable domains, and absolute integrability so that sums/integrals may be reordered. The canonicalization map $\mathcal{C} : \mathcal{E} \to \mathcal{E}$ is defined as the result of recursively applying the following rewrite rules until a fixed point is reached (no rule applies), with all sorting performed deterministically according to $\prec$:*

1. ***Conditional expansion (Bayes rule):*** *replace every conditional term by a quotient of joint terms,*

$$P(A \mid B) \;\mapsto\; \frac{P(A, B)}{P(B)} \qquad \textit{(for } P(B) > 0\textit{)}.$$

2. ***Flattening and canceling fractions:*** *rewrite nested quotients into a single quotient and cancel common factors,*

$$\frac{\frac{A}{B}}{\frac{C}{D}} \;\mapsto\; \frac{AD}{BC}, \qquad \frac{A \cdot B}{A \cdot C} \;\mapsto\; \frac{B}{C} \quad \textit{(for } A \neq 0\textit{)}.$$

3. ***Independent-factor extraction and trivial marginals:*** *inside sums/integrals, pull out factors that do not contain the bound variable as a free variable (i.e., not already marginalized out),*

$$\int_x A(x)B \;\mapsto\; B \int_x A(x), \qquad x \notin \mathrm{free}(B).$$

   Then remove primitive marginalizations, e.g.

$$\int_X P(\ldots, X, \ldots)\, dX \;\mapsto\; P(\ldots).$$

   In particular, marginalizing a joint density over all of its arguments yields 1.

4. ***Canonical ordering of bound variables and joint arguments:*** *reorder sums/integrals by $\prec$ and reorder joint density arguments accordingly, e.g.,*

$$\int_{X_2, X_1} \mapsto \int_{X_1, X_2}, \qquad P(X_2, X_1, \ldots) \;\mapsto\; P(X_1, X_2, \ldots).$$

5. ***Consistent sorting of commutative operations:*** *sort factors in commutative products deterministically (treat divisions as multiplication by inverses).*

This procedure turns every obtained identifying expression into a canonical graph-free form, thus allowing for de-duplication of expressions per graph and node pair and allowing for fast comparison of expressions between graphs. Let $\equiv_{\mathrm{gf}}$ denote the equivalence relation generated by the graph-free probability identities and rewrite rules used by the canonicalizer, excluding graph-induced nested Markov equalities. We prove soundness and completeness of $\mathcal{C}$ with respect to this graph-free rewrite system for fixing-generated identification expressions.

Theorem 2 ($\mathcal{C}$ is sound & complete)**.** *Let $e_1, e_2 \in \mathcal{E}$ be fixing-generated identification expressions. Then $\mathcal{C}(e_1) = \mathcal{C}(e_2)$ is equivalent to $e_1 \equiv_{\text{gf}} e_2$.*

We prove both directions step by step in Appendix E.

Thus, the canonicalizer is sound and complete for the symbolic equivalences that FID intentionally uses, i.e., ordinary graph-free transformations of fixing-generated expressions. Next, we propose a symbolic verifier $\mathcal{V}$ that compares canonicalized estimands across graphs.

Definition 3 (Symbolic Verifier $\mathcal{V}$)**.** *Given a reference graph G, a candidate graph H, a pair (T, Y), and an identification strategy $\mathcal{I}$, define the (multi)sets*

$$E_G := \text{set}\big(\mathcal{C}(\mathcal{I}(G, T, Y))\big),\ E_H := \text{set}\big(\mathcal{C}(\mathcal{I}(H, T, Y))\big),$$

where $\emptyset$ denotes "not identifiable." We write $p(y)$ for the degenerate case $Y \notin \text{de}(T)$. Then the symbolic verifier can be defined as Algorithm 1.

The normalization by $|E_H|$ is deliberate: in the directional setting, H is the proposed graph and E_H is the set of estimands it asserts. When $E_H \neq \emptyset$, $\mathcal{V}$ measures the fraction of candidate estimands not supported by the reference graph G, giving a precision-style error. The empty-set cases are handled separately. Two non-identifiable graphs yield zero disagreement, while a candidate graph that fails to identify an effect identified by G receives maximal disagreement. This asymmetry here is intentional, since exchanging G and H yields the complementary omission-style error.

Algorithm 1 Symbolic Verifier $\mathcal{V}$

Input: reference G, candidate H, pair (T, Y), identifier $\mathcal{I}$, canonicalizer $\mathcal{C}$
$E_G \leftarrow \text{set}(\mathcal{C}(\mathcal{I}(G, T, Y)))$
$E_H \leftarrow \text{set}(\mathcal{C}(\mathcal{I}(H, T, Y)))$
if $E_H = \emptyset$ and $E_G = \emptyset$ **then**
  return 0
else if $E_G \neq \emptyset$ and $E_H = \emptyset$ **then**
  return 1
else
  return $1 - \frac{|E_G \cap E_H|}{|E_H|}$
end if

Definition 3 allows $\mathcal{V}$ to admit a simple set-prediction interpretation. $\mathcal{V}(G, H; T, Y)$ can be read as the fraction of estimands asserted by H not supported by G—a precision-style error, crucial for avoiding false claims in safety-critical domains. Reversing the roles yields the complementary omission score, $\mathcal{V}(H, G; T, Y)$—a recall-style error measuring missed reference estimands, a priority in exploratory discovery. We now aggregate one or both of these per-pair scores over a set $\mathbf{S}$ of treatment-outcome pairs to define a graph-level distance between ADMGs.

Definition 4 (Fixing Identification Distance (FID))**.** *We define our fixing identification distance via*

$$d_{\mathcal{I}}(G, H; \mathbf{S}) \ = \sum_{(T,Y)\in\mathbf{S}} \mathcal{V}(G, H; T, Y)\ , \tag{2}$$

and its normalized version as

$$\bar{d}_{\mathcal{I}}(G, H; \mathbf{S}) := \frac{1}{|\mathbf{S}|}\, d_{\mathcal{I}}(G, H; \mathbf{S})\ . \tag{3}$$

Since (normalized) FID are non-symmetric, we also introduce a symmetrized version

$$d^{\text{sym}}_{\mathcal{I}}(G_1, G_2; \mathbf{S}) \ = \ \frac{1}{2}\Big(d_{\mathcal{I}}(G_1, G_2; \mathbf{S}) + d_{\mathcal{I}}(G_2, G_1; \mathbf{S})\Big). \tag{4}$$

and denote its normalized version by $\bar{d}^{\text{sym}}_{\mathcal{I}}(G_1, G_2; \mathbf{S})$.

As aggregated proportions, unlike edit-based graph scores, FID generally can take continuous values. FID is a *pre-metric*: it is nonnegative and vanishes on the diagonal, but it need not be symmetric (except after symmetrization) or satisfy the triangle inequality. This reflects its query-dependent and task-aligned nature. A zero directional FID means only that, for the chosen treatment-effect set S, the candidate graph H asserts no estimand unsupported by the reference graph G. It does not imply graph equality or agreement on queries outside S.

When a causal discovery method outputs an equivalence class rather than a single graph (e.g., a CPDAG or a PAG), FID is generally set-valued; we therefore report a range over consistent realizations (see Appendix C for details). Our canonicalizer $\mathcal{C}$ is designed to be sound and practically effective for the expression classes encountered here; Appendix F.1 reports diagnostic ablations that quantify its de-duplication behavior and influence on resulting distances.

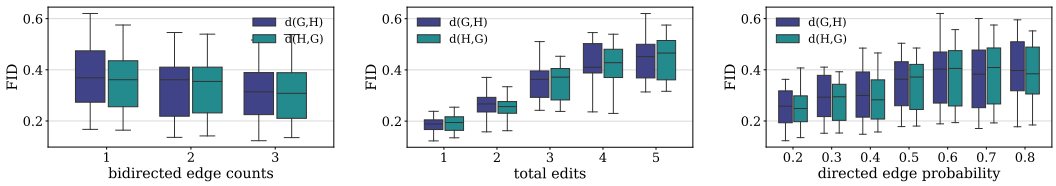

Figure 3: We show $\bar{d}_{\mathcal{I}}(G, H; \mathbf{S})$ (blue) and $\bar{d}_{\mathcal{I}}(H, G; \mathbf{S})$ (teal) under different aggregations over bidirected edge counts (left), total number of edits (middle), and directed edge probability (right).

4 Empirical Results

We first use controlled simulations to illustrate the behavior of FID and then compare it with existing graph distances. The goal of these experiments is not to show that FID is uniformly superior, but to verify that it responds coherently to perturbations of the graphical structures used by fixing-based identification: ancestry, districts, intrinsic sets. These mechanisms are present independently of the graph size, although larger graphs create more possible queries and fixing sequences. Unlike SID and AID, FID operates on ADMGs. The most direct comparison in the latent setting is with a separation-based distance (SD) computed on MAGs. Unless noted otherwise, we report the normalized score $\bar{d}_{\mathcal{I}}^{\text{sym}}$ from Definition 4.

Graph Generation. We sample Erdös–Rényi (ER) with n_{vars} observed variables via: 1) Draw a random topological order π of the nodes. For each ordered pair (i, j) with $\pi(i) < \pi(j)$, include a directed edge $i \to j$ independently with probability p_{dir}. 2) Add exactly n_{bi} bidirected edges by uniformly sampling that many unordered node pairs iteratively without replacement.

We use a grid over $(p_{\text{dir}}, n_{\text{bi}})$ and draw the same number of graphs per setting. In our configuration we set $n_{\text{vars}} = 5$, sweep $p_{\text{dir}} \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ and $n_{\text{bi}} \in \{1, 2, 3\}$, and generate 20 graphs per combination, for a total of $7 \times 3 \times 20 = 420$ ADMGs. For $\mathbf{S}$ we use all possible tuples of distinct nodes and aggregate results over graph pairs by averaging.

FID Results. To obtain a candidate ADMG H, we apply $k \in \{1, 2, 3, 4, 5\}$ edits to a given G chosen from the following types: reverse a directed edge (*reverse dir*), turn a directed edge into a bidirected edge (*dir to bi*), a bidirected edge into a directed edge (*bi to dir*), add a directed edge (*add dir*), delete directed edge (*del dir*), add bidirected edge (*add bi*), delete bidirected edge (*del bi*). Edits are applied sequentially (skipping ones that would create cycles). The three panels in Figure 3 show different aggregations of the corresponding distances: (i) group by the number of bidirected edges in G and pool results over all k and p_{dir}; (ii) pool all G and group by total edits k; (iii) group by the p_{dir} used for G and pool all k and n_{bi}. As expected, FID clearly increases with the number of total edits. The original density of directed edges also increases FID, which may be explained by an edit in a dense graph having broader consequences for resulting cause-effect estimands than in sparse graphs.

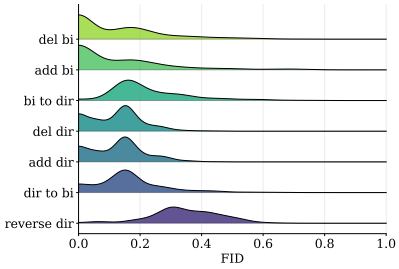

Figure 4: FID ($\bar{d}_{\mathcal{I}}^{\text{sym}}$) distributions for the different edit types.

We next isolate single-edit effects according to the different types of edits. Starting from G, we apply exactly one edit to obtain H and evaluate $\bar{d}_{\mathcal{I}}^{\text{sym}}(G, H; \mathbf{S})$. Figure 4 shows the distribution of distances for different edit types. While all of these edits would count equally in a simple SHD metric, there are substantial differences in FID. Most notably, reversing an edge simultaneously changes ancestor relations and can create/remove v-structures or open/close back-door paths. Even one such edit can thus lead to a large FID. We can similarly hypothesize that changing types of edges (*dir to bi*, *bi to dir*) or removing/adding directed edges lead to substantial changes in causal ordering and thus have a sizable impact on cause-effect estimands. The addition or deletion of bidirected edges (introducing/removing latent confounding) on average leads to the lowest FID; however, the distribution has a heavy tail, i.e., these edits can lead to substantial differences in downstream cause-effect estimation captured by FID.

Comparison with Other Graph Distances. We compare our estimand-based distance to two structural baselines that can be applied in the presence of latent confounding: SD (we use the normalized ZL variant) and SHD. Because SD is defined on MAGs, we convert our ADMGs to MAGs as follows. Each reference ADMG G is mapped to a Markov-equivalent MAG $\tilde{G}$ using the procedure of Hu and Evans [2020]. We then generate perturbed ADMGs by applying $k \in \{1, \dots, 7\}$ random allowed edits, convert the resulting graphs to MAGs, and obtain $(\tilde{G}, \tilde{H})$. The conversion can collapse distinct ADMG perturbations to the same MAG, so the number of MAG discrepancies may differ from the number of ADMG edits.

For each pair $(\tilde{G}, \tilde{H})$ we compute $\mathrm{SD}(\tilde{G}, \tilde{H})$, $\mathrm{SHD}(\tilde{G}, \tilde{H})$, and our symmetric normalized distance $\bar{d}_{\mathcal{I}}^{\mathrm{sym}}(\tilde{G}, \tilde{H}; \mathbf{S})$. Figure 5 shows scatter plots across all graphs and edit levels.

The comparison with ZL-SD shows that FID and ZL-SD capture different aspects of graph differences albeit being weakly correlated (Pearson $r = 0.292$). SD aggregates changes in (m-)separations; many distinct edit patterns collapse to the same SD value, producing vertical stripes. Small SD moves that open/close a confounded path can cause large jumps in FID, while large SD moves can be redundant for the fixing formulas and leave FID small. Hence, FID and ZL-SD are complementary graph distances.

The comparison with SHD shows a stronger correlation (Pearson $r = 0.820$), but the scatter plot still shows substantial variation within given SHD bins. The distances span a wide range, reflecting that different edit types perturb the causal-effect task to very different degrees. Our single-edit analysis (Figure 4) shows that edits counted identically by SHD can have significantly different consequences for the induced identifying expressions. In this sense, SHD captures only how much the graph is edited, while FID additionally captures how it is edited in terms of its implications for causal-effect identification.

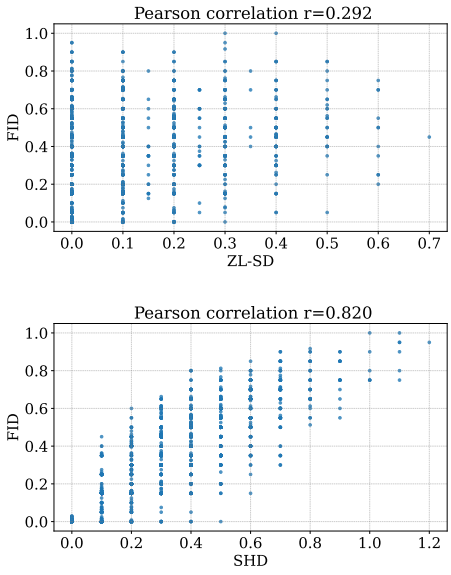

Figure 5: Comparison of our FID ($\bar{d}_{\mathcal{I}}^{\mathrm{sym}}$) with SHD and ZL-SD on MAGs. FID shows positive correlation with both, but still adds new information beyond existing metrics.

5 Conclusion

Summary. We introduced the *Fixing Identification Distance* (FID), an estimand-level distance measure for ADMGs that compares graphs by the causal answers they entail to cause-effect estimation queries under latent confounding. FID combines fixing-based identification with symbolic canonicalization and a symbolic verifier to (i) determine identifiability and (ii) compare the resulting functionals across treatment-outcome pairs, yielding a fractional, task-aligned distance. A symmetric variant supports settings without a designated ground truth. This reframes graph evaluation around the downstream causal queries that many researchers ultimately care about.

Empirically, FID behaves coherently under controlled perturbations, increasing with both edit count and directed-edge density. The results suggest that FID can distinguish structural edits by their consequences for downstream estimands. Edits that are identical under cardinality-based metrics can have different effects on fixing-based identification. While FID correlates with SHD on total error counts, its divergence from ZL-SD suggests that it captures information not reducible to global separation patterns. FID is therefore not intended to replace structural distances such as SHD in all settings. Instead, it complements them by asking whether graph differences change the causal estimands available for a specified set of treatment–outcome queries. Since DAGs are ADMGs without bidirected edges, FID also applies to fully observed settings. Overall, FID provides a query-aligned discrepancy for ADMGs that can support both causal discovery evaluation and sensitivity analysis over alternative assumed graphs.

Limitation. The main computational bottleneck is the growth in the number of valid fixing-generated expressions, together with the quadratic number of ordered treatment–outcome pairs if all pairs are evaluated. Our optimized implementation substantially reduces the practical computa-

tional burden through memoization, early deduplication, and caching. Nevertheless, the worst-case combinatorial complexity remains, and runtime can increase sharply for larger or denser graphs, as shown by the runtime distributions in Appendix F.3. This limitation should be interpreted together with the intended use of FID. FID is not designed as a replacement for cheap structural scores (SHD) in large-scale screening over thousands of variables. Its main use case is targeted evaluation of causal discovery outputs or sensitivity analysis over a limited set of scientifically relevant treatment–outcome queries, especially when latent confounding makes adjustment-based distances insufficient. In such settings, the relevant question is often not whether two large graphs differ globally, but whether they support the same causal estimand for a specific intervention of interest.

Acknowledgments and Disclosure of Funding

This work has been supported by the German Federal Ministry of Education and Research (Grant: 01IS24082). This work has been supported by the European Union (ERC, DYNAMICAUS, 101221985). The authors declare no competing interests.

References

Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. ISSN 00905364, 21688966.

Yashas Annadani, Jonas Rothfuss, Alexandre Lacoste, Nino Scherrer, Anirudh Goyal, Yoshua Bengio, and Stefan Bauer. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

Susan Athey and Guido W Imbens. The state of applied econometrics: Causality and policy evaluation. *Journal of Economic perspectives*, 31(2):3–32, 2017.

Masood A Badri, Samaa Attia, and Abdulla M Ustadi. Healthcare quality and moderators of patient satisfaction: testing for causality. *International journal of health care quality assurance*, 22(4): 382–410, 2009.

Daniel C Castro, Ian Walker, and Ben Glocker. Causality matters in medical imaging. *Nature Communications*, 11(1):3673, 2020.

Mathieu Chevalley, Yusuf H Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab. A large-scale benchmark for network inference from single-cell perturbation data. *Communications Biology*, 8(1):412, 2025.

Martijn de Jongh and Marek J Druzdzel. A comparison of structural distance measures for causal Bayesian network models. *Recent advances in intelligent information systems, challenging problems of science, computer science series*, pages 443–456, 2009.

Rajeev H. Dehejia and Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association*, 94(448):1053–1062, 1999.

Marco Eigenmann, Sach Mukherjee, and Marloes Maathuis. Evaluation of causal structure learning algorithms via risk estimation. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 151–160. PMLR, 03–06 Aug 2020.

Juan L Gamella, Jonas Peters, and Peter Bühlmann. The causal chambers: Real physical systems as a testbed for ai methodology. *arXiv preprint arXiv:2404.11341*, 2024.

Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, Miltiadis Allamanis, and Cheng Zhang. Deep End-to-end Causal Inference. *Open Source Software Package*, February 2022.

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*, 32, 2019.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

Moritz Hardt. The emerging science of machine learning benchmarks. Online at `https://mlbenchmarks.org`, 2025. Manuscript.

Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E Oliphant. Array programming with NumPy. *Nature*, 2020.

Christina Heinze-Deml, Marloes H Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5(1):371–391, 2018.

Leonard Henckel, Emilija Perković, and Marloes H. Maathuis. Graphical criteria for efficient total effect estimation via adjustment in causal linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2):579–599, 03 2022. ISSN 1369-7412. doi: 10.1111/rssb.12451.

Leonard Henckel, Theo Würtzen, and Sebastian Weichwald. Adjustment identification distance: A gadjid for causal structure learning, 2024.

Rebecca J Herman, Jonas Wahl, Urmi Ninad, and Jakob Runge. Unitless unrestricted Markov-consistent scm generation: Better benchmark datasets for causal discovery. *arXiv preprint arXiv:2503.17037*, 2025.

Z Hu and R Evans. Faster algorithms for Markov equivalence. *36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 739–748, 2020.

John D Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 2007.

Guido W Imbens and Jeffrey M Wooldridge. Recent developments in the econometrics of program evaluation. *Journal of economic literature*, 47(1):5–86, 2009.

Jaron J. R. Lee, Rohit Bhattacharya, Razieh Nabi, and Ilya Shpitser. Ananke: A python package for causal inference using graphical models, 2023.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. *Advances in Neural Information Processing Systems*, 35: 13104–13118, 2022.

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

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, 2000.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 2011.

Emilija Perković, Johannes Textor, Markus Kalisch, and Marloes H Maathuis. Complete graphical characterization and construction of adjustment sets in Markov equivalence classes of ancestral graphs. *Journal of Machine Learning Research*, 18(220):1–62, 2018.

Eric Perrier, Seiya Imoto, and Satoru Miyano. Finding optimal Bayesian network given a super-structure. *Journal of Machine Learning Research*, 9(74):2251–2286, 2008.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Comput.*, 27(3):771–799, March 2015. ISSN 0899-7667.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Maxime Peyrard and Robert West. A ladder of causal distances, 2021.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34:27772–27784, 2021.

Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30 (4):962 – 1030, 2002. doi: 10.1214/aos/1031689015.

Thomas S. Richardson. A factorization criterion for acyclic directed mixed graphs, 2014.

Thomas S. Richardson, Robin J. Evans, James M. Robins, and Ilya Shpitser. Nested Markov properties for acyclic directed mixed graphs. *The Annals of Statistics*, 51(1), February 2023. ISSN 0090-5364. doi: 10.1214/22-aos2253.

Felix L. Rios, Giusi Moffa, and Jack Kuipers. Benchpress: A versatile platform for structure learning in causal and probabilistic graphical models. *Journal of Statistical Software*, 114(12):1–43, 2025. doi: 10.18637/jss.v114.i12.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005.

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

William H Sewell and Vimal P Shah. Social class, parental encouragement, and educational aspirations. *American journal of Sociology*, 73(5):559–572, 1968.

Ilya Shpitser, Tyler VanderWeele, and James M Robins. On the validity of covariate adjustment for estimating causal effects. *arXiv preprint arXiv:1203.3515*, 2012.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal dags via directed clique tree, 2020.

Johannes Textor and Maciej Liskiewicz. Adjustment criteria in causal diagrams: An algorithmic perspective. *arXiv preprint arXiv:1202.3764*, 2012.

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, 2016.

The pandas development team. pandas-dev/pandas: Pandas, February 2020.

Jin Tian and Judea Pearl. A general identification condition for causal effects. In *Eighteenth National Conference on Artificial Intelligence*, page 567–573, USA, 2002. American Association for Artificial Intelligence. ISBN 0262511290.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, October 2006.

Guido van Rossum and Fred L Drake. *Python 3 Reference Manual*. CreateSpace, 2009.

Jussi Viinikka, Ralf Eggeling, and Mikko Koivisto. Intersection-validation: A method for evaluating structure learning without ground truth. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1570–1578. PMLR, 09–11 Apr 2018.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J van der Walt, Matthew Brett, Joshua Wilson, K Jarrod Millman, Nikolay Mayorov, Andrew R J Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E A Quintero, Charles R Harris, Anne M Archibald, Antônio H Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022.

Jonas Wahl and Jakob Runge. Separation-based distance measures for causal graphs, 2025.

Jonas Bernhard Wildberger, Siyuan Guo, Arnab Bhattacharyya, and Bernhard Schölkopf. On the interventional kullback-leibler divergence. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing, editors, *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pages 328–349. PMLR, 11–14 Apr 2023.

Andy B Yoo, Morris A Jette, and Mark Grondona. Slurm: Simple linux utility for resource management. In *Workshop on job scheduling strategies for parallel processing*, pages 44–60. Springer, 2003.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: theory and practice. *International Journal of Approximate Reasoning*, 151:101–129, 2022.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

A Additional Background

Basic graph notation. For a graph G over nodes $\mathbf{V}$ and $X \in \mathbf{V}$, $\mathrm{pa}_G(X)$, $\mathrm{ch}_G(X)$, $\mathrm{de}_G(X)$, $\mathrm{an}_G(X)$, and $\mathrm{nd}_G(X)$ denote the parents, children, descendants, ancestors, and non-descendants of X in G, respectively.

$$\begin{aligned}
\mathrm{pa}_G(X) &:= \{\, Y \in \mathbf{V} \setminus \{X\} : Y \to X \text{ is an edge in } G \,\},\\
\mathrm{ch}_G(X) &:= \{\, Y \in \mathbf{V} \setminus \{X\} : X \to Y \text{ is an edge in } G \,\},\\
\mathrm{an}_G(X) &:= \{\, Y \in \mathbf{V} : \text{there exists a directed path } Y \rightsquigarrow X \,\}\{X\},\\
\mathrm{de}_G(X) &:= \{\, Y \in \mathbf{V} : \text{there exists a directed path } X \rightsquigarrow Y \,\},\\
\mathrm{nd}_G(X) &:= \mathbf{V} \setminus (\mathrm{de}_G(X) \cup \{X\}).
\end{aligned}$$

A (directed) *path* in G is a sequence of distinct nodes $v_0, \ldots, v_k$ such that $v_i \to v_{i+1}$ is a directed edge of G for all i. By convention, a node v_i is neither an ancestor, nor a descendant, nor a non-descendant of itself. If G contains bidirected edges ($\leftrightarrow$), the *siblings* of X are its bidirected neighbors, and the *district* $\mathrm{dis}_G(X)$ is the bidirected-connected component containing X. In DAGs, the (probabilistic) *Markov blanket* of X is $\mathrm{mb}_G(X) = \mathrm{pa}_G(X) \cup \mathrm{ch}_G(X) \cup \big(\mathrm{pa}_G(\mathrm{ch}_G(X)) \setminus \{X\}\big)$. In CADMGs used by nested Markov models, the *Markov blanket* that appears in the nested factorization is $\mathrm{mb}_G(X) = \big(\mathrm{dis}_G(X) \setminus \{X\}\big) \cup \mathrm{pa}_G\big(\mathrm{dis}_G(X)\big)$.

Induced subgraph. Let G be a (mixed) graph on vertex set $\mathbf{V}$ with edge set E (edges may be directed $\to$ or bidirected $\leftrightarrow$). For $A \subseteq \mathbf{V}$, the *induced subgraph* of G on A, denoted G_A, is the graph whose

$$\text{vertex set is } A, \qquad \text{edge set is } E_A \;=\; \{\, e \in E : \text{both endpoints of } e \text{ lie in } A \,\},$$

with each edge in E_A retaining the same type and orientation it has in G. Equivalently, G_A is obtained by deleting all vertices in $\mathbf{V} \setminus A$ and then removing any edges incident to the deleted vertices.

Graph classes. A *DAG* has only directed edges ($\to$) and no directed cycles. An *ADMG* (acyclic directed mixed graph) has directed and bidirected edges but no directed cycles; bows ($X \to Y$ together with $X \leftrightarrow Y$) are allowed. A *CADMG* $G(\mathbf{V}, \mathbf{W})$ partitions nodes into random $\mathbf{V}$ and fixed $\mathbf{W}$ and contains no arrowheads into $\mathbf{W}$. A *CPDAG* (completed partially directed acyclic graph) compactly represents a Markov equivalence class of DAGs [Andersson et al., 1997]: directed edges are oriented the same way in every DAG in the class, while undirected edges indicate reversible orientations. Two DAGs are *Markov equivalent* if and only if they share the same skeleton and v-structures. An *ancestral graph* is a mixed graph with (i) no directed cycles; (ii) no arrowhead into an ancestor (e.g., if $X \leftrightarrow Y$ then neither $X \to Y$ nor $Y \to X$ is present); and (iii) if $X - Y$ is undirected, then X and Y have no incident arrowheads. A *MAG* (maximal ancestral graph) Richardson and Spirtes [2002] is an ancestral graph which is maximal if and only if for every pair of nonadjacent vertices X, Y, there exists some set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$. Intuitively, in a MAG every missing edge corresponds to at least one conditional independence; adding any edge would change the independence model. A *PAG* (partial ancestral graph) represents a Markov equivalence class of MAGs [Richardson and Spirtes, 2002]. Its edges carry endpoint marks: tails "$-$", arrowheads "$>$", and circles "$\circ$", where circles indicate uncertainty about the invariant endpoint mark across the equivalence class.

Separation. In DAGs, d-separation characterizes conditional independencies. For mixed graphs (ancestral graphs, ADMGs), the corresponding notion is *m-separation*: A path is m-connecting given $\mathbf{Z}$ if every non-collider on the path is *not* in $\mathbf{Z}$ and every collider on the path is an ancestor of some node in $\mathbf{Z}$; if no such path exists, the endpoints are m-separated given $\mathbf{Z}$.

Latent projection and districts. Given a DAG $G(\mathbf{V} \cup \mathbf{L})$ with latents $\mathbf{L}$, its *latent projection* onto $\mathbf{V}$ is the ADMG obtained by (i) adding $X \to Y$ if there is a directed path $X \rightsquigarrow Y$ whose non-endpoints are latent, and (ii) adding $X \leftrightarrow Y$ if there is a path between X and Y whose non-endpoints are non-colliders and latent. Bidirected-connected components in a (C)ADMG are called *districts*.

Markov equivalence of ADMG and MAG. Two (mixed) graphs on the same vertices are *Markov equivalent* if they imply exactly the same set of conditional independencies by m-separation. Every ADMG has a Markov-equivalent MAG obtained by the projection in Algorithm 4 [Hu and Evans, 2020].

Tails. For $\mathbf{A} \subseteq \mathbf{V}$, write $G_{\mathbf{A}}$ for the induced subgraph on $\mathbf{A}$. For a set S, let $\mathrm{an}_G(S)$ be all ancestors of nodes in S (including S). The *district* of v in $G_{\mathbf{A}}$, denoted $\mathrm{dis}_{G_{\mathbf{A}}}(v)$, is the bidirected-connected component containing v in $G_{\mathbf{A}}$. Define the *tail* [Richardson, 2014] of a single vertex v by

$$\mathrm{tail}_G(v) \;:=\; \big(\mathrm{dis}_{G_{\{v\}\cup \mathrm{an}_G(\mathrm{pa}_G(v))}}(v) \setminus \{v\}\big) \;\cup\; \mathrm{pa}_G\big(\mathrm{dis}_{G_{\{v\}\cup \mathrm{an}_G(\mathrm{pa}_G(v))}}(v)\big).$$

Intuitively, $\mathrm{tail}_G(v)$ collects variables that, in the ancestor subgraph around v's parents, share v's bidirected component or parent of that component. In the projected MAG these become the parents of v.

Heads. A set $\mathbf{H} \subseteq \mathbf{V}$ is a *head* [Richardson, 2014] if it is barren (no member has a descendant within $\mathbf{H}$ other than itself) and $\mathbf{H}$ lies in a single district of the ancestor subgraph $G_{\mathrm{an}_G(\mathbf{H})}$. When $|\mathbf{H}| = 2$, say $\mathbf{H} = \{v, w\}$, this exactly means v and w lie in the same district of $G_{\mathrm{an}_G(\{v,w\})}$. In the projection, such pairs correspond to bidirected edges $v \leftrightarrow w$ in the MAG.

B FID on DAGs and CPDAGs

A DAG is an ADMG without bidirected edges, so FID applies directly. In a DAG, all effects are identifiable. This result also applies to the fixing operation. In a fully observed DAG, every interventional distribution $p(y \mid do(t))$ is identifiable. Moreover, every node forms a singleton district; hence, all nodes are fixable and reachable. While both AID and FID are identification-based, AID depends on a chosen adjustment strategy (e.g., parent, ancestor, or optimal adjustment), whereas FID derives the fixing-based estimand implied by the graph. Consequently, they may disagree for a given pair (T, Y).

Example. In Figure 6, let $T = X_1$ and $Y = X_3$, the identifier (parent and ancestor adjustment) of the AID provides the same expression

$$\mathcal{I}(G; x_1, x_3) = \int p(x_3 \mid x_1, x_2) p(x_2) dx_2 = \mathcal{I}(H; x_1, x_3)\,.$$

Therefore, the verifier of AID would give a zero distance. By contrast, fixing algorithms disagree on these two graphs. On graph G, $Y^* = \{X_2, X_3\}$ and the identifier of FID gives the following functional (before converting to symbolic form)

$$p(x_3 \mid do(x_1)) = \int \phi_{X_1,X_3}(p(\mathbf{x_V}); G)\phi_{X_1,X_2}(p(\mathbf{x_V}); G) dx_2.$$

For graph H, $Y^* = \{X_3\}$ and the functional provided by the FID identifier is

$$p(x_3 \mid do(x_1)) = \phi_{x_1,x_2}(p(\mathbf{x_V}; H)).$$

These two estimands are generally not algebraically equal when evaluated under distributions Markov to G (in G, $X_2 \not\perp\!\!\!\perp X_1$ and $X_2 \not\perp\!\!\!\perp X_3 \mid X_1$), so FID records a disagreement for (T, Y).

By contrast, depending on the adjustment strategy, AID may output the same adjustment expression for both graphs (e.g., a strategy that always adjusts for X_2), leading to zero AID on this pair. FID is therefore more sensitive to the minimal fixing-based estimand implied by each DAG. In this case, AID sees these as equivalent under adjustment, but FID identifies a structural difference in the fixing paths, correctly detecting that an adjustment is required in G but not in H.

The DAG case considered above extends naturally to CPDAGs, which encode Markov equivalence classes of DAGs. Since different consistent orientations within a CPDAG may induce different identifying functionals for a given (T, Y), a single scalar FID is no longer well-defined. Following the SID treatment [Peters and Bühlmann, 2015], we therefore evaluate FID over all DAG extensions of each CPDAG and summarize the result as a range. Concretely, for CPDAGs C_1, C_2, we report the minimum and maximum normalized directional FID attained by any pair of DAGs $G \in [C_1]$, $H \in [C_2]$. The interval collapses to a point when both inputs are DAGs. In practice, exact enumeration is feasible for small graphs; for larger CPDAGs, one can sample consistent orientations to approximate the bounds.

Let C be a CPDAG and $[C]$ the set of DAGs that are Markov equivalent to C (its DAG extensions). For a set of ordered treatment-outcome pairs $\mathbf{S} \subseteq \{(T, Y) \in \mathbf{V} \times \mathbf{V} : T \neq Y\}$, define the *directional*



Figure 6: Example illustrating the discrepancy between AID and FID. AID sees these as equivalent under adjustment, but FID identifies a structural difference in the fixing paths.

FID range between two CPDAGs C_1, C_2 as

$$\mathrm{FID}_{\mathcal{I}}^{\mathrm{range}}(C_1, C_2; \mathbf{S}) \;:=\; \big[\, \underline{d},\ \overline{d}\,\big] \quad \text{with} \quad \begin{aligned} \underline{d} &:= \min_{G\in[C_1]} \min_{H\in[C_2]} \bar{d}_{\mathcal{I}}(G, H; \mathbf{S}), \\ \overline{d} &:= \max_{G\in[C_1]} \max_{H\in[C_2]} \bar{d}_{\mathcal{I}}(G, H; \mathbf{S}), \end{aligned}$$

where $\bar{d}_{\mathcal{I}}$ is the normalized directional FID (Equation (3)). The interval contracts to a point if both graphs are DAGs. A symmetric variant follows by replacing $\bar{d}_{\mathcal{I}}$ with its symmetrization $\bar{d}_{\mathcal{I}}^{\mathrm{sym}}$ (Definition 4). A detailed algorithm can be found in Algorithm 2.

Algorithm 2 CPDAG-FID range (exact or sampled)

Input: CPDAGs $\mathcal{C}_1, \mathcal{C}_2$, pair set $\mathbf{S}$, normalized directional FID routine $\bar{d}_{\mathcal{I}}$ (Definition 4), mode $\in \{\textsc{Exact}, \textsc{Sample}\}$, budget B
if EXACT **then**
  $\mathcal{G} \leftarrow [\mathcal{C}_1]$
  $\mathcal{H} \leftarrow [\mathcal{C}_2]$
else
  $\mathcal{G} \leftarrow B$ DAGs sampled uniformly from $[\mathcal{C}_1]$
  $\mathcal{H} \leftarrow B$ DAGs sampled uniformly from $[\mathcal{C}_2]$
end if
$\underline{d} \leftarrow +\infty,\ \overline{d} \leftarrow 0$
for $G \in \mathcal{G}$ **do**
  for $H \in \mathcal{H}$ **do**
    $s \leftarrow \bar{d}_{\mathcal{I}}(G, H; \mathbf{S})$
    $\underline{d} \leftarrow \min(\underline{d}, s);\ \overline{d} \leftarrow \max(\overline{d}, s)$
  end for
end for
return $[\underline{d}, \overline{d}]$

C FID on PAGs via ADMG Realizations

Partial ancestral graphs (PAGs) arise as the standard output of constraint-based causal discovery procedures with latent confounding, such as FCI and its variants [Spirtes et al., 2000]. A PAG $\mathcal{P}$ represents a Markov equivalence class of maximal ancestral graphs (MAGs); we write $[\mathcal{P}]$ for the set of MAGs consistent with $\mathcal{P}$. Since our identifier $\mathcal{I}$ is defined for ADMGs (latent confounding without selection), we focus on the common "no selection bias" setting and range over those MAG realizations that contain no undirected (selection) edges. In this regime, MAG realizations are of the same directed–bidirected form as ADMGs.

As in the CPDAG case, a single scalar FID is generally not well-defined for a PAG because different consistent realizations can induce different identifying functionals for the same query (T, Y). Following the SID-style treatment [Peters and Bühlmann, 2015], we therefore evaluate FID over realizations represented by the PAG and summarize the result as a range. Concretely, we define the induced ADMG realization set

$$[\mathcal{P}]_{\mathrm{ADMG}} \;:=\; \big\{\, \mathrm{admg}(M) \;:\; M \in [\mathcal{P}] \text{ and } M \text{ has no undirected edges} \,\big\},$$

where $\mathrm{admg}(M)$ denotes the graph obtained by retaining all directed edges and representing latent confounding via bidirected edges. Exact enumeration of $[\mathcal{P}]_{\mathrm{ADMG}}$ is feasible only for small graphs; for larger PAGs, we approximate the bounds by sampling consistent realizations (e.g., by sampling a consistent MAG extension without undirected edges, then mapping it to an ADMG).

Formally, let G be an ADMG and $\mathcal{P}$ a PAG over the same node set $\mathbf{V}$, and let $\mathbf{S} \subseteq \{(T, Y) \in \mathbf{V} \times \mathbf{V} : T \neq Y\}$ be a set of ordered treatment–outcome pairs. We define the *directional ADMG–PAG FID range* as

$$\mathrm{FID}_{\mathcal{I}}^{\mathrm{range}}(G, \mathcal{P}; \mathbf{S}) := \left[\ \underline{d},\ \overline{d}\ \right], \quad \underline{d} := \min_{H \in [\mathcal{P}]_{\mathrm{ADMG}}} \bar{d}_{\mathcal{I}}(G, H; \mathbf{S}), \quad \overline{d} := \max_{H \in [\mathcal{P}]_{\mathrm{ADMG}}} \bar{d}_{\mathcal{I}}(G, H; \mathbf{S}),$$

where $\bar{d}_{\mathcal{I}}$ denotes the normalized directional FID (Equation (3)). A symmetric variant is obtained by replacing $\bar{d}_{\mathcal{I}}$ with its symmetrization $\bar{d}_{\mathcal{I}}^{\mathrm{sym}}$ (Definition 4). The interval contracts to a point when the PAG restricts the relevant realization set to (effectively) a single ADMG for the comparisons at hand.

Algorithm 3 ADMG–PAG FID range (exact or sampled realizations)

Input: ADMG G, PAG $\mathcal{P}$, pair set $\mathbf{S}$, normalized directional FID routine $\bar{d}_{\mathcal{I}}$ (Definition 4), mode $\in \{\textsc{Exact}, \textsc{Sample}\}$, budget B
if EXACT **then**
    $\mathcal{H} \leftarrow [\mathcal{P}]_{\mathrm{ADMG}}$
else
    $\mathcal{H} \leftarrow B$ ADMG realizations sampled from $[\mathcal{P}]_{\mathrm{ADMG}}$ ▷ e.g., sample a consistent MAG extension without undirected edges, then map to ADMG
end if
$\underline{d} \leftarrow +\infty,\ \overline{d} \leftarrow 0$
for $H \in \mathcal{H}$ **do**
    $s \leftarrow \bar{d}_{\mathcal{I}}(G, H; \mathbf{S})$
    $\underline{d} \leftarrow \min(\underline{d}, s);\ \overline{d} \leftarrow \max(\overline{d}, s)$
end for
return $[\underline{d}, \overline{d}]$

If $\mathcal{P}$ encodes selection bias (i.e., some consistent MAG realizations contain undirected edges), then restricting to $[\mathcal{P}]_{\mathrm{ADMG}}$ amounts to imposing a no-selection modeling assumption, and mapping general MAGs to ADMGs would discard selection information. Extending FID to ancestral graphs with selection semantics is conceptually possible but requires an identifier beyond the ADMG fixing framework; we leave this extension to future work.

D Experiment Details

D.1 Hardware

Our experiments were conducted on a scheduled cluster with heterogeneous CPU nodes, and the most common types are Intel Xeon 6248r and Intel Xeon Gold 6142. For a single job, we utilized 64 CPU cores and 64GB of memory.

D.2 ADMG $\rightarrow$ MAG Projection

In Section 4, we convert an ADMG G into a Markov equivalent MAG G_m by reconstructing exactly those directed and bidirected edges that are determined by the *tails* of single vertices and by *heads* of size two. In particular (Lemma 3.7 in Hu and Evans, 2020), for any vertices v, w: (i) $w \rightarrow v$ is present in G_m if and only if $w \in \mathrm{tail}_G(v)$; (ii) $v \leftrightarrow w$ is present in G_m if and only if $\{v, w\}$ is a head in G (equivalently, v and w are in the same district inside the ancestor subgraph of $\{v, w\}$). The procedure below computes these objects in polynomial time and yields a MAG that is Markov equivalent to the input ADMG (Table 5 in Hu and Evans, 2020).

Algorithm 4 ADMG $\to$ MAG projection (Algorithm 2 in Hu and Evans, 2020)

Input: ADMG $G = (V, E)$ (access to $\mathrm{pa}_G(\cdot)$, $\mathrm{an}_G(\cdot)$, districts)
Output: MAG $G_m = (V, E_m)$ Markov equivalent to G
Initialize G_m with vertex set V and no edges: $E_m \leftarrow \emptyset$
for each $v \in V$ **do**
    $A_v \leftarrow \{v\} \cup \mathrm{an}_G(\mathrm{pa}_G(v))$
    $D_v \leftarrow \mathrm{dis}_{G_{A_v}}(v)$
    $\mathrm{tail}_G(v) \leftarrow \big(D_v \setminus \{v\}\big) \cup \mathrm{pa}_G(D_v)$
    for each $w \in \mathrm{tail}_G(v)$ **do**
        add edge $w \to v$ to E_m
    end for
end for
for each unordered pair $\{v, w\} \subseteq V$ **with** $v \notin \mathrm{an}_G(w)$ **and** $w \notin \mathrm{an}_G(v)$ **and** v, w in the same district of G **do**
    $A_{vw} \leftarrow \{v, w\} \cup \mathrm{an}_G(\{v, w\})$
    $D_{vw} \leftarrow \mathrm{dis}_{G_{A_{vw}}}(v)$
    if $w \in D_{vw}$ **then**
        add edge $v \leftrightarrow w$ to E_m
    end if
end for
return G_m

D.3 Numerical Validation via Plug-In Estimation

A natural question is whether one should validate FID by numerically evaluating the identifying expressions (estimands) that our identifier returns, e.g., by estimating each expression from data and comparing the resulting numerical causal-effect estimates. While such a validation is possible in principle, it is not well-posed in the level of generality considered in this work and would primarily assess additional, distribution-dependent choices that are outside the scope of a graph-only distance.

First, an identifying expression is a symbolic functional, not an estimator. Turning a functional into a numerical procedure requires selecting an estimator class and often additional modeling assumptions. This issue is particularly pronounced here because the expressions produced by fixing can involve nested marginalizations, conditional densities, and ratios; naive plug-in estimation is generally statistically and computationally fragile (e.g., due to density estimation, division by small quantities, or high-dimensional conditioning). Constructing well-behaved estimators typically requires restricting to a concrete setting (e.g., linear-Gaussian models, discrete finite-state models, or parametric/semi-parametric estimators), and the resulting numerical behavior will depend on these choices.

Second, a symmetric quantitative comparison between two candidate graphs requires specifying a data-generating regime, since we generally do not have "ground truth data" from both graphs (or sometimes from neither graph). Consequently, a controlled numerical validation would need to be conducted on synthetic data, which in turn requires choosing (i) a model class and (ii) a meta-distribution over model parameters (e.g., distributions over edge coefficients, noise variances, and intercepts in a linear-Gaussian SEM). The conclusions of such a study would be inherently conditional on these design choices and may not transfer beyond the chosen regime.

Third, within a fixed graph G, all identifying expressions returned by $\mathcal{I}(G, T, Y)$ are mathematically equivalent representations of the same causal functional. Under correct specification and with consistent estimators, their numerical estimates agree asymptotically when data are generated from G; differences across expressions then primarily reflect finite-sample effects and estimator variance, rather than differences in the underlying causal conclusions. In contrast, when a candidate graph H is incorrect for the data-generating process, some identifying expressions produced by H may correspond to non-identifying or misspecified functionals; detecting such discrepancies is exactly the purpose of FID, which compares identification structure symbolically without committing to a particular estimator family or model class.

For these reasons, we focus on a distance measure defined purely at the graph level, designed to capture generic downstream implications for causal-effect identification in the presence of latent confounding. Distribution-dependent numerical comparisons can be useful for specific applications and estimator choices, but they address a different question—namely, the finite-sample behavior of selected estimators under selected data-generating regimes—than the graph-only objective of FID.

E Graph-Free Soundness and Completeness of the Canonicalizer

This section proves the soundness and completeness claim stated in Theorem 2. To make the proof structure clearer, we split the argument into two separate results: Theorem 3 establishes soundness, while Theorem 4 establishes graph-free completeness. The result is deliberately graph-free. It applies only to equalities generated by ordinary probability identities for kernels, not equalities that rely on additional graphical constraints from a particular ADMG.

Although the results in this section are presented in integral over continuous variables for notational simplicity reason, the same proof applies also to sums.

E.1 Fixing-Generated Kernel and Identification Expressions

Let V be a finite set of variables. A kernel on random variables R indexed by fixed variables W is written as

$$q_R(x_R \mid x_W),$$

and is normalized in x_R:

$$\int_{x_R} q_R(x_R \mid x_W) = 1.$$

We define the class $\mathcal{K}$ of *fixing-generated kernel expressions* recursively.

1. The observed joint density
$$q_V(x_V) = p(x_V)$$
belongs to $\mathcal{K}$.
2. If $q_R(x_R \mid x_W) \in \mathcal{K}$ and $A \subseteq R$, then the marginal kernel
$$\mathrm{Marg}_{R\setminus A}(q_R)(x_A \mid x_W) := \int_{x_{R\setminus A}} q_R(x_R \mid x_W)$$
belongs to $\mathcal{K}$.
3. If $q_R(x_R \mid x_W) \in \mathcal{K}$ and $A, B \subseteq R$ are disjoint, then the conditional kernel
$$q(A \mid B, W) := \frac{q(A, B \mid W)}{q(B \mid W)}$$
belongs to $\mathcal{K}$, whenever the denominator is positive.
4. If $q_R(x_R \mid x_W) \in \mathcal{K}$ and $r \in R$ is fixed, let
$$nd_R(r) \subseteq R \setminus \{r\}$$
denote the nondescendants of r among the current random variables. The fixed kernel is defined by
$$\phi_r(q_R) := \frac{q_R(x_R \mid x_W)}{q_R(x_r \mid x_{nd_R(r)}, x_W)},$$
and is viewed as a kernel over $R \setminus \{r\}$ indexed by $W \cup \{r\}$. This expression belongs to $\mathcal{K}$.

A *fixing-generated identification expression* is an expression obtained by taking a finite product of fixing-generated kernel expressions and marginalizing over a finite set of non-query variables, as in the identifying functional in Theorem 1. For the proof, we introduce the auxiliary expansion map below to formalize the symbolic language used by Definition 2. Thus the canonicalizer may be viewed as first expanding fixing-generated kernels into sums/integrals, products, quotients, and probability terms, and then applying the graph-free rewrite rules of Definition 2. Therefore, this auxiliary expansion map can be seen as a bridge between kernels and probability expressions.

E.2 Expansion into Symbolic Expressions

The canonicalizer is applied after fixing-generated kernels have been written in a symbolic form involving only the observed joint density. In the proof, we denote this symbolic expansion by

$$\mathrm{Exp}\{\cdot\}.$$

The notation is only used to describe the expression language precisely; it is not meant as an additional algorithmic step beyond the canonicalization procedure.

For kernel expressions, expansion is defined recursively. For the base expression,

$$\mathrm{Exp}\{p(x_V)\} = p(x_V).$$

For marginalization,

$$\mathrm{Exp}\{\mathrm{Marg}_{R\setminus A}(q_R)(x_A \mid x_W)\} = \int_{x_{R\setminus A}} \mathrm{Exp}\{q_R(x_R \mid x_W)\}.$$

For conditioning,

$$\mathrm{Exp}\{q(A \mid B, W)\} = \frac{\mathrm{Exp}\{q(A, B \mid W)\}}{\mathrm{Exp}\{q(B \mid W)\}}.$$

For fixing,

$$\mathrm{Exp}\{\phi_r(q_R)\} = \frac{\mathrm{Exp}\{q_R(x_R \mid x_W)\}}{\mathrm{Exp}\{q_R(x_r \mid x_{nd_R(r)}, x_W)\}}.$$

For an identification expression of the form

$$\int_{x_Z} \prod_{\ell=1}^{m} q_\ell, \qquad q_\ell \in \mathcal{K},$$

we define

$$\mathrm{Exp}\left\{\int_{x_Z} \prod_{\ell=1}^{m} q_\ell\right\} = \int_{x_Z} \prod_{\ell=1}^{m} \mathrm{Exp}\{q_\ell\}.$$

The result of expansion may contain nested sums, products, and quotients. In particular, a kernel produced by several marginalization, conditioning, or fixing operations generally expands to a compound symbolic expression, rather than to a single density term. Variables that are marginalized may still occur syntactically in the expanded expression, but only as bound variables under sums or integrals.

E.3 Graph-free rewrite system

Let $\equiv_{\mathrm{gf}}$ denote the smallest congruence on fixing-generated identification expressions and their expanded symbolic forms generated by the following graph-free probability identities and rewrite rules. These rules are graph-free because they only use ordinary probability identities for kernels and symbolic algebraic manipulations. They do not include equalities that rely on additional graphical constraints from a particular ADMG.

The rules are organized in two parts. The first part contains the kernel operations that occur in fixing-generated expressions, i.e., conditioning, marginalization, and fixing. The second part contains the algebraic simplifications used after expansion, such as normalization, reordering, factor extraction, and cancellation.

Conditioning.

$$q(A \mid B, W) \equiv_{\mathrm{gf}} \frac{q(A, B \mid W)}{q(B \mid W)}.$$

Marginalization.

$$\mathrm{Marg}_{R\setminus A}(q_R)(x_A \mid x_W) \equiv_{\mathrm{gf}} \int_{x_{R\setminus A}} q_R(x_R \mid x_W).$$

Fixing.

$$\phi_r(q_R) \equiv_{\text{gf}} \frac{q_R}{q_R(x_r \mid x_{nd_R(r)}, x_W)}.$$

Normalization of generated kernels. For any fixing-generated kernel expression $q_R(x_R \mid x_W) \in \mathcal{K}$,

$$\int_{x_R} q_R(x_R \mid x_W) \equiv_{\text{gf}} 1.$$

This is included as a graph-free normalization identity. Lemma 3 verifies that the canonicalizer realizes this identity after expansion into the symbolic expression.

Product, quotient, and summation rules. The relation $\equiv_{\text{gf}}$ also contains associativity and commutativity of multiplication, deterministic reordering of sums, extraction of factors independent of bound variables,

$$\int_x A(x)B \equiv_{\text{gf}} B \int_x A(x), \qquad x \notin \text{free}(B),$$

and cancellation of identical nonzero factors,

$$\frac{A \cdot B}{B \cdot C} \equiv_{\text{gf}} \frac{A}{C}.$$

E.4 Soundness and Completeness Proof

Recall that $\mathcal{C}$ denotes the canonicalization map corresponding to Definition 2. In this appendix, $\mathcal{C}$ can be viewed as a map on the expanded expression language $\mathcal{E}$ described above. Hence, fixing-generated identification kernels are first converted to their expanded representation $\text{Exp}\{\cdot\}$. The map then applies the rewrite rules from Definition 2 recursively until a fixed point is reached, yielding a deterministic graph-free normal form. It applies the rewrite rules from Definition 2, including fraction cancellation, extraction of factors independent of the summation or integration variables, trivial marginal simplification, and deterministic ordering. Bound variables are always interpreted with their proper scope, i.e., a variable that occurs only inside an inner sum or integral is not treated as free in the surrounding expression.

We first prove that the proposed canonicalizer is well-defined and terminates.

Lemma 1 (Termination)**.** *For every fixing-generated identification expression $e \in \mathcal{E}$, the canonicalizer $C(e)$ terminates and returns a unique fixed point determined only by e and by the fixed variable order $\prec$.*

Proof. Every expression in $\mathcal{E}$ is finite and is built from finitely many probability terms, products, quotients, sums, and integrals. The canonicalizer applies the rewrite rules of Definition 2 recursively, with deterministic sorting according to the fixed order $\prec$.

Termination follows because the procedure acts on a finite expression and each rewrite step can apply only finitely many times. Conditional expansion removes conditional probability terms from the representation. Quotient flattening removes nested quotient structure by rewriting it into a single numerator-denominator form. Canonical ordering sorts finite lists of bound variables, joint-density arguments, and commutative factors. After sorting, the same ordering step is idempotent. Independent-factor extraction can move a given factor across a given sum or integral binder only finitely many times, because the expression contains only finitely many factors and binders. Trivial marginal removal strictly removes marginalization operations over primitive joint-density terms. Finally, common-factor cancellation strictly removes identical numerator-denominator factor pairs. None of these operations introduces an unbounded supply of new probability terms, binders, or cancellable factors. Therefore the repeated application of the canonicalization rules reaches a fixed point after finitely many steps.

Uniqueness follows from determinism. The variable order $\prec$ fixes the ordering of variables, joint arguments, bound variables, and commutative factors. The remaining rewrite rules are applied syntactically to the finite expression and have deterministic outcomes. Thus the fixed point returned by the procedure is independent of incidental implementation choices and depends only on e and $\prec$. Hence $C(e)$ is a well-defined canonical form. □

We now prove the soundness of the proposed canonicalizer.

Theorem 3 (Soundness)**.** *For any expanded fixing-generated identification expressions $e_1, e_2 \in \mathcal{E}$, if*

$$\mathcal{C}(e_1) = \mathcal{C}(e_2),$$

then

$$e_1 \equiv_{\text{gf}} e_2.$$

Proof. We first observe that every one-step rewrite performed by $\mathcal{C}$ preserves $\equiv_{\text{gf}}$. Indeed, the rewrite rules used by $\mathcal{C}$ in Definition 2 are instances of the graph-free rules defining $\equiv_{\text{gf}}$ in Appendix E.3. Conditional expansion is exactly the conditioning rule. Fraction flattening, cancellation, and sorting of commutative products are instances of the product and quotient algebra rules, using the strict positivity assumption from Definition 2. Reordering bound variables is the deterministic ordering of sums and integrals. Extracting factors that do not contain the bound variable as a free variable is the factor-extraction rule. Finally, primitive marginal collapse is the marginalization rule applied to a joint density term.

Thus, if $e \to e'$ is a single rewrite step used by $\mathcal{C}$, then

$$e \equiv_{\text{gf}} e'.$$

Since $\equiv_{\text{gf}}$ is a congruence, the same remains true when the rewrite is applied inside a larger expression. By Lemma 1, iterating over the finite sequence of rewrite steps used to obtain the fixed point $\mathcal{C}(e)$, we get

$$e \equiv_{\text{gf}} \mathcal{C}(e).$$

Now suppose that $\mathcal{C}(e_1) = \mathcal{C}(e_2)$. Then we have

$$e_1 \equiv_{\text{gf}} \mathcal{C}(e_1) = \mathcal{C}(e_2) \equiv_{\text{gf}} e_2,$$

and hence $e_1 \equiv_{\text{gf}} e_2$. □

Lemma 2 (Expansion respects marginal construction)**.** *Let $q_R(x_R \mid x_W) \in \mathcal{K}$ and let $A \subseteq R$. Define*

$$q_A := \text{Marg}_{R \setminus A}(q_R).$$

Then

$$\mathcal{C}\left(\text{Exp}\{q_A(x_A \mid x_W)\}\right) = \mathcal{C}\left(\int_{x_{R \setminus A}} \text{Exp}\{q_R(x_R \mid x_W)\}\right).$$

Proof. By the recursive definition of the expansion map,

$$\text{Exp}\{q_A(x_A \mid x_W)\} = \text{Exp}\{\text{Marg}_{R \setminus A}(q_R)(x_A \mid x_W)\} = \int_{x_{R \setminus A}} \text{Exp}\{q_R(x_R \mid x_W)\}.$$

Thus the two arguments of $\mathcal{C}$ are identical up to deterministic ordering of bound variables. Therefore, their canonical forms are equal. □

Lemma 3 (Normalization of fixing-generated kernels)**.** *For every $q_R(x_R \mid x_W) \in \mathcal{K}$,*

$$\mathcal{C}\left(\int_{x_R} \text{Exp}\{q_R(x_R \mid x_W)\}\right) = 1.$$

Proof. We prove the statement by structural induction on the construction of q_R.

Base case. If $q_R(x_R) = p(x_R)$, then

$$\int_{x_R} \text{Exp}\{q_R(x_R)\} = \int_{x_R} p(x_R).$$

By trivial marginalization in Definition 2 over the full support of x_R, this canonicalizes to 1.

Marginalization step. Suppose

$$q_A(x_A \mid x_W) = \mathrm{Marg}_{R\setminus A}(q_R)(x_A \mid x_W).$$

Then

$$\begin{aligned}\mathcal{C}\left(\int_{x_A} \mathrm{Exp}\{q_A(x_A \mid x_W)\}\right) &= \mathcal{C}\left(\int_{x_A}\int_{x_{R\setminus A}} \mathrm{Exp}\{q_R(x_R \mid x_W)\}\right)\\ &= \mathcal{C}\left(\int_{x_R} \mathrm{Exp}\{q_R(x_R \mid x_W)\}\right)\\ &= 1.\end{aligned}$$

The first equality follows directly from Lemma 2. The second equality uses deterministic reordering of sums and the last equality follows from the induction hypothesis.

Conditioning step. Suppose

$$q(A \mid B, W) = \frac{q(A, B \mid W)}{q(B \mid W)}.$$

Then

$$\mathcal{C}\left(\int_{x_A} \mathrm{Exp}\{q(A \mid B, W)\}\right) = \mathcal{C}\left(\int_{x_A} \frac{\mathrm{Exp}\{q(A, B \mid W)\}}{\mathrm{Exp}\{q(B \mid W)\}}\right).$$

The denominator has no free occurrence of x_A, so the factor-extraction rule gives

$$\mathcal{C}\left(\frac{\int_{x_A} \mathrm{Exp}\{q(A, B \mid W)\}}{\mathrm{Exp}\{q(B \mid W)\}}\right).$$

By Lemma 2,

$$\mathcal{C}\left(\int_{x_A} \mathrm{Exp}\{q(A, B \mid W)\}\right) = \mathcal{C}\left(\mathrm{Exp}\{q(B \mid W)\}\right).$$

Thus numerator and denominator canonicalize to identical subexpressions and cancel each other. The result is 1.

Fixing step. Suppose

$$q'_{R\setminus\{r\}}(x_{R\setminus\{r\}} \mid x_W, x_r) = \phi_r(q_R) = \frac{q_R(x_R \mid x_W)}{q_R(x_r \mid x_N, x_W)},$$

where

$$N = nd_R(r) \subseteq R \setminus \{r\}.$$

Let

$$D = R \setminus (\{r\} \cup N).$$

We need to prove

$$\mathcal{C}\left(\int_{x_N, x_D} \mathrm{Exp}\{q'_{R\setminus\{r\}}(x_N, x_D \mid x_W, x_r)\}\right) = 1.$$

By the definition of conditioning,

$$q_R(x_r \mid x_N, x_W) = \frac{q_R(x_r, x_N \mid x_W)}{q_R(x_N \mid x_W)}.$$

Therefore

$$q' = q_R(x_r, x_N, x_D \mid x_W) \cdot \frac{q_R(x_N \mid x_W)}{q_R(x_r, x_N \mid x_W)}.$$

After expansion,

$$\mathrm{Exp}\{q'\} = \mathrm{Exp}\{q_R(x_r, x_N, x_D \mid x_W)\} \cdot \frac{\mathrm{Exp}\{q_R(x_N \mid x_W)\}}{\mathrm{Exp}\{q_R(x_r, x_N \mid x_W)\}}.$$

Hence

$$\mathcal{C}\left(\int_{x_N,x_D} \operatorname{Exp}\{q'\}\right)$$
$$= \mathcal{C}\left(\int_{x_N,x_D} \operatorname{Exp}\{q_R(x_r, x_N, x_D \mid x_W)\} \frac{\operatorname{Exp}\{q_R(x_N \mid x_W)\}}{\operatorname{Exp}\{q_R(x_r, x_N \mid x_W)\}}\right).$$

For fixed x_N, the quotient

$$\frac{\operatorname{Exp}\{q_R(x_N \mid x_W)\}}{\operatorname{Exp}\{q_R(x_r, x_N \mid x_W)\}}$$

has no free occurrence of x_D. It may contain variables in D as bound variables inside inner sums, but those are not free occurrences. Therefore the factor-extraction rule gives

$$\mathcal{C}\left(\int x_N \frac{\operatorname{Exp}\{q_R(x_N \mid x_W)\}}{\operatorname{Exp}\{q_R(x_r, x_N \mid x_W)\}} \int x_D \operatorname{Exp}\{q_R(x_r, x_N, x_D \mid x_W)\}\right).$$

By Lemma 2,

$$\mathcal{C}\left(\int_{x_D} \operatorname{Exp}\{q_R(x_r, x_N, x_D \mid x_W)\}\right) = \mathcal{C}\left(\operatorname{Exp}\{q_R(x_r, x_N \mid x_W)\}\right).$$

Thus the inner integral canonicalizes to the same expression as the denominator and cancels. We obtain

$$\mathcal{C}\left(\int_{x_N} \operatorname{Exp}\{q_R(x_N \mid x_W)\}\right).$$

The kernel $q_R(x_N \mid x_W)$ is the marginal of $q_R(x_R \mid x_W)$ over $R \setminus N$. By the marginalization step and the induction hypothesis, this canonicalizes to 1. Therefore the fixed kernel is normalized.

This completes the induction. □

Lemma 4 (Context compatibility)**.** *Let a and b be expanded symbolic expressions such that $\mathcal{C}(a) = \mathcal{C}(b)$. Let $\mathfrak{C}[\cdot]$ be any capture-avoiding expression context generated by finite products, quotients, sums, and integrals, with clearly scoped bound variables. Then*

$$\mathcal{C}\big(\mathfrak{C}[a]\big) = \mathcal{C}\big(\mathfrak{C}[b]\big).$$

Proof. By definition, $\mathcal{C}(e)$ is obtained by applying the rewrite rules until a fixed point is reached. Hence $\mathcal{C}$ is idempotent for every expanded expression e:

$$\mathcal{C}(\mathcal{C}(e)) = \mathcal{C}(e)$$

We prove the claim by structural induction on the context $\mathfrak{C}[\cdot]$. Equivalently, we show that the canonical form of $\mathfrak{C}[e]$ depends on e only through $\mathcal{C}(e)$.

For the base context $\mathfrak{C}[\cdot] = [\cdot]$, the claim is exactly the assumption $\mathcal{C}(a) = \mathcal{C}(b)$.

For the induction step, suppose the context is formed from a smaller context by one of the allowed operations: multiplication, division, summation, or integration. By the induction hypothesis, the subexpression produced by inserting a and the one produced by inserting b have the same canonical form. The canonicalizer is recursive: it first canonicalizes subexpressions and then applies only deterministic rules to the resulting canonical subexpressions (i.e., deterministic ordering, factor extraction, marginal collapse, and cancellation). Therefore, replacing a subexpression by another one with the same canonical form cannot change the canonical form of the larger expression.

For summation and integration contexts, the capture-avoiding condition and the clear scoping of bound variables ensure that the replacement cannot change which variables occur freely in the surrounding expression. Thus the factor-extraction and marginal-collapse rules make the same deterministic choices in both expressions.

Hence each context constructor preserves equality of canonical forms. By structural induction, we have

$$\mathcal{C}\big(\mathfrak{C}[a]\big) = \mathcal{C}\big(\mathfrak{C}[b]\big).$$

□

Lemma 5 (One-step invariance)**.** *Let $e, e' \in \mathcal{E}$ be fixing-generated identification expressions. If e' is obtained from e by one generator of $\equiv_{\mathrm{gf}}$, or by the inverse of one such generator, then*

$$\mathcal{C}(e) = \mathcal{C}(e').$$

Proof. It suffices to show that a single application of each defining rewrite rule leaves the final canonical form unchanged. The reverse direction of each rule then follows by symmetry.

For conditioning and fixing, both sides have the same expanded form under the expansion map. Product reordering, sum reordering, extraction of independent factors, and cancellation are exactly among the graph-free rewrite rules applied by the canonicalizer.

For marginalization of a fixing-generated kernel factor, the equality follows from Lemma 2. If the marginalization occurs inside a larger identification expression, Lemma 4 lifts the equality to the whole expression.

For normalization of a fixing-generated kernel factor, the equality follows from Lemma 3. If the normalized kernel occurs inside a larger expression context, Lemma 4 applies similarly.

Therefore, every one-step graph-free rewrite is absorbed by the canonicalizer. □

We now prove completeness of the proposed canonicalizer.

Theorem 4 (Completeness)**.** *Let $e_1, e_2 \in \mathcal{E}$ be expanded fixing-generated identification expressions. If*

$$e_1 \equiv_{\mathrm{gf}} e_2,$$

then

$$\mathcal{C}(e_1) = \mathcal{C}(e_2).$$

Proof. By definition, $\equiv_{\mathrm{gf}}$ is the smallest congruence closed under the graph-free rewrite rules listed in Appendix E.3. Therefore, if $e_1 \equiv_{\mathrm{gf}} e_2$, there exists a finite chain of expressions

$$e_1 = e^{(0)}, e^{(1)}, \ldots, e^{(m)} = e_2,$$

such that each step from $e^{(j)}$ to $e^{(j+1)}$ applies one of these rewrite rules, in either direction, possibly inside a larger expression context.

By Lemma 5, each such step preserves the canonical form:

$$\mathcal{C}(e^{(j)}) = \mathcal{C}(e^{(j+1)}) \qquad \text{for all } j = 0, \ldots, m-1.$$

Chaining these equalities gives

$$\mathcal{C}(e_1) = \mathcal{C}(e^{(0)}) = \mathcal{C}(e^{(1)}) = \cdots = \mathcal{C}(e^{(m)}) = \mathcal{C}(e_2).$$

Hence we obtain

$$e_1 \equiv_{\mathrm{gf}} e_2 \quad \Longrightarrow \quad \mathcal{C}(e_1) = \mathcal{C}(e_2).$$

□

Theorem 4 is a relative completeness statement for the specified graph-free rewrite system. For the fixing-generated expressions considered here, it shows that $\mathcal{C}$ identifies exactly the equalities obtainable from the rules in Appendix E.3: marginalization, conditioning, fixing, normalization, deterministic reordering, extraction of independent factors, and cancellation.

With the above ingredients, we can prove Theorem 2 as follows:

Proof of Theorem 2. Let $e_1, e_2 \in \mathcal{E}$ be fixing-generated identification expressions.

For the soundness direction, suppose that

$$\mathcal{C}(e_1) = \mathcal{C}(e_2).$$

By Theorem 3, this implies

$$e_1 \equiv_{\mathrm{gf}} e_2.$$

Table 1: Canonicalizer compression rate (remaining / original number of expressions). "Full canon" is the canonicalizer used in the paper; other columns remove the indicated step while keeping all remaining steps. Smaller values indicate stronger de-duplication.

	Full canon	w/o canceling fractions	w/o marginalization	w/o canonical ordering
Compression rate	0.0004	0.2109	0.0004	1.0000

For the completeness direction, suppose that

$$e_1 \equiv_{\mathrm{gf}} e_2.$$

By Theorem 4, this implies

$$\mathcal{C}(e_1) = \mathcal{C}(e_2).$$

Combining the two implications gives

$$\mathcal{C}(e_1) = \mathcal{C}(e_2) \quad \Longleftrightarrow \quad e_1 \equiv_{\mathrm{gf}} e_2.$$

Thus the canonicalizer is sound and complete relative to the specified graph-free rewrite system. □

F Additional Experiment Results

F.1 Empirical Evaluation of the Canonicalizer

The results in Appendix E establish that the canonicalizer $\mathcal{C}$ is sound and complete relative to the specified graph-free rewrite system. The purpose of this subsection is therefore empirical rather than theoretical: we evaluate how much $\mathcal{C}$ reduces the redundancy created by enumerating valid fixing sequences, and how important the individual canonicalization steps are for the resulting FID values.

We first quantify the de-duplication effect of $\mathcal{C}$ using the compression rate, defined as

$$\text{compression rate} = \frac{\text{number of expressions after canonicalization}}{\text{number of expressions before canonicalization}},$$

computed per query and then averaged. We compare the full canonicalizer used throughout the paper with ablated variants in which one canonicalization step is removed while all other steps are kept unchanged. We do not ablate the conditional-expansion step. This step fixes the common representation class by rewriting conditional terms into joint-density ratios. If this step is removed, expressions may remain in different syntactic languages, making the remaining ablations difficult to interpret.

Table 1 shows that the full canonicalizer removes almost all syntactic redundancy generated by the fixing-sequence enumeration. The average compression rate of 0.0004 means that only a very small fraction of the original symbolic expressions remains after canonicalization. Canonical ordering is a crucial step. Without it, the compression rate becomes 1.0000, meaning that the canonicalizer fails to merge expressions that differ only by the order of factors, variables, or integration operations. Fraction cancellation is also important, since removing it increases the compression rate from 0.0004 to 0.2109. However, removing trivial marginals does not change the compression rate in these experiments. We keep this step in our proposed canonicalizer because it is part of the graph-free simplification system and it can shorten expressions before symbolic comparison.

We next evaluate how these ablations affect the measured graph distance. For each number of randomized graph edits n_{edits}, we compute the average FID using the full canonicalizer and the same ablated variants as above.

Table 2 shows that weakening the canonicalizer can dramatically inflate the measured distance. When fraction cancellation or canonical ordering is removed, the resulting FID values are close to 1 across most edit levels. This does not indicate that the graphs are necessarily maximally different in their causal implications. Instead, it shows that many equivalent fixing-generated expressions are no longer recognized as equal. The verifier then treats syntactic variants of the same graph-free functional as distinct, which artificially reduces the overlap between expression sets. Removing the

Table 2: Effect of canonicalization ablations on measured distance for varying numbers of graph edits n_{edits}. "Full canon" denotes the canonicalizer used in the paper; other columns remove the indicated step.

n_{edits}	Full canon	w/o canceling fractions	w/o marginalization	w/o canonical ordering
1	0.1475	0.9078	0.1475	0.9440
2	0.2878	0.9950	0.2878	0.9906
3	0.2553	0.9939	0.2553	0.9154
4	0.4908	0.9794	0.4908	0.9481
5	0.4072	0.8746	0.4072	0.9943

trivial-marginal step leaves the reported distances unchanged in these experiments, matching the compression-rate results in Table 1.

These ablation studies show that the canonicalizer is not merely a cosmetic symbolic simplification step. It is a necessary normalization layer for making the fixing-based verifier practical. It deduplicates the large number of syntactic variants produced by valid fixing sequences into a much smaller set of canonical expressions for fast comparison, and it prevents the measured FID from being dominated by irrelevant symbolic representation choices.

F.2 Tabulated Summaries

Tables 3 to 6 report the *directional*, normalized FID $\bar{d}_{\mathcal{I}}(\cdot, \cdot; \mathbf{S}) \in [0, 1]$ as precise counterparts to the box/violin plots in Section 4. Higher values mean a larger fraction of treatment-outcome estimands disagree after canonicalization.

Single-edit breakdown (Tables 3 and 4). For each $(n_{\text{bi}}, p_{\text{dir}})$ setting, we apply exactly one legal edit to a reference ADMG G to obtain H and average $\bar{d}_{\mathcal{I}}$ over graphs and edit locations. Table 3 lists $\bar{d}_{\mathcal{I}}(G, H; \mathbf{S})$ and Table 4 lists the reverse direction $\bar{d}_{\mathcal{I}}(H, G; \mathbf{S})$. The per-edit-type ordering matches Figure 4: *reverse dir* and *dir to bi* are largest on average, while *del bi* and *add bi* are smallest, with slight directional asymmetry between Table 3 and Table 4.

Edit-count trends (Tables 5 and 6). These tables aggregate $\bar{d}_{\mathcal{I}}$ by the number of successful edits $k \in \{1, \ldots, 5\}$ (columns), again for both directions. They complement the middle panel of Figure 3: values reflect averages over random edit types at each k, so exact numbers need not be strictly monotone across k within every $(n_{\text{bi}}, p_{\text{dir}})$ cell. Differences between Table 5 and Table 6 quantify the inherent directionality of FID.

Pooled edit-type means (Table 7). Finally, we pool across all $(n_{\text{bi}}, p_{\text{dir}}, k)$ and report the mean and standard error of $\bar{d}_{\mathcal{I}}$ per edit type. The ranking mirrors the ridgeline plot in Figure 4: edits that reverse a direction or introduce confounding (*reverse dir*, *dir to bi*) have the largest average impact on estimands; removing bidirected edges (*del bi*) has the smallest.

F.3 Computational Efficiency and Runtime Scaling

Implementation optimizations. Computing FID involves enumerating valid fixing sequences and comparing symbolic identifying expressions, which can lead to substantial redundant computation. We therefore introduced four implementation optimizations. First, because the result of fixing a given set of nodes is invariant to the order of any valid fixing sequence, we memoize the suffix search by the subset of nodes remaining to be fixed. This avoids repeatedly copying and expanding the same search states for different order prefixes and yielded a speedup of more than $10\times$ in our profiling experiments. Second, we remove exact duplicates among district-level expressions before constructing their Cartesian product in Equation (1). This reduced redundant symbolic expansion and yielded speedups exceeding $5\times$ for five-node graphs and $300\times$ for eight-node graphs in the profiled cases. Third, we cache calls to SymPy's recursive `together` routine to avoid repeated symbolic work.

Benchmark design. We measure wall-clock time for an FID evaluation on a graph pair and a single selected treatment–outcome query (T, Y), including identification, canonicalization, and symbolic comparison.

Table 3: Single-edit FID, $G \to H$ — mean $\bar{d}_{\mathcal{I}}$ by $(n_{\mathrm{bi}}, p_{\mathrm{dir}})$ and edit type.

bi	dir	reverse	di to bi	add dir	del dir	bi to di	add bi	del bi
1	0.2	0.3278	0.1556	0.1675	0.1722	0.1817	0.0875	0.0350
1	0.3	0.3368	0.1895	0.1725	0.1855	0.2150	0.1475	0.0863
1	0.4	0.3550	0.1975	0.1625	0.1425	0.2500	0.1050	0.1200
1	0.5	0.3500	0.1875	0.1550	0.1588	0.2500	0.1875	0.1675
1	0.6	0.3725	0.1950	0.1475	0.1588	0.2700	0.1988	0.2012
1	0.7	0.3575	0.1950	0.1276	0.1012	0.3125	0.2950	0.2879
1	0.8	0.3625	0.2550	0.0861	0.1046	0.3137	0.2063	0.2975
2	0.2	0.2750	0.1222	0.1750	0.1361	0.1800	0.0400	0.0750
2	0.3	0.3325	0.1750	0.1625	0.1500	0.1912	0.0700	0.0450
2	0.4	0.3325	0.1650	0.1613	0.1375	0.2137	0.1325	0.1600
2	0.5	0.3962	0.1567	0.1425	0.1350	0.2537	0.1825	0.1779
2	0.6	0.3250	0.2125	0.1000	0.1512	0.2767	0.2612	0.2300
2	0.7	0.3237	0.1450	0.1197	0.1137	0.2937	0.2075	0.2388
2	0.8	0.3292	0.1963	0.1162	0.0971	0.2854	0.2025	0.2037
3	0.2	0.2800	0.1300	0.1400	0.1267	0.1700	0.0500	0.0900
3	0.3	0.3000	0.1350	0.1700	0.1375	0.1975	0.0850	0.0588
3	0.4	0.3342	0.1605	0.1387	0.1487	0.1525	0.0625	0.1150
3	0.5	0.3850	0.1550	0.1350	0.1488	0.2025	0.0825	0.1213
3	0.6	0.3225	0.1713	0.1150	0.1213	0.2238	0.2275	0.1300
3	0.7	0.3250	0.1675	0.0921	0.0875	0.2450	0.1400	0.1850
3	0.8	0.3050	0.1500	0.0763	0.0450	0.1450	0.2200	0.2025

Table 4: Single-edit FID, $H \to G$ — mean $\bar{d}_{\mathcal{I}}$ by $(n_{\mathrm{bi}}, p_{\mathrm{dir}})$ and edit type.

bi	dir	reverse	di to bi	add dir	del dir	bi to di	add bi	del bi
1	0.2	0.3278	0.1556	0.1692	0.1722	0.1808	0.0875	0.0375
1	0.3	0.3382	0.1868	0.1750	0.1816	0.2150	0.1475	0.0837
1	0.4	0.3550	0.1975	0.1650	0.1425	0.2475	0.1050	0.1213
1	0.5	0.3500	0.1875	0.1573	0.1575	0.2500	0.1875	0.1675
1	0.6	0.3725	0.1963	0.1500	0.1550	0.2650	0.2037	0.2025
1	0.7	0.3600	0.1950	0.1360	0.1008	0.3125	0.2950	0.2875
1	0.8	0.3612	0.2550	0.0917	0.0975	0.3100	0.2112	0.2925
2	0.2	0.2750	0.1222	0.1750	0.1361	0.1800	0.0400	0.0750
2	0.3	0.3325	0.1725	0.1625	0.1500	0.1900	0.0700	0.0400
2	0.4	0.3325	0.1650	0.1637	0.1387	0.2125	0.1325	0.1600
2	0.5	0.3937	0.1525	0.1425	0.1325	0.2500	0.1838	0.1762
2	0.6	0.3250	0.2112	0.1000	0.1475	0.2712	0.2625	0.2250
2	0.7	0.3225	0.1450	0.1184	0.1100	0.2925	0.2125	0.2350
2	0.8	0.3250	0.1900	0.1145	0.0925	0.2800	0.2050	0.1975
3	0.2	0.2800	0.1300	0.1400	0.1267	0.1700	0.0500	0.0900
3	0.3	0.3000	0.1350	0.1700	0.1375	0.1975	0.0875	0.0588
3	0.4	0.3368	0.1605	0.1375	0.1474	0.1525	0.0625	0.1150
3	0.5	0.3850	0.1525	0.1350	0.1475	0.2025	0.0825	0.1200
3	0.6	0.3225	0.1700	0.1150	0.1200	0.2225	0.2275	0.1275
3	0.7	0.3263	0.1675	0.0921	0.0875	0.2450	0.1400	0.1850
3	0.8	0.3050	0.1500	0.0789	0.0450	0.1400	0.2200	0.2025

Table 5: FID vs. edit count, $G \to H$ — mean $\bar{d}_{\mathcal{I}}$ by $(n_{\mathrm{bi}}, p_{\mathrm{dir}})$ and number of edits k.

bi	dir	1 edit	2 edits	3 edits	4 edits	5 edits
1	0.2	0.1611	0.2358	0.2750	0.2925	0.3550
1	0.3	0.1905	0.2650	0.3575	0.3675	0.3525
1	0.4	0.0904	0.2575	0.4067	0.4025	0.4850
1	0.5	0.2081	0.3204	0.4000	0.4800	0.4425
1	0.6	0.2206	0.2649	0.4379	0.5342	0.5963
1	0.7	0.2396	0.3675	0.4075	0.4975	0.5912
1	0.8	0.2323	0.2712	0.4862	0.5225	0.5775
2	0.2	0.1434	0.2225	0.2550	0.3725	0.3475
2	0.3	0.1609	0.1800	0.2925	0.4000	0.3925
2	0.4	0.1861	0.2125	0.2950	0.3850	0.4125
2	0.5	0.2064	0.2275	0.3725	0.4425	0.4963
2	0.6	0.2224	0.3100	0.3650	0.4450	0.5150
2	0.7	0.2061	0.2538	0.4075	0.5125	0.5300
2	0.8	0.2044	0.3662	0.3875	0.4700	0.4838
3	0.2	0.1410	0.1675	0.2750	0.2575	0.3550
3	0.3	0.1549	0.2225	0.2550	0.3925	0.3250
3	0.4	0.1589	0.2125	0.2475	0.3225	0.3425
3	0.5	0.1758	0.2825	0.3800	0.3875	0.4425
3	0.6	0.1874	0.2600	0.4000	0.4088	0.4850
3	0.7	0.1775	0.2425	0.3850	0.4225	0.4550
3	0.8	0.1634	0.3075	0.3438	0.4525	0.4925

Table 6: FID vs. edit count, $H \to G$ — mean $\bar{d}_{\mathcal{I}}$ by $(n_{\mathrm{bi}}, p_{\mathrm{dir}})$ and number of edits k.

bi	dir	1 edit	2 edits	3 edits	4 edits	5 edits
1	0.2	0.1616	0.2350	0.2750	0.2925	0.3550
1	0.3	0.1902	0.2662	0.3575	0.3675	0.3525
1	0.4	0.1906	0.2575	0.4000	0.4025	0.4850
1	0.5	0.2082	0.3175	0.4000	0.4775	0.4450
1	0.6	0.2208	0.2621	0.4363	0.5325	0.5963
1	0.7	0.2410	0.3625	0.4075	0.4975	0.5875
1	0.8	0.2313	0.2675	0.4862	0.5237	0.5825
2	0.2	0.1434	0.2225	0.2550	0.3725	0.3488
2	0.3	0.1597	0.1850	0.2925	0.4000	0.3925
2	0.4	0.1865	0.2125	0.2950	0.3850	0.4125
2	0.5	0.2045	0.2300	0.3725	0.4425	0.4925
2	0.6	0.2207	0.3113	0.3650	0.4375	0.5150
2	0.7	0.2052	0.2525	0.4050	0.5138	0.5312
2	0.8	0.2007	0.3650	0.3875	0.4700	0.4825
3	0.2	0.1410	0.1675	0.2750	0.2575	0.3550
3	0.3	0.1552	0.2250	0.2550	0.3925	0.3250
3	0.4	0.1589	0.2137	0.2475	0.3225	0.3425
3	0.5	0.1750	0.2825	0.3800	0.3875	0.4425
3	0.6	0.1865	0.2600	0.4000	0.4075	0.4850
3	0.7	0.1777	0.2425	0.3850	0.4225	0.4525
3	0.8	0.1631	0.3075	0.3425	0.4525	0.4925

Table 7: Pooled single-edit impact — mean $\bar{d}_{\mathcal{I}}$ (± s.e.) by edit type across all settings.

	reverse dir	dir to bi	add dir	del dir	bi to dir	add bi	del bi
mean	0.3357	0.1726	0.1376	0.1303	0.2288	0.1526	0.1531
std. error	0.0048	0.0049	0.0039	0.0037	0.0060	0.0080	0.0073

We generate random ADMGs with $n \in \{3, \dots, 8\}$ nodes, directed-edge probability $p_{\text{dir}} \in \{0.2, 0.3, \dots, 0.8\}$, and bidirected-edge probability $p_{\text{bi}} = 0.2$.

In the bidirected-edge deletion experiment, we draw 10 graphs per (n, p_{dir}) setting, perturb each graph by deleting one bidirected edge, and evaluate six randomly selected treatment–outcome pairs per graph. This gives $6 \times 7 \times 10 \times 6 = 2{,}520$ timed graph–query evaluations.

For the additional random-perturbation experiment, we consider $p_{\text{dir}} \in \{0.5, 0.6, 0.7, 0.8\}$, sample five graphs per setting, apply one randomly selected feasible edge perturbation to each graph, and evaluate six randomly selected pairs, giving $6 \times 4 \times 5 \times 6 = 720$ further evaluations.

We report the median and 95th-percentile runtime in seconds, with the latter shown in square brackets. The experiments were run on an academic Slurm-managed CPU cluster with heterogeneous architectures containing more than 10 year old Intel chips as well as more modern CPUs. Therefore, the timing results in Tables 8 and 9 primarily indicate scaling behavior and overall orders of magnitude of wallclock runtimes rather than exactly reproducible numbers. The results indicate that while the median runtime scales reasonably, with increasing graph size we increasingly encounter extreme outliers taking up to 100 times longer than the median case.

Table 8: Wall-clock runtime (seconds) by number of nodes n and perturbation type. Entries give the median [95th percentile] over the corresponding timed graph–query evaluations.

n	Delete bidirected edge	Random perturbation
3	0.0189 [0.0751]	0.0198 [0.0711]
4	0.0381 [2.33]	0.0441 [0.546]
5	0.143 [9.78]	0.132 [3.51]
6	0.653 [281]	0.813 [81.8]
7	2.98 [247]	6.54 [685]

Table 9: Wall-clock runtime (seconds) by directed-edge probability p_{dir} and graph size n. Entries give the median [95th percentile] over the corresponding timed graph–query evaluations.

p_{dir}	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$
0.2	0.0168 [0.0753]	0.0241 [0.071]	0.074 [1.17]	0.248 [3.71]	1.02 [10.6]
0.3	0.0169 [0.0603]	0.0284 [0.172]	0.0919 [1.86]	0.438 [5.29]	1.85 [208]
0.4	0.0206 [0.0808]	0.0372 [2.72]	0.094 [2.17]	0.494 [8.54]	1.18 [22.2]
0.5	0.0193 [0.0688]	0.0693 [2.46]	0.174 [7.82]	0.572 [198]	7.85 [209]
0.6	0.0189 [0.0776]	0.0417 [2.33]	0.398 [12.7]	1.19 [218]	11.4 [1400]
0.7	0.0207 [0.0723]	0.0948 [0.921]	0.239 [9.26]	2.42 [554]	10.2 [497]
0.8	0.0214 [0.0721]	0.155 [3.53]	0.922 [25.5]	2.61 [890]	7.88 [7910]

Runtime scaling. Tables 8 and 9 show that typical runtimes increase with graph size and, broadly, with directed-edge density, while variation between graph instances can be substantial. In the sparsest tested regime ($p_{\text{dir}} = 0.2$), even seven-node graphs have a median runtime of about a second and also the 95th-percentile remains around 10 seconds. In contrast, more densely connected instances can have much longer runtimes: for example, at $n = 7$ and $p_{\text{dir}} = 0.8$, the median is 7.88 seconds but the 95th percentile reaches 7,910 seconds. The observed non-monotonicity across some settings also indicates that beyond n and p_{dir}, the runtime may also depend on the district structure and the number and form of fixing-generated expressions.

G Software

We use Python van Rossum and Drake [2009] and list all used packages in Table 10. Most of them are open software, with two exceptions: `dct-policy` and `sep_distances`, which are publicly available on `github.com`, but which do not include an explicit license.

Table 10: Overview of software and packages used in this work.

Name	Reference	License
Python	van Rossum and Drake [2009]	PSF License
Numpy	Harris et al. [2020]	BSD-style license
Matplotlib	Hunter [2007]	modified PSF
Scikit-learn	Pedregosa et al. [2011]	BSD 3-Clause
SLURM	Yoo et al. [2003]	modified GNU GPL v2
networkx	Hagberg et al. [2008]	BSD 3-Clause
sep_distances	Wahl and Runge [2025]	no license
Ananke	Lee et al. [2023]	Apache 2.0
Pandas	The pandas development team [2020]	BSD-style license
dct-policy	Squires et al. [2020]	no license
SciPy	Virtanen et al. [2020]	BSD 3-Clause

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our theoretical contributions and experiments support the claims made in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have a dedicated paragraph discussing the limitation of this work, that clarify the applicability and scope of the proposed method.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might

not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: We provide the proof of proposed theorem and additional lemmas in the Appendix.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The setting (including graph generation) of our experiments are provided in a dedicated section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

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

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

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

   Answer: [Yes]

   Justification: The scripts can be found in the supplementary materials.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: Our proposed distance (FID) does not require training or testing a model. We provide our experimental setup in a dedicated section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: These results are provided in both plots and tables. In our experiments, error bars indicate standard deviations/errors around means or quantiles around medians. The specific choices are described in the main text or figure captions.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Our experiments are conducted on a Slurm-based cluster using different types of hardware. Therefore, reported wall clock times and hardware usage remain machine-dependent and primarily serve to demonstrate scaling behavior rather than exact reproducible timings.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: Our research is aligned with NeurIPS code of Ethics.

Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: As a theoretical contribution to general machine learning, more specifically graphical modeling and causality, there are no immediate relevant societal implications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This work does not provide a model, instead it's a measure of graph similarity. We believe this question does not apply.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have a dedicated section listing the software packages used in this work and their license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our scripts are documented and easy to use. Our method does not involve model training or model hyperparameter tuning.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our work is purely computational, and it does not involve those points.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: Our work is purely computational. It does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not an important component of this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.