DAG-FM: A FOUNDATION MODEL FOR CAUSAL DISCOVERY UNDER HETEROGENEOUS CAUSAL MECHANISMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal discovery from observational tabular data remains fundamentally challenging, primarily due to the heterogeneity of underlying causal mechanisms and the high-dimensional combinatorial search space of Directed Acyclic Graphs (DAGs). In this paper, we propose **DAG-FM**, a novel foundation model architecture that amortizes causal discovery. Unlike direct matrix prediction, DAG-FM decomposes the causal discovery process into two auto-regressive stages using two specialized Transformer-based sub-modules: a leaf-node predictor and a parent-node predictor. To effectively model complex row-column interactions, we adopt a robust tabular interaction block to output feature-wise representations. Crucially, to handle diverse and unknown Functional Causal Model (FCM) assumptions in real-world scenarios, we introduce Mixture-of-Leaf-Experts (MoLE), allowing the model to dynamically route and adapt to identifiable mechanism families. Through an iterative inference algorithm, DAG-FM seamlessly extracts causal orderings and constructs valid DAGs. Extensive experiments demonstrate that DAG-FM achieves state-of-the-art performance on both synthetic benchmarks and complex real-world datasets, significantly outperforming traditional classical algorithms and recent foundation models in both accuracy and scalability.

1 INTRODUCTION

Causal discovery aims to recover causal structures from data (Pearl, 2009), playing a pivotal role in diverse fields such as bioinformatics (Zhang et al., 2013), epidemiology (Vandenbroucke et al., 2016), sociology (Huber, 2024), and manufacturing (Vuković & Thalmann, 2022). A primary objective of this task is to identify the underlying Directed Acyclic Graphs (DAGs) from observational data. Existing approaches, spanning constraint-based (Spirtes & Glymour, 1991; Spirtes, 1995), score-based (Cooper & Herskovits, 1992; Chickering, 2002), and optimization-based (Zheng et al., 2018; Bello et al., 2022) methods, have been extensively developed. However, without additional assumptions, these methods can only identify causal structures up to Markov equivalence classes. Functional Causal Models (FCMs) introduce supplementary structural assumptions into Structural Causal Models (SCMs), ensuring theoretical identifiability of the underlying DAGs from observational distributions when specific conditions are met. Notable examples include Linear Non-Gaussian Acyclic Models (LiNGAMs, Shimizu et al. (2006)), Additive Noise Models (ANMs, Hoyer et al. (2008)), Heteroscedastic Noise Models (HNMs, Tagasovska et al. (2020)), and Post-Nonlinear models (PNLs, Zhang & Hyvärinen (2009)).

Building upon the theoretical guarantees of DAG identifiability, FCM-based causal discovery methods have been developed for empirical DAG recovery. However, these methods typically assume homogeneous causal mechanisms, fundamentally limiting their generalizability. To render statistical hypothesis testing tractable, most existing approaches are narrowly tailored to specific theoretical frameworks, such as LiNGAM or ANM (Shimizu et al., 2006; 2011; Peters et al., 2014; Rolland et al., 2022). Alternatively, some methods derive causal orderings by extracting specific statistical features, yet they remain effective only under highly restrictive conditions (Reisach et al., 2021; 2023). Consequently, the robustness and validity of these traditional methods under model misspecification or within highly complex real-world scenarios remain largely questionable.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In parallel, amortized algorithms have emerged for causal discovery, empirically demonstrating consistent superiority over traditional methods in complex nonlinear data regimes (Lorch et al., 2022; Dhir et al., 2025; Thompson et al., 2026; Peng et al., 2026; Li et al., 2026). These approaches frame causal discovery as a table-to-graph prediction task, leveraging a foundation model pre-trained on synthetic data generated from complex prior spaces to predict causal graphs for unseen data. However, the correctness and performance of this learning paradigm are highly sensitive to the design of the prior space. For example, in bivariate causal discovery, Dhir et al. (2025) showed that if the prior space is non-identifiable, the output is ambiguous; conversely, Montagna et al. (2025) demonstrated that models trained on well-specified, identifiable prior spaces can achieve consistent results.

To address both the limited generalizability of FCM-based methods under heterogeneous mechanisms and the lack of theoretical guarantees in amortized inference, we propose integrating these two paradigms. To this end, we introduce **DAG-FM**, a foundation model for causal discovery that intrinsically guarantees the output of strictly identifiable DAGs solely from observational data.

Specifically, our contributions bridge theory and practice: Theoretically, we establish a design condition detailing how to heterogeneously incorporate FCM assumptions into the prior space. This formulation guarantees that the amortized model converges to a unique, identifiable DAG. Empirically, we decouple the DAG identification process into two synergistic sub-tasks: *leaf-node prediction* and *parent-node prediction*, conducting large-scale pre-training for each. Together, these two learned modules constitute DAG-FM, collaborating sequentially to reconstruct the DAG.

This architecture enables highly scalable inference, allowing DAG-FM to handle datasets with up to $100{,}000$ samples or $1{,}000$ dimensions (or $30{,}000$ samples and 100 dimensions simultaneously) on a single GPU with 24GB VRAM under half-precision `BP16`. Extensive experiments demonstrate that DAG-FM achieves state-of-the-art performance, significantly outperforming traditional approaches on both synthetic and real-world datasets, while also surpassing contemporary causal discovery foundation models across multiple evaluation metrics.

In summary, our primary contributions are as follows:

- We provide a design condition for the prior space that allows for heterogeneous causal mechanisms, while theoretically ensuring that the predictions of the amortized method converge to a unique DAG from observational data.
- We develop **DAG-FM**, a foundation model for causal discovery consisting of two sub-modules, which guarantees the accurate prediction of a unique DAG directly from observational data.
- We provide extensive experiments demonstrating that DAG-FM achieves state-of-the-art performance on both synthetic and real-world benchmark tests.

2 Related Works

FCM-based Causal Discovery FCM-based causal discovery aims to recover the underlying causal structure by restricting the causal mechanisms within a SCM. Theoretically, to achieve the ultimate goal of DAG identifiability, FCM approaches impose specific structural constraints on the underlying causal mechanisms. For instance, LiNGAM (Shimizu et al., 2006) elegantly established that causal directions are uniquely identifiable under the assumptions of linear causal relationships and non-Gaussian additive noise. Subsequently, Hoyer et al. (2008) extended this identifiability to ANM with nonlinear causal mechanisms and general noise distributions. Building upon this, Zhang & Hyvärinen (2009) further generalized the identifiability conditions to post-nonlinear settings, characterizing an even broader space of identifiable models. More recently, these identifiability conditions have been further relaxed to accommodate heteroscedastic noises (Immer et al., 2023; Strobl & Lasko, 2023), while contemporary theoretical efforts (Xi et al., 2025; Chen et al., 2026) strive to generalize FCM framework to even broader conditions.

In practice, a corresponding lineage of multivariate causal discovery algorithms has been explicitly developed to translate these theoretical guarantees into empirical DAG recovery. For example, within the LiNGAM framework, algorithms such as Direct-LiNGAM (Shimizu et al., 2011) and ICA-LiNGAM (Shimizu et al., 2006) are capable of identifying the entire DAG. For ANMs, a diverse suite of methods has been deployed, encompassing RESIT (Peters et al., 2014), CAM (Bühlmann et al., 2014), SCORE (Rolland et al., 2022), DAS (Montagna et al., 2023b), and NoGAM (Montagna et al., 2023a). Similarly, addressing HNMs, recent algorithmic developments include

SkewScore (Lin et al., 2025) and HOST (Duong & Nguyen, 2022). However, these classical methods heavily premise on the homogeneity of the imposed mechanisms. Consequently, their robustness in handling heterogeneous mechanisms or severe model misspecifications remain largely uncertain.

Amortized Causal Discovery Amortized causal discovery employs a large structured-data model (LDM) to predict causal graphs directly from tabular data by treating structured data as joint distributions (Zhang et al., 2025b). This predictive model is pre-trained by sampling random causal graphs along with their corresponding observations from an artificially constructed prior space. Unlike tabular foundation models, which utilize in-context learning to predict missing entries at the cell level, amortized causal discovery operates at the variable level to infer the causal relationships.

For instance, AVICI (Lorch et al., 2022) employs an Axial Transformer to model interactions across tabular features, aggregates feature representations through max-pooling, and utilizes a bilinear head to predict the probability of edge existence in the adjacency matrix. Similarly, CSIvA (Ke et al., 2023) predicts edge probabilities autoregressively. Related approaches, such as CauScale (Peng et al., 2026) and TabCausal (Li et al., 2026), further extend this paradigm by scaling the training and inference processes to larger regimes.

However, a significant challenge with these methods is their direct modeling of the adjacency matrix, which fails to formally guarantee that the output is a DAG. BCNP (Dhir et al., 2025) and Arrow (Thompson et al., 2026) address this deficiency by decomposing the prediction task into causal ordering and an upper-triangular matrix. Other strategies focus on alternative forms of causal structures: for instance, SiCL (Zhang et al., 2025a) aims to predict Markov equivalence classes, TCD-DL (Kim et al., 2025) focuses on predicting the ancestor set for a target node, and TabOrder (Xu et al., 2026) is dedicated to predicting the causal ordering of variables.

3 PRELIMINARIES

Structural Causal Model An SCM involving d variables is defined as a triplet $\phi = (\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{U})$, where $\boldsymbol{X}$ and $\boldsymbol{U}$ denote endogenous and exogenous variables, respectively. For each $i \in I = \{1, \ldots, d\}$, the relationship is defined as $X_i = f_i(\boldsymbol{X}_{\text{pa}(i)}, U_i)$, where $\text{pa}(i) \subseteq I \setminus \{i\}$ denotes the set of parents of X_i, and f_i represents the causal mechanism. We assume the SCM is Markovian (Pearl, 2009), implying $U_j \perp\!\!\!\perp U_i$ for all $i \neq j$. The graph $\mathcal{G} = (I, \mathcal{E})$, where $(j, i) \in \mathcal{E}$ if and only if $j \in \text{pa}(i)$, is known as the causal graph and is a DAG. A topological ordering τ of this graph defines a causal ordering.

Amortized Causal Discovery Let $p(\phi)$ be a prior distribution over an artificially constructed prior space Φ of SCMs. An SCM $\phi \sim p(\phi)$ with d variables corresponds to a specific DAG $\mathcal{G}$. Observational data $\mathcal{D} = \{\boldsymbol{x}_k\}_{k=1}^n \in \mathbb{R}^{n \times d}$ is sampled from the observational distribution $p(\boldsymbol{X} \mid \phi)$. The posterior distribution of the causal graph given the observational data is defined as:

$$P(\mathcal{G} \mid \mathcal{D}) \propto \int P(\mathcal{G} \mid \phi)\, p(\mathcal{D} \mid \phi)\, p(\phi)\, d\phi. \tag{1}$$

To predict $\mathcal{G}$ given $\mathcal{D}$, an amortized model $Q_\theta(\mathcal{G} \mid \mathcal{D})$ is constructed to approximate the posterior distribution $P(\mathcal{G} \mid \mathcal{D})$. This is achieved by minimizing the $\text{KL}(P \| Q_\theta)$ divergence, which is equivalent to maximizing the log-likelihood $\mathbb{E}_{\mathcal{G},\mathcal{D}}[\log Q_\theta(\mathcal{G} \mid \mathcal{D})]$. During training, we perform large-scale sampling of random SCMs ϕ from $p(\phi)$ to generate corresponding pairs of $(\mathcal{G}, \mathcal{D})$ for pre-training the model Q_θ. During inference, the model takes $\mathcal{D}$ as input and produces $\mathcal{G}$ through either sampling or maximum a posteriori (MAP) estimation.

4 DAG-FM: THE DAG FOUNDATION MODEL

4.1 DAG IDENTIFIABILITY VIA MECHANISM FAMILIES

The asymptotic identifiability of DAGs in amortized methods hinges on whether the MAP estimator $\arg\max_{\mathcal{G}} Q_\theta(\mathcal{G} \mid \mathcal{D})$ converges to a unique ground-truth graph as the model parameters θ are optimized and the sample size $|\mathcal{D}|$ increases. By the properties of KL divergence and the Law of Large

Numbers, we have:

$$\lim_{|\mathcal{D}|\to\infty} Q_{\theta^*}(\mathcal{G}\,|\,\mathcal{D}) = \lim_{|\mathcal{D}|\to\infty} P(\mathcal{G}\,|\,\mathcal{D}) = P(\mathcal{G}\,|\,p(\boldsymbol{X})), \tag{2}$$

where $\theta^* = \arg\min_\theta \mathrm{KL}(P\|Q_\theta)$ denotes the optimal parameters under which $Q_\theta(\mathcal{G}\mid\mathcal{D})$ matches the posterior $P(\mathcal{G}\mid\mathcal{D})$. As $|\mathcal{D}|\to\infty$, the empirical distribution $\hat{p}_{\mathcal{D}}(\boldsymbol{X}) = \sum_{i=1}^{|\mathcal{D}|}\delta_{\boldsymbol{x}_i}(\boldsymbol{X})$ converges almost surely to the true observational distribution $p(\boldsymbol{X})$. Consequently, the convergence of the amortized model to a unique graph is equivalent to the uniqueness of the posterior map $\arg\max_{\mathcal{G}} P(\mathcal{G}\,|\,p(\boldsymbol{X}))$.

Definition 4.1 (Almost Sure Identifiability)**.** A DAG $\mathcal{G}$ is said to be identifiable almost surely in the posterior if there exists a graph $\mathcal{G}$ such that $P(\mathcal{G}\,|\,p(\boldsymbol{X})) = 1$.

To guarantee this, we incorporate assumptions rooted in established FCM literature, focusing on four primary classes of causal mechanisms:

Definition 4.2 (Mechanism Families $\mathcal{F}$)**.** For a mechanism $f(\boldsymbol{X}_{\mathrm{pa}(i)}, U_i)$, we define[1]:

- $\mathcal{F}_{\text{LiNGAM}}$: Specified by a vector $\boldsymbol{w}\in\mathbb{R}^d$ such that $f(\boldsymbol{X},U) = \boldsymbol{w}^\intercal\boldsymbol{X} + U$, where U_i is a non-Gaussian variable.
- $\mathcal{F}_{\text{ANM}}$: Specified by a non-linear function $g:\mathbb{R}^d\to\mathbb{R}$ such that $f(\boldsymbol{X},U) = g(\boldsymbol{X}) + U$.
- $\mathcal{F}_{\text{HNM}}$: Specified by a function $g:\mathbb{R}^d\to\mathbb{R}$ and a non-constant function $h:\mathbb{R}^d\to\mathbb{R}_+$ such that $f(\boldsymbol{X},U) = g(\boldsymbol{X}) + h(\boldsymbol{X})\,U$.
- $\mathcal{F}_{\text{PNL}}$: Specified by a function $g:\mathbb{R}^d\to\mathbb{R}$ and a monotonic non-linear function $h:\mathbb{R}\to\mathbb{R}$ such that $f(\boldsymbol{X},U) = h(g(\boldsymbol{X}) + U)$.

Assumption 4.3 (Prior Space)**.** (i) Causal sufficiency (no unobserved confounders) and faithfulness (conditional independencies imply d-separation) hold. (ii) For any $\phi = (\boldsymbol{X}, \boldsymbol{f}, p(\boldsymbol{U}))\in\Phi$, each mechanism f_i belongs to the family $\mathcal{F}_i\in\{\mathcal{F}_{\text{LiNGAM}}, \mathcal{F}_{\text{ANM}}, \mathcal{F}_{\text{HNM}}, \mathcal{F}_{\text{PNL}}\}$.

In stark contrast to traditional FCM-based methods that assume homogeneous data generation, Assumption 4.3 permits the SCM to exhibit *heterogeneous mechanisms*. That is, an SCM can be elegantly formulated as a composite mixture of mechanisms drawn from fundamentally different families. The following theoretical results guarantee that, within a prior space satisfying Assumption 4.3, the true DAG is almost surely identifiable from the posterior.

Theorem 4.4. *Under Assumption 4.3, given that Y is the effect variable, if $Y = f(\boldsymbol{X}, U)$ induces the observational distribution $p(\boldsymbol{X}, Y)$ and the family of the mechanism f is $\mathcal{F}$, then the posterior probability $P(\mathcal{F}\,|\,p(\boldsymbol{X}, Y)) = 1$.*

In essence, under the heterogeneous setup delineated in Assumption 4.3, Theorem 4.4 demonstrates that the exact nature of the mechanism family can be perfectly discriminated by the posterior, given only the observational distribution.

Theorem 4.5. *Under Assumption 4.3 and some additional identifiability conditions[2], if the SCM $\phi = (\boldsymbol{X}, \boldsymbol{f}, p(\boldsymbol{U}))$ induces the observational distribution $p(\boldsymbol{X})$ and the causal graph $\mathcal{G}$, then the posterior probability is $P(\mathcal{G}\,|\,p(\boldsymbol{X})) = 1$, i.e., $\mathcal{G}$ is identifiable almost surely.*

In other words, we establish a sufficient condition for prior space design that guarantees DAG identifiability. As long as the prior space constrains randomly sampled causal mechanisms to belong to one of the four families in Definition 4.2 (and satisfies the additional identifiability conditions), while ensuring that causal sufficiency and faithfulness hold, the DAG is guaranteed to be identifiable under the posterior distribution. Furthermore, this theoretically ensures that the amortized model will inherently converge, mapping observational data to a uniquely identifiable DAG.

4.2 Prior Space Construction

To construct a prior space Φ that is both expressive and adheres to the theoretical requirements stipulated Assumption 4.3, we design random SCMs using the following components.

[1] Here, we additionally assume non-degeneracy, i.e., the functional form cannot coincidentally degenerate into another mechanism family.

[2] These additional identifiability conditions require that condition (iv) in Assumption A.2 and Assumption A.4 hold simultaneously.

Random Graphs We employ three generative graph models: Erdős–Rényi, Barabási–Albert, and Watts–Strogatz. The number of nodes $d \in \{2, \dots, 100\}$.

Random Noise Exogenous variables are sampled from a comprehensive library of distributions: Normal, Laplace, Uniform, Student's t, Log-Normal, Gumbel, Exponential, χ^2, Beta, and Gamma. Each distribution is parameterized by randomly sampled hyper-parameters.

Random Mechanisms To implement the underlying causal mechanisms, we consider a diverse repertoire of random functions, including random linear functions, Multi-Layer Perceptrons (MLPs), periodic functions, Gaussian Processes with Random Fourier Features (GP-RFF), and Random Forests. In addition to the four strictly defined mechanism families outlined in Definition 4.2, we introduce a generalized causal mechanism formally defined as $f(\boldsymbol{X}, U)$. This is implemented by treating the exogenous noise U directly as an intrinsic input dimension during the forward pass. Although this general mechanism currently lacks a rigorous theoretical guarantee for strict DAG identifiability, we empirically observe that the implicit inductive biases of the function approximators still endow it with a non-trivial degree of practical identifiability. Consequently, for each local structure within the randomly sampled DAG, we uniformly sample from these mechanism families to instantiate the respective causal assignment.

To prevent the model from exploiting spurious statistical artifacts that could artificially trivialize DAG identification on synthetic benchmarks (Reisach et al., 2021; 2023), we enforce strict distribution normalization. Specifically, immediately following each forward inference step of the structural equation $Y = f(\boldsymbol{X}, U)$, the generated output Y is dynamically standardized to have zero mean and unit variance (Ormaniec et al., 2025).

Post-processing Adopting the data augmentation pipeline from Hollmann et al. (2025), we subject a fraction of the features to non-linear warping and discretization. Warping functions include sinh-arcsinh transformations, residual flows, and asymmetric power functions. Discretization is performed via quantile binning, uniform windowing, and ordinal encoding to simulate real-world data.

Sampling Tabular Data After generating the full feature set, we organize the observations into a tabular dataset $\mathcal{D} \in \mathbb{R}^{n \times d}$ and shuffle the feature order. Unlike Hollmann et al. (2025); Qu et al. (2026), we perform no feature dropout to strictly satisfy the causal sufficiency assumption. While the training sample size is fixed at $n = 1024$, our amortized model is designed to generalize to varying sample sizes during inference.

4.3 Decomposition of DAG Identification

While Theorem 4.5 guarantees the existence of a unique identifiable DAG under Assumption 4.3, existing amortized methods like Lorch et al. (2022) often parameterize the adjacency matrix directly. This approach faces two primary hurdles: the lack of formal DAG constraints and the quadratic scaling of the output space with respect to dimensionality, which impedes scalability to high-dimensional datasets. Following Dhir et al. (2025); Thompson et al. (2026), we decompose the DAG identification into two sub-problems: *order identification* and *DAG pruning*, formulated as:

$$P(\mathcal{G} \,|\, p(\boldsymbol{X})) = \sum_{\tau} P(\mathcal{G} \,|\, \tau, p(\boldsymbol{X}))\, P(\tau \,|\, p(\boldsymbol{X})). \tag{3}$$

Each sub-problem is further decomposed into recursive prediction tasks.

Amortized Order Identification via Leaf Prediction Inspired by order-based causal discovery, we identify the causal ordering τ by recursively uncovering the leaves of the DAG. The task of identifying a leaf node i from a set of variables $\boldsymbol{X}$ is framed as $\arg\max_i P(L_i = 1 \,|\, p(\boldsymbol{X}))$, where the event L_i indicates whether i is a leaf node in the DAG. This transformation reduces the ordering problem to a sequence of binary classification tasks.

Amortized DAG Pruning via Parent Prediction Given a causal ordering τ, the DAG skeleton is constrained to an upper triangular structure. To avoid the quadratic growth of predicting the entire adjacency matrix, we recursively prune the graph by identifying the parents of each leaf node.

Specifically, the identification of a parent j for a confirmed leaf node i is framed as $\arg\max P(E_{ji} | L_i = 1, p(\boldsymbol{X}))$, where E_{ji} denotes the presence of a directed edge $(j \to i)$. This process remains a tractable series of binary classification tasks.

Mixture of Experts for Leaf Prediction Since our approach relies on the theoretical conditions of FCM-based causal discovery under Assumption 4.3, we analogously draw inspiration from classical FCM practices. Empirically, discovering varying mechanism families requires developing distinct, tailored algorithms; these identification methods exhibit specific focal preferences and generally operate flawlessly only under strict assumption matching. Motivated by this, to effectively handle heterogeneous causal mechanisms, we design a novel framework denoted as *Mixture-of-Leaf-Expert (MoLE)*. This design encourages different experts to dynamically attend to distinct data features, thereby enhancing the holistic predictive capability of the model across each mechanism family.

We refine the leaf node prediction process into two hierarchical steps: (i) identifying the underlying mechanism family; (ii) predicting the leaf node conditioned on that family. This is formalized as:

$$P(L_i = 1 | p(\boldsymbol{X})) = \sum_{\mathcal{F}_i} P(L_i = 1 | \mathcal{F}_i, p(\boldsymbol{X}))\, P(\mathcal{F}_i | p(\boldsymbol{X})), \tag{4}$$

where, according to Theorem 4.4, $P(\mathcal{F}_i | p(\boldsymbol{X}))$ effectively acts as a degenerate distribution. This framework offers three primary advantages: First, it modularizes the amortized inference process by assigning specialized experts to distinct mechanism families. Second, it provides a degree of interpretability by explicitly surfacing the inferred mechanism class and its corresponding theoretical foundation. Third, this framework enables human intervention: one may manually adjust $P(\mathcal{F}_i | p(\boldsymbol{X}))$ to incorporate external expert priors, moving beyond the "black-box" nature of fully automated amortized inference.

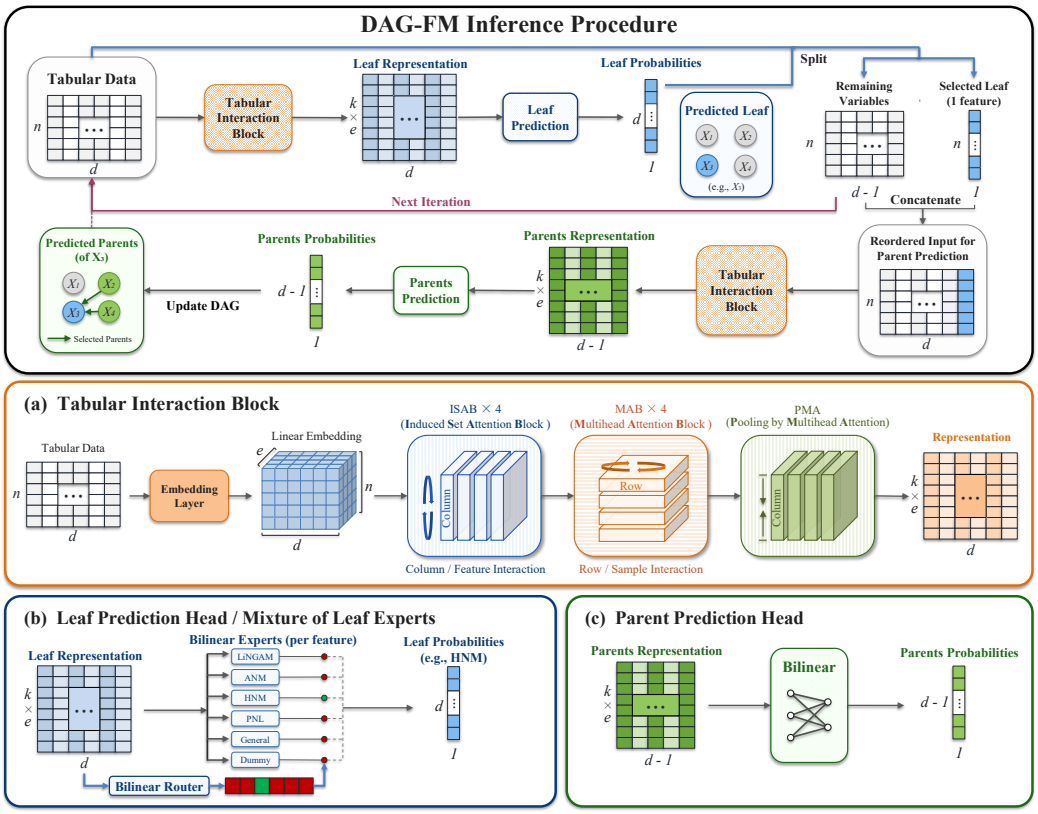


Figure 1: Architecture and inference procedure of DAG-FM. (a) Tabular Interaction Block; (b) Leaf Prediction Head; (c) Parent Prediction Head.

4.4 FOUNDATION MODEL ARCHITECTURE

DAG-FM employs two Transformer-based sub-modules, $Q_{\theta_1}(\boldsymbol{L}\,|\,\mathcal{D})$ and $Q_{\theta_2}(\boldsymbol{E}_{:,i}\,|\,L_i, \mathcal{D})$, to amortize the processes of leaf-node prediction and parent-node prediction, respectively. The architecture is illustrated in Figure 1.

Tabular Interaction Block Unlike previous approaches (Lorch et al., 2022; Peng et al., 2026; Li et al., 2026) that utilize Axial Transformers to alternate between row and column interactions or rely on max-pooling for feature-wise representations, we leverage the backbone architecture from TabICLv2 (Qu et al., 2026). The block is composed of four cascaded modules: a linear embedding layer, a Set Transformer with four ISAB blocks (Lee et al., 2019) for row-wise (sample) interaction, a Transformer with four MABs (Vaswani et al., 2017) for column-wise (feature) interaction, and a PMA (Lee et al., 2019) to aggregate row-specific information. As shown in Figure 1 (a), this transformation yields a representation of size $k \times e$ for each feature, where k denotes the number of seed vectors in the PMA and e the embedding dimension.

Algorithm 1 DAG-FM Inference Procedure

Require: Tabular data $\mathcal{D}$, threshold α
Ensure: Causal DAG $\mathcal{G}$, causal ordering τ

$I \leftarrow \{1, \ldots, d\}$, $\tau \leftarrow []$, $\mathcal{E} \leftarrow \emptyset$
while $|I| > 0$ **do**
  $l \leftarrow \arg\max_{i \in I} Q_{\theta_1}(L_i\,|\,\mathcal{D}_I)$
  $S \leftarrow \{j \in I \setminus \{l\} \mid Q_{\theta_2}(E_{jl}\,|\,L_l, \mathcal{D}_I) > \alpha\}$
  for $s \in S$ **do**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(s, l)\}$
  end for
  $I \leftarrow I \setminus \{l\}$, $\tau \leftarrow [\tau, l]$
end while
return $\mathcal{G}, \tau$

Leaf Prediction Head We integrate the MoLE framework into the leaf prediction head. It consists of $m+1$ bilinear heads corresponding to the mechanism families in our prior space, outputting both routing probabilities and family-specific expert predictions. Following Lorch et al. (2022), each bilinear head is parameterized as $\text{sigmoid}(\frac{1}{E}\sum_{i=1}^{E} \boldsymbol{u}_i^\intercal \boldsymbol{v}_i + \boldsymbol{b}_i)$, where $\boldsymbol{u}, \boldsymbol{v}$ are generated by linear and LayerNorm layers. We apply a top-k strategy (setting $k = 1$) to weight the expert predictions for the final leaf probability.

In practice, in addition to the four mechanism families specified in Assumption 4.3, two supplementary experts are included: (i) An expert specifically dedicated to the aforementioned general mechanism; and (ii) As dummy expert designed to handle source nodes (since source nodes consist strictly of pure noise and can thus be classified into any mechanism family), as well as any other potential unseen cases or scenarios that cannot even be categorized under the general mechanism.

Parent Prediction Head We also employ a bilinear head followed by a sigmoid activation to predict the probability of a node being a parent. However, to enable the model to distinguish between the queried entities (all potential parent nodes) and the query target (the leaf node) during tabular interactions, we position the known leaf node as the first feature input and incorporate an additional learnable type token (Devlin et al., 2019) during the embedding stage. At inference time, nodes with a predicted probability exceeding the threshold $\alpha = 0.5$ are identified as parent nodes.

Training Procedure The base models are optimized via the following amortized objectives:

$$\arg\min_{\theta_1} -\mathbb{E}_{\mathcal{D},\mathcal{G}}\left[\log Q_{\theta_1}(\boldsymbol{L}\,|\,\mathcal{D})\right], \quad \arg\min_{\theta_2} -\mathbb{E}_{\mathcal{D},\mathcal{G},L_i}\left[\log Q_{\theta_2}(\boldsymbol{E}_{:,i}\,|\,L_i, \mathcal{D})\right], \tag{5}$$

where $\mathcal{D}, \mathcal{G}$ are sampled from the prior space, and $\boldsymbol{L}$ is the binary label vector of leaf nodes. Within MoLE, the leaf log-likelihood is factored as:

$$\log Q_{\theta_1}(\boldsymbol{L}\,|\,\mathcal{D}) = \log \sum_{\mathcal{F}_i} \left(Q_{\theta_1}(\boldsymbol{L}\,|\,\mathcal{F}_i, \mathcal{D}) \cdot Q_{\theta_1}(\mathcal{F}_i\,|\,\mathcal{D})\right), \tag{6}$$

where $Q_{\theta_1}(\mathcal{F}_i\,|\,\mathcal{D})$ denotes the routing probability. In practice, we use Binary Cross-Entropy (BCE) loss to optimize both leaf-node prediction and parent-node prediction, while training the router with the known mechanism-family labels using Cross-Entropy loss.

Inference Procedure The DAG identification procedure is detailed in Algorithm 1.

5 EXPERIMENTS

Baselines Our experiments encompass the following causal discovery baselines: (i) FCM-based methods: Direct-LiNGAM (Shimizu et al., 2011), ICA-LiNGAM (Shimizu et al., 2006), CAM (Bühlmann et al., 2014), RESIT (Peters et al., 2014), SCORE (Rolland et al., 2022), DAS (Montagna et al., 2023b), NoGAM (Montagna et al., 2023a), HOST (Duong & Nguyen, 2023), CaPS (Xu et al., 2024), and SkewScore (Lin et al., 2025); (ii) Amortized algorithms: AVICI (Lorch et al., 2022), CauScale (Peng et al., 2026), TabCausal (Li et al., 2026), and FoundCause (Blöbaum et al., 2026).

Metrics For the DAG identification task, we report edge-level precision (`Prec.`), recall (`Rec.`), F1 score (`F1`), Jaccard index (`Jac.`), and normalized Structural Hamming Distance (`nSHD`). To emphasize the capability of discovering DAG structures rather than merely predicting adjacency matrices, we add an extra penalty of 1 to the SHD if the predicted graph contains reversed edges, and an additional 1 if it contains cycles. For the causal order identification task, we report the Order Divergence Rate (`ODR`) (Rolland et al., 2022).

Synthetic Benchmark Experiments First, we construct a synthetic benchmark, **Hetero**, by sampling 100 distinct SCMs from the prior space (see Section B.2 for detailed construction) to evaluate the DAG identification performance of the aforementioned methods under heterogeneous causal mechanisms. The results are presented in Table 1. On the DAG identification task, DAG-FM significantly outperforms several concurrently proposed causal discovery foundation models, achieving the best performance across all metrics.

DAG Performance on Synthetic Heterogeneous Mechanism Benchmark ($n = 1000, d = 20$)

Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓
AVICI (Lorch et al., 2022)	$0.58_{0.19}$	$0.30_{0.19}$	$0.37_{0.20}$	$0.25_{0.16}$	$0.23_{0.21}$	$0.06_{0.03}$
CauScale (Peng et al., 2026)	$0.57_{0.18}$	$0.34_{0.19}$	$0.41_{0.19}$	$0.27_{0.16}$	$0.23_{0.21}$	$0.07_{0.00}$
TabCausal (Li et al., 2026)	$\underline{0.63}_{0.18}$	$0.50_{0.19}$	$0.54_{0.17}$	$0.39_{0.17}$	$\underline{0.21}_{0.21}$	$0.01_{0.00}$
FoundCause (Blöbaum et al., 2026)	$0.57_{0.22}$	$\mathbf{0.66}_{0.18}$	$\underline{0.57}_{0.16}$	$\underline{0.42}_{0.16}$	$0.23_{0.17}$	$1.51_{0.01}$
DAG-FM (ours)	$\mathbf{0.72}_{0.15}$	$\underline{0.63}_{0.20}$	$\mathbf{0.66}_{0.16}$	$\mathbf{0.51}_{0.18}$	$\mathbf{0.18}_{0.17}$	$0.48_{0.01}$

Table 1: DAG identification performance on the synthetic heterogeneous mechanism benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD` and `Time`), compared with other amortized algorithms. Subscripts denote 95% confidence intervals.

Next, we construct additional synthetic benchmarks in the same manner from subspaces of specific causal mechanism families (see Section B.2 for details) to demonstrate the accuracy of DAG-FM in identifying causal orders under homogeneous causal mechanisms. As shown in Table 2, on the causal order identification task, DAG-FM is second only to Direct-LiNGAM on LiNGAM, while significantly surpassing other FCM-based algorithms across all other settings.

Real-World Benchmark Experiments Finally, we apply DAG-FM to real-world benchmarks to evaluate its generalization capability. We select two benchmarks: (i) Sachs (Sachs et al., 2005): A protein signaling dataset containing 7,466 samples and 11 variables; (ii) Causal Chamber (Gamella et al., 2025): A diverse set of physical causal systems containing 10,000 samples across 38 variables, from which we use only the 20 non-constant variables.

As shown in Table 3, DAG-FM is capable of processing these datasets, whereas several baseline algorithms cannot directly handle the combination of high dimensionality and large sample size. DAG-FM outperforms contemporary causal discovery foundation models on multiple metrics and achieves the best performance on Sachs. These results indicate that DAG-FM generalizes well to real-world settings and support its practical applicability.

Additional Results In Appendix C, we provide additional experimental results under larger sample sizes, higher dimensions, various perturbations, and out-of-distribution (OOD) settings to further demonstrate the robustness and generalization of DAG-FM. Furthermore, Appendix D presents ablation studies on the MoLE framework to highlight its effectiveness, along with more comprehensive experimental evaluations covering a broader range of baseline models.

Order Performance on All Synthetic Benchmarks ($n = 1000, d = 20$)						
Method	**LiNGAM**	**ANM**	**HNM**	**PNL**	**General**	**Hetero**
ICA-LiNGAM (Shimizu et al., 2006)	$0.18_{0.15}$	$0.50_{0.13}$	$0.49_{0.16}$	$0.47_{0.12}$	$0.49_{0.12}$	$0.48_{0.12}$
Direct-LiNGAM (Shimizu et al., 2011)	$\mathbf{0.02}_{0.03}$	$0.41_{0.13}$	$0.41_{0.15}$	$0.40_{0.14}$	$0.52_{0.13}$	$0.38_{0.15}$
CAM (Bühlmann et al., 2014)	$0.63_{0.18}$	$0.30_{0.12}$	$0.47_{0.18}$	$0.42_{0.12}$	$\underline{0.27}_{0.13}$	$0.40_{0.13}$
RESIT (Peters et al., 2014)	$0.26_{0.15}$	$\underline{0.21}_{0.13}$	$0.44_{0.15}$	$\underline{0.35}_{0.13}$	$0.41_{0.13}$	$0.37_{0.12}$
SCORE (Rolland et al., 2022)	$0.36_{0.14}$	$0.29_{0.14}$	$0.33_{0.16}$	$0.38_{0.14}$	$0.35_{0.16}$	$0.35_{0.15}$
DAS (Montagna et al., 2023b)	$0.36_{0.14}$	$0.29_{0.14}$	$0.33_{0.16}$	$0.38_{0.14}$	$0.35_{0.16}$	$0.35_{0.15}$
NoGAM (Montagna et al., 2023a)	$0.33_{0.13}$	$0.28_{0.14}$	$\underline{0.32}_{0.15}$	$0.37_{0.14}$	$0.33_{0.15}$	$\underline{0.34}_{0.14}$
HOST (Duong & Nguyen, 2023)	$0.66_{0.15}$	$0.58_{0.14}$	$0.55_{0.15}$	$0.51_{0.13}$	$0.60_{0.13}$	$0.57_{0.14}$
CaPS (Xu et al., 2024)	$0.35_{0.15}$	$0.30_{0.14}$	$\underline{0.32}_{0.16}$	$0.37_{0.14}$	$0.35_{0.16}$	$0.35_{0.15}$
SkewScore (Lin et al., 2025)	$0.54_{0.14}$	$0.55_{0.16}$	$0.53_{0.17}$	$0.54_{0.13}$	$0.52_{0.14}$	$0.50_{0.15}$
DAG-FM (ours)	$\underline{0.08}_{0.08}$	$\mathbf{0.14}_{0.09}$	$\mathbf{0.16}_{0.10}$	$\mathbf{0.14}_{0.10}$	$\mathbf{0.20}_{0.11}$	$\mathbf{0.15}_{0.10}$

Table 2: Causal order identification performance across all synthetic benchmarks (lower is better for ODR), compared with other FCM-based methods. Here, the benchmarks **LiNGAM**, **ANM**, **HNM**, and **PNL** represent homogeneous causal mechanisms, **General** denotes the general causal mechanisms in Section 4.2, and **Hetero** corresponds to the benchmark in Table 1. Subscripts denote 95% confidence intervals.

DAG Performance on Real-world Benchmarks												
Method	**Sachs** ($n = 7466, d = 11$)						**Causal Chamber** ($n = 10\text{k}, d = 20$)					
	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓
AVICI (Lorch et al., 2022)	Out of Memory						Out of Memory					
CauScale (Peng et al., 2026)	Out of Memory						Out of Memory					
TabCausal (Li et al., 2026)	0.38	<u>0.30</u>	<u>0.33</u>	<u>0.20</u>	0.44	0.56	<u>0.71</u>	<u>0.51</u>	<u>0.60</u>	<u>0.43</u>	<u>0.14</u>	0.59
FoundCause (Blöbaum et al., 2026)	<u>0.50</u>	0.25	<u>0.33</u>	<u>0.20</u>	<u>0.36</u>	7.32	0.53	**0.59**	0.56	0.39	0.19	9.73
DAG-FM (ours)	**0.58**	**0.35**	**0.44**	**0.28**	**0.33**	1.19	**0.80**	<u>0.51</u>	**0.62**	**0.45**	**0.13**	1.81

Table 3: DAG identification performance on two real-world benchmarks (higher is better for Prec., Rec., F1, and Jac.; lower is better for nSHD), compared with other amortized algorithms.

6 Conclusion

In this work, we presented **DAG-FM**, a highly adaptable and scalable foundation model for causal discovery from observational tabular data. By moving away from direct, monolithic adjacency matrix prediction, DAG-FM reframes causal DAG identification as a sequential prediction process. It employs two amortized Transformer sub-modules to iteratively identify leaf nodes and their respective parents, ensuring the structural validity of the generated causal graph. A key contribution of our architecture is the integration of the MoLE framework, which effectively bridges the gap between theoretical FCM assumptions and practical data by dynamically routing predictions across mixed, diverse, heterogeneous causal mechanisms.

Extensive empirical evaluations confirm the superiority of DAG-FM. On synthetic benchmarks featuring heterogeneous causal mechanisms, DAG-FM outperforms existing amortized baselines, including powerful concurrent causal discovery foundation models, in DAG recovery. Moreover, DAG-FM significantly surpasses traditional FCM-based methods in causal order identification across both heterogeneous and homogeneous causal mechanisms. Furthermore, its state-of-the-art performance on two real-world benchmarks underscores its robust generalizability and capacity to handle high-dimensional, large-sample settings where traditional algorithms often struggle.

Future Work While DAG-FM demonstrates formidable capabilities in causal discovery under the assumption of causal sufficiency, extending this framework to handle unobserved latent confounders and missing data presents an exciting avenue for future research. Additionally, scaling the foundation model with broader and more diverse pre-training corpora could further enhance its zero-shot inference capabilities across interdisciplinary scientific domains. Finally, as DAG-FM is currently tailored for observational causal discovery, extending its theoretical identifiability guarantees and framework to interventional settings represents another promising direction.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, language polishing, and the generation of scientific figures. We did not use generative AI tools for any other tasks that are required or recommended to be disclosed. We reviewed and verified all AI-assisted work, including the accuracy of the translation, the language editing, and the consistency of the generated figures. We take full responsibility for the final content of this work, including all text, claims, and scientific figures produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the complete assumptions and proofs for the theoretical results in Appendix A, and a detailed description of the experimental settings in Appendix B.

REFERENCES

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization. *Advances in Neural Information Processing Systems*, 35:8226–8239, 2022.

Patrick Blöbaum, Krishnakumar Balasubramanian, and Shiva Prasad Kasiviswanathan. Foundcause: Causal discovery with latent confounders from observational data, 2026.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014. ISSN 00905364.

Yikang Chen, Xingzhe Sun, and Dehui du. Causal discovery via quantile partial effect. In *The Fourteenth International Conference on Learning Representations*, 2026.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9(4):309–347, October 1992.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Anish Dhir, Matthew Ashman, James Requeima, and Mark van der Wilk. A meta-learning approach to bayesian causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Bao Duong and Thin Nguyen. Bivariate causal discovery via conditional divergence. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 236–252. PMLR, 11–13 Apr 2022.

Bao Duong and Thin Nguyen. Heteroscedastic causal structure learning. In *Frontiers in Artificial Intelligence and Applications*, Frontiers in artificial intelligence and applications. IOS Press, September 2023.

Juan L Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for AI methodology. *Nat. Mach. Intell.*, 7(1):107–118, January 2025.

Michaela Hardt, William Roy Orchard, Patrick Blöbaum, Elke Kirschbaum, and Shiva Kasiviswanathan. The petshop dataset — finding causes of performance issues across microservices. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pp. 957–978. PMLR, 01–03 Apr 2024.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

Martin Huber. An introduction to causal discovery. *Schweiz. Z. Volkswirtsch. Stat.*, 160(1), October 2024.

Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. Root cause analysis of failures in microservices through causal discovery. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 31158–31170. Curran Associates, Inc., 2022.

Alexander Immer, Christoph Schultheiss, Julia E Vogt, Bernhard Schölkopf, Peter Bühlmann, and Alexander Marx. On the identifiability and estimation of causal location-scale noise models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14316–14332. PMLR, 23–29 Jul 2023.

Nan Rosemary Ke, Silvia Chiappa, Jane X Wang, Jorg Bornschein, Anirudh Goyal, Melanie Rey, Theophane Weber, Matthew Botvinick, Michael Curtis Mozer, and Danilo Jimenez Rezende. Learning to induce causal structure. In *International Conference on Learning Representations*, 2023.

Jang-Hyun Kim, Claudia Skok Gibbs, Sangdoo Yun, Hyun Oh Song, and Kyunghyun Cho. Large-scale targeted cause discovery via learning from simulated data. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019.

Zi-Rong Li, Si-Yang Liu, Tian-Zuo Wang, and Han-Jia Ye. Tabcausal: Pretraining across causal environments for tabular causal discovery, 2026.

Yingyu Lin, Yuxing Huang, Wenqin Liu, Haoran Deng, Ignavier Ng, Kun Zhang, Mingming Gong, Yian Ma, and Biwei Huang. A skewness-based criterion for addressing heteroscedastic noise in causal discovery. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 13104–13118. Curran Associates, Inc., 2022.

Takashi Nicholas Maeda and Shohei Shimizu. Rcd: Repetitive causal discovery of linear non-gaussian acyclic models with latent confounders. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 735–745. PMLR, 26–28 Aug 2020.

Giovanni Menegozzo, Diego Dall'Alba, and Paolo Fiorini. Cipcad-bench: Continuous industrial process datasets for benchmarking causal discovery methods. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 2124–2131, 2022. doi: 10.1109/CASE49997.2022.9926420.

Francesco Montagna, Nicoletta Noceti, Lorenzo Rosasco, Kun Zhang, and Francesco Locatello. Causal discovery with score matching on additive models with arbitrary noise. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing (eds.), *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pp. 726–751. PMLR, 11–14 Apr 2023a.

Francesco Montagna, Nicoletta Noceti, Lorenzo Rosasco, Kun Zhang, and Francesco Locatello. Scalable causal discovery with score matching. In Mihaela van der Schaar, Cheng Zhang, and Dominik Janzing (eds.), *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213 of *Proceedings of Machine Learning Research*, pp. 752–771. PMLR, 11–14 Apr 2023b.

Francesco Montagna, Max Cairney-Leeming, Dhanya Sridhar, and Francesco Locatello. Demystifying amortized causal discovery with transformers, 2025.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17943–17954. Curran Associates, Inc., 2020.

Weronika Ormaniec, Scott Sussex, Lars Lorch, Bernhard Schölkopf, and Andreas Krause. Standardizing structural causal models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Bo Peng, Sirui Chen, Jiaguo Tian, Yu Qiao, and Chaochao Lu. Causcale: Neural causal discovery at scale. In *Forty-third International Conference on Machine Learning*, 2026.

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(58):2009–2053, 2014.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. 2026.

Alexander Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27772–27784. Curran Associates, Inc., 2021.

Alexander Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 785–807. Curran Associates, Inc., 2023.

Paul Rolland, Volkan Cevher, Matthäus Kleindessner, Chris Russell, Dominik Janzing, Bernhard Schölkopf, and Francesco Locatello. Score matching enables causal discovery of nonlinear additive noise models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18741–18753. PMLR, 17–23 Jul 2022.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005. doi: 10.1126/science.1105809.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006.

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvärinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, and Kenneth Bollen. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12 (33):1225–1248, 2011.

Peter Spirtes. Directed cyclic graphical representations of feedback models. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pp. 491–498, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Eric V. Strobl and Thomas A. Lasko. Identifying patient-specific root causes with the heteroscedastic noise model. *Journal of Computational Science*, 72:102099, 2023. ISSN 1877-7503.

Natasa Tagasovska, Valérie Chavez-Demoulin, and Thibault Vatter. Distinguishing cause from effect using quantiles: Bivariate quantile causal discovery. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9311–9323. PMLR, 13–18 Jul 2020.

Ryan Thompson, He Zhao, Daniel M. Steinberg, and Edwin V. Bonilla. Arrow: A foundation model for causal discovery, 2026.

Jan P Vandenbroucke, Alex Broadbent, and Neil Pearce. Causality and causal inference in epidemiology: the need for a pluralistic approach. *International Journal of Epidemiology*, 45(6): 1776–1786, 12 2016. ISSN 0300-5771. doi: 10.1093/ije/dyv341.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Matej Vuković and Stefan Thalmann. Causal discovery in manufacturing: A structured literature review. *Journal of Manufacturing and Materials Processing*, 6(1), 2022. ISSN 2504-4494. doi: 10.3390/jmmp6010010.

Johnny Xi, Hugh Dance, Peter Orbanz, and Benjamin Bloem-Reddy. Distinguishing cause from effect with causal velocity models. In *Forty-second International Conference on Machine Learning*, 2025.

Sascha Xu, Sarah Mameche, and Jilles Vreeken. Learning causal orderings for in-context tabular prediction, 2026.

Zhuopeng Xu, Yujie Li, Cheng Liu, and Ning Gui. Ordering-based causal discovery for linear and nonlinear relations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 4315–4340. Curran Associates, Inc., 2024.

Naiyu Yin, Tian Gao, Yue Yu, and Qiang Ji. Effective causal discovery under identifiable heteroscedastic noise model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16486–16494, 2024.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7154–7163. PMLR, 09–15 Jun 2019.

Bin Zhang, Chris Gaiteri, Liviu-Gabriel Bodea, Zhi Wang, Joshua McElwee, Alexei A. Podtelezhnikov, Chunsheng Zhang, Tao Xie, Linh Tran, Radu Dobrin, Eugene Fluder, Bruce Clurman, Stacey Melquist, Manikandan Narayanan, Christine Suver, Hardik Shah, Milind Mahajan, Tammy Gillis, Jayalakshmi Mysore, Marcy E. MacDonald, John R. Lamb, David A. Bennett, Cliona Molony, David J. Stone, Vilmundur Gudnason, Amanda J. Myers, Eric E. Schadt, Harald Neumann, Jun Zhu, and Valur Emilsson. Integrated systems approach identifies genetic nodes and networks in late-onset alzheimer's disease. *Cell*, 153(3):707–720, 2013. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2013.03.030.

Jiaru Zhang, Rui Ding, Qiang Fu, Huang Bojun, zizhen Deng, Yang Hua, Haibing Guan, Shi Han, and Dongmei Zhang. Learning identifiable structures avoids bias in DNN-based supervised causal learning. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025a.

Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pp. 647–655, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, Shuyang Li, Tianyang Zhang, Yue He, Yuanrui Wang, Yunjia Zhang, Zijing Xu, Dongzhe Li, Fang Gao, Hao Zou, Jiandong Liu, Jiashuo Liu, Jiawei Xu, Kaijie Cheng, Kehan Li, Linjun Zhou, Qing Li, Shaohua Fan, Xiaoyu Lin, Xinyan Han, Xuanyue Li, Yan Lu, Yuan Xue, Yuanyuan Jiang, Zimu Wang, Zhenlei Wang, and Peng Cui. Limix: Unleashing structured-data modeling capability for generalist intelligence, 2025b.

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 PROOFS

We provide the proofs for the theories in the main text under the identifiability framework based on Quantile Partial Effects (QPE) (Chen et al., 2026). This framework decouples the functional descriptions in Assumption 4.3 into a form more conducive to analysis from the observational distribution, thereby circumventing the need to specify the noise terms and exact causal mechanisms. Let $F(y \mid \boldsymbol{x})$ denote the conditional cumulative distribution function (CDF) corresponding to the conditional density function $p(y|\boldsymbol{x})$, and $Q(y|\boldsymbol{x})$ denote the conditional quantile function. Furthermore, for all probability density functions $p(\boldsymbol{x})$ appearing in our analysis, we assume they always exist, are strictly positive, and are at least of class C^k when their k-th order derivatives are involved. We define the Quantile Partial Effect as follows:

Definition A.1 (Quantile Partial Effect)**.** The QPE of a random variable Y given $\boldsymbol{X}$ is formulated as $\boldsymbol{\psi}(y|\boldsymbol{x}) = \nabla_{\boldsymbol{x}} Q(y|\boldsymbol{x})$, where the corresponding quantile is given by $\tau = Q^{-1}(y|\boldsymbol{x}) = F(y|\boldsymbol{x})$.

The QPE defined in Definition A.1 is a quantity that depends solely on the observational distribution and is uniquely determined by it. In the subsequent analysis, the QPE $\boldsymbol{\psi}(y \mid \boldsymbol{x})$ is expressed as a finite linear span with respect to y:

$$\boldsymbol{\psi}(y|\boldsymbol{x}) = \sum_{i=1}^{m} c_i(\boldsymbol{x})\, b_i(y), \tag{7}$$

where the finite set of functions $\boldsymbol{b} = (b_1, \ldots, b_m)$, referred to as *basis functions*, depends only on y, while the *coefficient functions* c_i depend only on $\boldsymbol{x}$. According to the derivations in (Chen et al., 2026), the QPEs of the observational distributions corresponding to the four mechanism cases in Assumption 4.3 can all be decoupled into this form:

FCM	Functional Form	QPE Form	QPE Basis Functions
LiNGAM	$Y = \boldsymbol{w}^\intercal \boldsymbol{X} + U$	$\boldsymbol{w}$	1
ANM	$Y = g(\boldsymbol{X}) + U$	∇g	1
HNM	$Y = g(\boldsymbol{X}) + h(\boldsymbol{X})\, U$	$\left(\nabla g - \frac{g}{h}\nabla h\right) + \frac{\nabla h}{h}\, y$	$1, y$
PNL	$Y = h(g(\boldsymbol{X}) + U)$	$\nabla g\, \overline{h}$	$\overline{h}$

Table 4: The QPEs and their basis functions corresponding to the four mechanism families. Here, $h > 0$ in HNM, and $\overline{h} = h'(h^{-1})$ in PNL.

This implies that the observational distribution alone is fully capable of distinguishing the functional forms of the causal mechanisms:

Theorem 4.4. *Under Assumption 4.3, given that Y is the effect variable, if $Y = f(\boldsymbol{X}, U)$ induces the observational distribution $p(\boldsymbol{X}, Y)$ and the family of the mechanism f is $\mathcal{F}$, then the posterior probability $P(\mathcal{F}|p(\boldsymbol{X}, Y)) = 1$.*

Proof. First, the spaces spanned by the QPEs for the four cases in Definition 4.2 are mutually non-overlapping due to the non-degeneracy assumption. Then:

- $\mathcal{F} = \mathcal{F}_{\text{LiNGAM}}$: For LiNGAM, its QPE is always a constant vector.
- $\mathcal{F} = \mathcal{F}_{\text{ANM}}$: For ANM with a non-linear g, ∇g will never degenerate into a constant vector. Therefore, the QPE space of ANM does not overlap with that of LiNGAM.
- $\mathcal{F} = \mathcal{F}_{\text{HNM}}$: For HNM with a non-constant $h : \mathbb{R}^d \to \mathbb{R}_+$, firstly, its QPE is guaranteed to exist (since $h > 0$ holds strictly, the QPE will never be ill-defined at any location due to $h = 0$). Secondly, $\frac{\nabla h}{h}$ must be a non-zero vector because h is non-constant. Thus, the basis function y in the QPE of HNM will not degenerate into a constant 1. This ensures that the QPE space of HNM does not overlap with those of ANM and LiNGAM.
- $\mathcal{F} = \mathcal{F}_{\text{PNL}}$: For PNL with a monotonic non-linear function h, the space where its QPE resides contains only the basis function $\overline{h} = h'(h^{-1})$. Since h is non-linear, $\overline{h}$ will certainly not degenerate to be independent of y. Hence, its QPE basis function will not degenerate into 1, ensuring that the

QPE space of PNL does not overlap with those of ANM and LiNGAM. Furthermore, because the basis function of PNL consists solely of $\overline{h}$, rather than the two basis functions present in HNM, the QPE space of PNL does not overlap with that of HNM either.

The above analysis demonstrates that, under the four cases in Definition 4.2, there is a mapping from the QPE $\boldsymbol{\psi}(y \mid \boldsymbol{x})$ to the mechanism family $\mathcal{F}$. Meanwhile, the observational distribution $p(\boldsymbol{X}, Y)$ uniquely determines a QPE $\boldsymbol{\psi}(y \mid \boldsymbol{x})$ via its density function (by the definition of QPE). Therefore, within the prior space described in Assumption 4.3, the observational distribution uniquely identifies a mechanism family; that is, $P(\mathcal{F} \mid p(\boldsymbol{X}, Y)) = 1$. □

Next, we analyze the identifiability assumptions concerning the causal direction, which necessitates the introduction of the Wronskian determinant. For a set of multivariate functions $f_1, \ldots, f_k$ where each $f_i : \mathbb{R}^d \to \mathbb{R}$, the determinant

$$W_X(f_1, \ldots, f_k) = \det \begin{bmatrix} f_1 & f_2 & \cdots & f_k \\ \partial_x f_1 & \partial_x f_2 & \cdots & \partial_x f_k \\ \vdots & \vdots & \ddots & \vdots \\ \partial_x^{k-1} f_1 & \partial_x^{k-1} f_2 & \cdots & \partial_x^{k-1} f_k \end{bmatrix}$$

is called the Wronskian determinant with respect to the covariate X.

Assumption A.2. For a given set of basis functions $\boldsymbol{b} = (b_1, \ldots, b_m)$, let $s_{i,j}(\boldsymbol{x}) = \partial_{x_i} \partial_{x_j} \log p(\boldsymbol{x})$ and $\boldsymbol{\eta}_i(\boldsymbol{x}) = \partial_{x_i} \big(\boldsymbol{b}(x_i)\, \partial_{x_i} \log p(\boldsymbol{x}) + \boldsymbol{b}'(x_i) \big)$. Assume that:

(i) The set of basis functions $\boldsymbol{b} = (b_1, \ldots, b_m)$ is specified;

(ii) For the causal mechanism $Y = f(\boldsymbol{X}, U)$, the QPE $\boldsymbol{\psi}(y \mid \boldsymbol{x})$ of the effect variable Y conditioned on its parent variables $\boldsymbol{X}$ lies in the linear span of the basis functions $\boldsymbol{b}$;

(iii) The QPE $\boldsymbol{\psi}(x_i \mid \boldsymbol{x}_{-i})$ of the variable X_i also lies in the linear span of the basis functions $\boldsymbol{b}$;

(iv) For any $X_j \in \boldsymbol{X}_{-i}, X_k \in \boldsymbol{X}_{-j}$, there exists some $\boldsymbol{x}$ such that $W_{X_j}(s_{j,k}(\boldsymbol{x}), \boldsymbol{\eta}_j(\boldsymbol{x})) \neq 0$.

Specifically, for any variable X_i, as long as its QPE $\boldsymbol{\psi}(x_i \mid \boldsymbol{x}_{-i})$ satisfies the finite linear span form (Equation 7) with respect to the given basis functions $\boldsymbol{b}$, it can be shown that $W_{X_i}(s_{i,j}, \boldsymbol{\eta}_i) \equiv 0$ for any $X_j \in \boldsymbol{X}_{-i}$ and any $\boldsymbol{x}$ (Chen et al., 2026, Theorem 3.6 and Corollary 3.8). Assumption A.2 couples this finite linear span form with the cause-and-effect relationship, and rules out coincidences where this necessary condition also identically holds for other variables $X_k \in \boldsymbol{X}_{-i}$.

Lemma A.3 (Chen et al. 2026, Corollary 3.8)**.** *If Assumption A.2 holds, then the variable X_i satisfying condition (iii) in Assumption A.2 is the unique effect variable.*

This lemma provides a pathway for a unified analysis of identifiability from observational distributions to causal directions and across different mechanism families. Although it can directly unify LiNGAM, ANM, and HNM, it cannot be directly applied to PNL. This is because, without any additional constraints, the basis functions in PNL remain dependent on the post-nonlinear functions, which violates the premise of specified basis functions (condition (i) in Assumption A.2). To address this, we introduce another assumption to restrict PNL:

Assumption A.4. For the post-nonlinear function h in PNL, assume that there is only a finite number of possible functions $\overline{h} = h'(h^{-1})$ within the prior space.

In practice, this assumption is highly likely to hold within the artificially constructed prior spaces for pre-training. Since the training only proceeds for a finite number of steps, only a finite number of post-nonlinear functions h are sampled from the PNL mechanisms. Consequently, the total number of $\overline{h}$ involved in this prior space can be considered finite[3]. Under Assumption A.4, we can incorporate PNL into the same framework as Lemma A.3, thereby proving the following:

Corollary A.5. *Assuming that condition (ii) in Assumption 4.3, condition (iv) in Assumption A.2, and Assumption A.4 hold simultaneously, the variable X_i satisfying condition (iii) in Assumption A.2 is the unique effect variable.*

[3]We acknowledge that Assumption A.4 is a compromise introduced to connect our setting to Lemma A.3. We believe that the identifiability result may still hold even when this assumption is violated; however, this case cannot currently be incorporated into the existing proof framework and is left for future work.

Proof. As shown in Table 4, the basis functions for LiNGAM, ANM, and HNM are completely determined (the basis $\boldsymbol{b}$ are 1, 1, and $(1, y)$, respectively). Furthermore, Assumption A.4 postulates that there are only finitely many possible $\overline{h}$, denoted by $(\overline{h}_1, \ldots, \overline{h}_k)$. We can then construct $\boldsymbol{b} = (1, y, \overline{h}_1, \ldots, \overline{h}_k)$ such that LiNGAM, ANM, HNM, and the PNL model satisfying Assumption A.4 all reside within the same finite-dimensional space spanned by the given basis $\boldsymbol{b}$. Consequently, conditions (i), (ii), and (iii) in Assumption A.2 are all satisfied. The proof is then completed by applying Lemma A.3. □

Based on the identifiability of the effect variables (i.e., the leaf nodes in a DAG), we can recursively prove the identifiability of the graph from the observational distribution:

Theorem 4.5. *Under Assumption 4.3 and some additional identifiability conditions, if the SCM $\phi = (\boldsymbol{X}, \boldsymbol{f}, p(\boldsymbol{U}))$ induces the observational distribution $p(\boldsymbol{X})$ and the causal graph $\mathcal{G}$, then the posterior probability is $P(\mathcal{G} \,|\, p(\boldsymbol{X})) = 1$, i.e., $\mathcal{G}$ is identifiable almost surely.*

Proof. Consider an SCM ϕ satisfying the assumptions, which induces the causal graph $\mathcal{G}$ and the observational distribution $p(\boldsymbol{X})$. By causal faithfulness and causal sufficiency, the skeleton of $\mathcal{G}$ and the directions of some edges, namely those in v-structures, are identifiable. This identifies a Markov equivalence class containing $\mathcal{G}$. We therefore focus on the directions of edges that remain undetermined within this Markov equivalence class.

Suppose that $\mathcal{G}' \neq \mathcal{G}$ is another graph in this equivalence class, induced by another SCM ϕ'. The difference between $\mathcal{G}'$ and $\mathcal{G}$ lies in the directions of certain edges. Consider each pair i, j such that (i, j) is an edge in $\mathcal{G}$ while (j, i) is an edge in $\mathcal{G}'$. Then, under ϕ, we have $X_i = f_i(\boldsymbol{X}_{\text{pa}(i)}, U_i)$, whereas under ϕ', we have $X_j = f_j(\boldsymbol{X}_{\text{pa}(j)}, U_j)$. Let N denote the set consisting of i, j, and all their neighbors in the skeleton. By adding null terms to f_i and f_j, we can write $X_i = f_i'(\boldsymbol{X}_{N \setminus i}, U_i)$ and $X_j = f_j'(\boldsymbol{X}_{N \setminus j}, U_j)$, where f_i' and f_j' clearly still satisfy the preceding assumptions.[4] These two equations simultaneously induce the observational distribution $p(\boldsymbol{X}_N)$.

However, under Corollary A.5, the preceding assumptions rule out the existence of two effect variables in $\boldsymbol{X}_N$. This yields a contradiction. Hence, the initial assumption is false, and $\mathcal{G}' = \mathcal{G}$. Therefore, $P(\mathcal{G} \,|\, p(\boldsymbol{X})) = 1$. □

[4] Although U_i and $\boldsymbol{X}_{N \setminus i}$ are not independent in this representation, this does not affect condition (ii) in Assumption 4.3 and condition (iv) in Assumption A.2, which is required in Corollary A.5.

B EXPERIMENTAL DETAILS

B.1 PRIOR SPACE CONSTRUCTION

Random Graph The experiments involve four types of random graphs:

- **Erdős–Rényi graph**: This random graph model assumes that edges between any pair of nodes are established independently with a fixed probability.
- **Barabási–Albert graph**: This model generates scale-free networks with power-law degree distributions by dynamically adding nodes and edges via a preferential attachment mechanism.
- **Watts–Strogatz graph**: This model constructs small-world networks with high clustering coefficients and short average path lengths by randomly rewiring edges of a regular ring lattice with a given probability.
- **Stochastic Block Model graph**: This model partitions network nodes into predefined community blocks and generates edges according to an independent probability matrix between and within blocks, widely used for modeling networks with prominent community structures.

During training, we only use Erdős–Rényi, Barabási–Albert, and Watts–Strogatz graphs, while the Stochastic Block Model appears exclusively in the OOD evaluation phase. The number of nodes during training is uniformly sampled from $[2, 100]$, corresponding to the number of features in the tabular dataset $\mathcal{D}$ during training. The expected graph density is sampled from $[0, 1]$, where we encourage greater coverage of sparse graphs by employing a truncated exponential distribution over $[0, 1]$ with its expected value set to 0.25.

Random Noise During training, the exogenous variables cover Normal, Laplace, Uniform, Student's T, LogNormal, Gumbel, Exponential, χ^2, Beta, and Gamma distributions. During OOD evaluation, Half-Normal and Weibull distributions are used instead. The hyperparameters for these distributions are randomly selected, configured as follows:

- **Uniform**: Uniform distribution over $[0, 1]$.
- **Normal**, **Laplace**, **LogNormal**, **Gumbel**: The location parameter μ and scale parameter σ are fixed to 0 and 1, respectively.
- **Exponential**: The rate parameter λ is uniformly sampled from $[0.5, 2.0]$.
- **Beta**, **Gamma**: All concentration parameters, including c_0 and c_1, are uniformly sampled from $[0.5, 5.0]$.
- **Student's T**, $\boldsymbol{\chi^2}$: The degrees of freedom parameter ν is uniformly sampled from $[2.1, 5.0]$.
- **Half-Normal**, **Weibull**: The scale parameter σ and concentration parameter c are fixed to 1.

We then sample 1D random noise variables i.i.d. from these randomly instantiated distributions. Subsequently, these noise samples are normalized and participate in the forward inference of the random causal mechanisms with controlled signal-to-noise ratios (SNR).

Random Mechanisms The building blocks of various causal mechanisms are randomly sampled functions $\mathbb{R}^d \to \mathbb{R}$. We employ the following random functions:

- **Random Linear**: The most basic affine transformation, defined as $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{W} + \boldsymbol{b}$. The weight matrix $\boldsymbol{W} \in \mathbb{R}^{d\times 1}$ and bias $\boldsymbol{b} \in \mathbb{R}$ are initialized using a uniform distribution and frozen afterwards. This transformation contains no additional hyperparameters to search.
- **MLP (Multilayer Perceptron)**: A fully connected neural network with L hidden layers, expressed as $f(\boldsymbol{x}) = \boldsymbol{W}_{L+1}\sigma(\dots\sigma(\boldsymbol{x}^\top \boldsymbol{W}_1 + \boldsymbol{b}_1)\dots) + \boldsymbol{b}_{L+1}$, where all network weights are frozen after initialization. This model involves two hyperparameters: first, the hidden layer configuration controlling depth and width (e.g., $[128, 128]$ denotes two hidden layers of equal width), sampled from combinations of 1, 2, or 3 layers with layer widths from $\{32, 64, 128, 256\}$; second, the activation function σ, discretely and uniformly sampled from the candidate set $\{\text{Tanh}, \text{Sigmoid}, \text{ReLU}, \text{GELU}, \text{SiLU}, \text{Softsign}\}$.
- **Periodic Functions**: A single-hidden-layer sinusoidal network based on random frequency mapping, defined as $f(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \sin(\boldsymbol{x}^\top \boldsymbol{w}_i + b_i)$, where feature weights $\boldsymbol{w}_i \sim \mathcal{N}(0, \gamma^2 \boldsymbol{I})$, biases $b_i \sim \mathcal{U}(0, 2\pi)$, and output weights $v_i \sim \mathcal{N}(0, 1)$. The output is scaled by $1/\sqrt{m}$ to maintain numerical variance stability. This function involves two hyperparameters: the number of

frequencies m, uniformly sampled from $[1, 200]$; and the frequency scaling factor γ controlling the width of the frequency distribution, continuously and uniformly sampled from $[0.1, 2.0]$.

- **GP-RFF (Random Fourier Feature Gaussian Process)**: Approximates a Gaussian Process with an RBF kernel using random Fourier features. It is defined as $f(\boldsymbol{x}) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} v_i \cos(\boldsymbol{x}^\top \boldsymbol{w}_i + b_i)$, where weights $\boldsymbol{w}_i \sim \mathcal{N}(0, l^{-2}\boldsymbol{I})$, biases $b_i \sim \mathcal{U}(0, 2\pi)$, and output weights $v_i \sim \mathcal{N}(0, 1)$. This model also contains two hyperparameters: the feature count m, uniformly sampled from $[10, 1024]$; and the lengthscale l of the RBF kernel, which dictates the variance of input projection weights ($1/l^2$) and is continuously sampled from $[0.1, 3.0]$.
- **Random Forest**: An ensemble model composed of T random decision stumps, defined as $f(\boldsymbol{x}) = \sum_{t=1}^{T} \left(v_t^{(R)} \mathbb{I}(\mathcal{C}_t(\boldsymbol{x})) + v_t^{(L)} (1 - \mathbb{I}(\mathcal{C}_t(\boldsymbol{x}))) \right)$, where $\mathbb{I}(\cdot)$ is the indicator function, and prediction values at left and right leaf nodes are initialized as $v_t^{(L)}, v_t^{(R)} \sim \mathcal{N}(0, \frac{1}{T})$. Its hyperparameters include the number of trees T sampled from $[1, 200]$, and the stump type determining the splitting condition $\mathcal{C}_t(\boldsymbol{x})$, which is chosen discretely between two types: axis-aligned, with splitting condition $\mathcal{C}_t(\boldsymbol{x}) := x_j > \tau$ ($j \sim \mathcal{U}\{1, d\}$, threshold $\tau \sim \mathcal{N}(0, 1)$); and oblique, with splitting condition $\mathcal{C}_t(\boldsymbol{x}) := \boldsymbol{x}^\top \boldsymbol{w} + b > 0$ ($\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{I})$, $b \sim \mathcal{N}(0, 1)$).

The random function type is uniformly sampled from the available candidate set associated with the specific causal mechanism. Next, the sampled random functions are used to construct the following six types of causal mechanisms:

- **LiNGAM**: Constructed via $Y = g(\boldsymbol{X}) + U$, where g is a random linear function, and U is guaranteed not to be sampled from Normal noise.
- **ANM**: Constructed via $Y = g(\boldsymbol{X}) + U$, where g is any random function (except the random linear function).
- **HNM**: Constructed via $Y = g(\boldsymbol{X}) + h(\boldsymbol{X})\, U$, where g and h are random functions, and h is constrained to be non-negative via an absolute value operation after sampling.
- **PNL**: Constructed via $Y = h(g(\boldsymbol{X}) + U)$, where g is a random function, and h is a 1D invertible distortion (see the "Post-processing" paragraph for details).
- **General**: Constructed via $Y = g(\boldsymbol{X}, U)$, where U is treated as an input dimension of g, and g is any random function except the random linear function to prevent degeneration into linear mechanisms.
- **PNL-HNM**: Constructed via $Y = h(a(\boldsymbol{X}) + b(\boldsymbol{X})\, U)$, where a and b are random functions, and h is a 1D invertible distortion.

The causal mechanism type is uniformly sampled from an available candidate set. For example, during training, the candidate set includes all mechanisms except PNL-HNM (detailed candidate sets for testing are provided in Section B.2). PNL-HNM is exclusively reserved for OOD evaluations.

Next, we inject noise with controlled signal-to-noise ratios (SNR), with the SNR (in dB) uniformly sampled from $[-5\text{ dB}, 10\text{ dB}]$. Specifically, for a causal mechanism $Y = f(\boldsymbol{X}, U)$, we compute the variance of $f(\boldsymbol{X}, 0)$ based on existing endogenous variables $\boldsymbol{X}$ and zero noise, and then scale the noise variance such that the ratio of noise variance to function variance matches the sampled SNR. This strategy ensures numerical stability during data sampling while encompassing both low- and high-noise regimes, diversifying the prior space to enhance model generalization. Finally, following (Ormaniec et al., 2025), we directly normalize the resulting Y after forward inference $Y = f(\boldsymbol{X}, U)$.

Post-processing Inspired by the tabular data processing approach in Hollmann et al. (2025), we randomly select a subset of features to undergo 1D warping and discretization. We consider the following 1D invertible warping functions, which are also used in PNL and PNL-HNM:

- **Sinh-Arcsinh**: A smooth nonlinear transformation constructed via hyperbolic sine and its inverse, defined as $f(x) = \sinh(\tau \operatorname{arcsinh}(x) + \epsilon)$. The shape of the warped input distribution is controlled by two hyperparameters: the skewness parameter ϵ, sampled continuously from $[-1.0, 1.0]$, which controls asymmetry; and the tail-weight parameter τ, sampled continuously from $[0.5, 2.0]$, which controls tail heaviness and decay rate.
- **Residual Flow**: A 1D invertible flow module based on a single-hidden-layer residual network, defined as $f(x) = c_{\text{safe}} x + \boldsymbol{w}^\top \tanh(x\boldsymbol{v} + \boldsymbol{b})$. To strictly guarantee monotonicity (i.e., every-

where differentiable with a strictly positive derivative to ensure invertibility), the linear term coefficient must be constrained. Since the minimum derivative contribution of the $\tanh$ layer is $\sum_{i=1}^{k} \min(w_i v_i, 0)$, a safety constant $c_{\text{safe}} = \alpha - \sum_{i=1}^{k} \min(w_i v_i, 0)$ is introduced, where α is the base slope. Hidden parameters $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{b} \in \mathbb{R}^k$ are randomly initialized from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and frozen. This function contains two hyperparameters: the number of hidden components k, discretely sampled from $\{1, 2, \ldots, 10\}$, and the base slope α, continuously sampled from $[0.1, 2.5]$.

- **Asymmetric Power**: An asymmetric power function constructed around a specified split point μ, enabling independent controllable nonlinear warping on both sides. Formally, $f(x) = \mu + s_r \max(x - \mu, 0)^{p_r} - s_l \max(\mu - x, 0)^{p_l}$. Under our setting, the left and right scaling factors s_l and s_r are fixed to 1.0, allowing the model to focus on controlling non-linearity via polynomial exponents. The remaining three hyperparameters are independently sampled: the split point $\mu \sim \mathcal{U}[-1.0, 1.0]$, and the exponents $p_l, p_r \sim \mathcal{U}[1.0, 2.5]$.

We also consider the following discretization functions:

- **Quantile**: Discretization based on the empirical cumulative distribution of the input data $\boldsymbol{x}$. By choosing equidistant quantiles in $(0, 1)$ as thresholds, the continuous values are partitioned into K discrete bins, ensuring an approximately equal sample count per bin. The hyperparameter, number of bins K, is uniformly sampled from the integer range $[2, 50]$.
- **Uniform**: Equal-width binning based on the absolute numerical range of the input data $\boldsymbol{x}$. By taking the minimum and maximum of the input tensor, the value range $[\min(\boldsymbol{x}), \max(\boldsymbol{x})]$ is linearly divided into K intervals of equal width. The bin count K is similarly sampled from $[2, 50]$.
- **Ordinal**: Order-based discretization. The high-dimensional tensor $\boldsymbol{x}$ is flattened into a 1D sequence and sorted in ascending order. Then, $K - 1$ absolute values at equal index intervals are extracted as cut-off thresholds. This method relies solely on relative rankings rather than continuous distributions. The number of bins K is uniformly sampled from $[2, 50]$.

In training and default evaluation configurations, the proportions of features undergoing warping and discretization each vary within $[0, 0.5]$ of total features, following a truncated exponential distribution with an expected value set to 0.25.

B.2 Benchmarks

Synthetic Benchmarks To evaluate performance under heterogeneous or homogeneous causal mechanisms, we construct the following synthetic benchmarks:

- **LiNGAM**: Causal mechanisms are restricted to LiNGAM, without warping or discretization.
- **ANM**: Causal mechanisms are restricted to ANM, without warping or discretization.
- **HNM**: Causal mechanisms are restricted to HNM, without warping or discretization.
- **PNL**: Causal mechanisms are restricted to PNL, without warping or discretization.
- **General**: Causal mechanisms are restricted to General, with default warping and discretization proportions.
- **Hetero**: Causal mechanisms follow the default configuration where all aforementioned mechanisms are selected with equal probability, with default warping and discretization proportions.

Each benchmark consists of 100 randomly sampled test cases, with sample size $n = 1000$ and feature dimension $d = 20$. Except for the specified mechanisms, all other configurations are identical to the prior space construction during training.

When evaluating specific properties, we vary only the target hyperparameter during sampling (e.g., varying only sample size n when evaluating sensitivity to sample size), while keeping all other settings identical to **Hetero**.

Real-World Benchmarks The real-world benchmarks used in our experiments are detailed below:

- **Sachs** (Sachs et al., 2005): We use the version of the Sachs dataset from DAG-GNN (Yu et al., 2019), which contains 11 nodes, 20 edges, and 7,466 samples.
- **Causal Chamber** (Gamella et al., 2025): We use data from the light tunnel chamber under standard configuration, containing 10,000 samples and 38 variables. We remove 18 constant variables and perform causal discovery on the remaining 20 non-constant variables.

In Appendix D, we additionally evaluate the methods on three real-world benchmarks. However, unlike the two benchmarks described above, their ground-truth causal graphs are not rigorously established through physical interventions. Therefore, we use them only as references for evaluating causal discovery performance in real-world settings:

- **Sockshop** (Ikram et al., 2022): A replica of an online sock-selling application consisting of 13 microservices implemented using different technologies. Each service runs in a separate virtual machine or container, communicates through HTTP API requests, and provides metrics such as CPU and memory utilization, latency, and error counts. The benchmark contains 50 root cause analysis test cases. We use the microservice dependency graph as the ground-truth causal graph.
- **Petshop** (Hardt et al., 2024): A dataset for evaluating root cause analysis in microservice-based applications. It contains latency, request, and availability metrics collected from a distributed application at five-minute intervals. The benchmark contains 50 root cause analysis test cases. We use the inverse call graph of the microservices as the ground-truth causal graph.
- **CIPCAD** (Menegozzo et al., 2022): A collection of datasets for fault detection and process control, including data generated by the Tennessee Eastman simulator and data collected from an ultra-processed food manufacturing plant. The latter includes two ground-truth structures, which we treat as two separate datasets. For each dataset, we use the corresponding process-control structure as the ground-truth causal graph.

B.3 Baselines

Appendix D presents the complete experimental results involving a total of 24 baseline methods:

(i) **Traditional search-based methods**: PC (Spirtes & Glymour, 1991), FCI (Spirtes, 1995), GES (Chickering, 2002);

(ii) **LiNGAM**: ICA-LiNGAM (Shimizu et al., 2006), Direct-LiNGAM (Shimizu et al., 2011);

(iii) **ANM**: CAM (Bühlmann et al., 2014), RESIT (Peters et al., 2014), RCD (Maeda & Shimizu, 2020), SCORE (Rolland et al., 2022), DAS (Montagna et al., 2023b), NoGAM (Montagna et al., 2023a), CaPS (Xu et al., 2024);

(iv) **HNM**: HOST (Duong & Nguyen, 2023), ICDH (Yin et al., 2024), SkewScore (Lin et al., 2025);

(v) **Optimization-based algorithms**: NOTEARS (Zheng et al., 2018), GOLEM (Ng et al., 2020), GraNDAG (Lachapelle et al., 2020), DAG-GNN (Yu et al., 2019), DAGMA (Bello et al., 2022);

(vi) **Amortized algorithms**: AVICI (Lorch et al., 2022), CauScale (Peng et al., 2026), TabCausal (Li et al., 2026), FoundCause (Blöbaum et al., 2026).

Among these methods, PC and FCI only output equivalence classes. For unoriented edges, following (Li et al., 2026), we orient them according to the original variable ordering in the benchmark dataset (rather than the ground-truth order). For NOTEARS and DAGMA, both linear and nonlinear variants are included. For amortized algorithms, we evaluate AVICI using its public `scm-v0` checkpoint, CauScale using its public `synthetic` checkpoint, and TabCausal and FoundCause using their respective sole public checkpoints.

B.4 Training Procedure

All models, including leaf prediction and parent prediction modules, are implemented using PyTorch and trained with a fixed random seed of 42 to ensure reproducibility. We optimize the model for a total of $500,000$ iterations using a global batch size of 64 (in practice, to prevent out-of-memory errors, gradient accumulation is applied with a mini-batch size of 1). The optimization process employs a peak learning rate of 1×10^{-4} along with a weight decay coefficient of 1×10^{-6} for regularization. To enhance numerical stability during early training, a linear learning rate warmup schedule is applied over the first $2,000$ steps. Additionally, gradient clipping with a maximum norm threshold of 2.0 is enforced to mitigate gradient explosion.

C Additional Experiments: Scaling, Robustness, and Generalization

This appendix presents additional experimental results evaluating the scaling capability, robustness under various data perturbations, and OOD generalization ability of DAG-FM in comparison with other amortized causal discovery methods. All experiments were conducted in an inference environment with 24GB VRAM; runs that encountered out-of-memory (OOM) or other engineering execution failures are omitted. Unless otherwise specified, the configurations for the synthetic benchmarks mirror the **Hetero** setting described in Section B.2. Furthermore, since baseline methods are incapable of outputting causal orderings, we report topological order identification performance exclusively for DAG-FM.

C.1 Scaling Capabilities

Scaling to Large Sample Sizes Figure 2 illustrates the performance of different amortized methods as the sample size increases (with the number of dimensions fixed at $d = 20$). Without specialized engineering optimizations, both AVICI and CauScale encounter OOM errors under large sample regimes. In contrast, DAG-FM achieves superior asymptotic performance and significantly outperforms TabCausal and FoundCause, demonstrating the structural advantages of the DAG-FM architecture. In the large-sample regime, DAG-FM has the second-shortest runtime, after TabCausal, and runs faster than FoundCause.

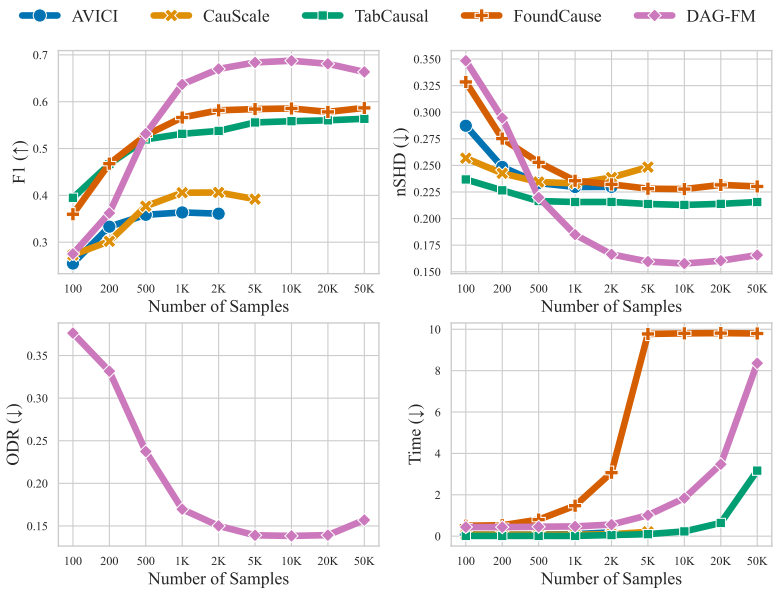


Figure 2: Performance of DAG identification (F1 and nSHD), order identification (ODR), and runtime (in seconds) under different numbers of samples.

Scaling to High Dimensions Figure 3 presents the performance of amortized models as the number of features increases (with the sample size fixed at $n = 1000$). AVICI and CauScale fail due to OOM errors in high-dimensional scenarios. Meanwhile, DAG-FM outperforms TabCausal and FoundCause at all evaluated dimensions except $d = 500$; at this extremely high dimensionality, its performance is comparable to that of TabCausal. In terms of runtime, DAG-FM is slower than TabCausal because it recursively iterates over the d variables.

C.2 Robustness

Robustness to Graph Density Real-world causal graphs vary considerably in density, encompassing both sparse and highly dense structures. Figure 4 displays the performance of amortized

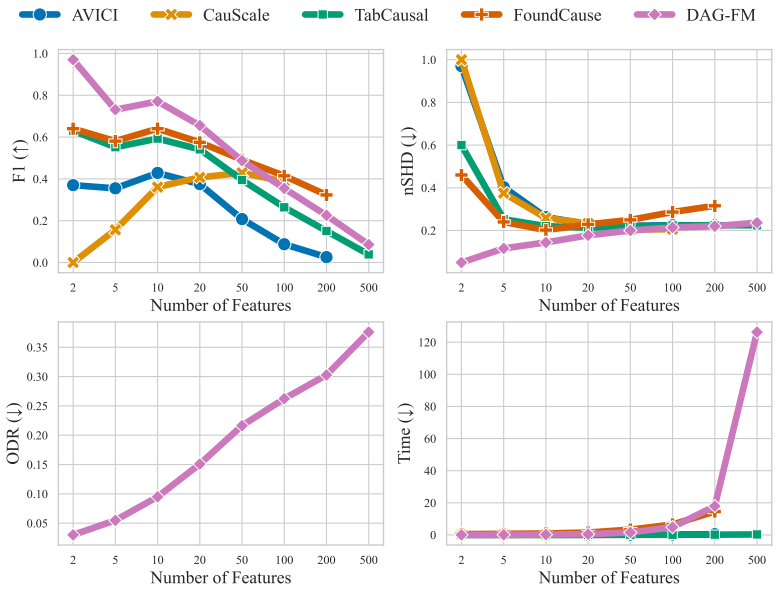


Figure 3: Performance of DAG identification (F1 and nSHD), order identification (ODR), and runtime (in seconds) under different numbers of features.

approaches as graph density increases ($\rho = 2|\mathcal{E}|/(d(d-1))$, where $|\mathcal{E}|$ denotes the number of edges in the graph). DAG-FM substantially outperforms competing methods on sparse graphs and ranks second only to FoundCause on full DAGs.

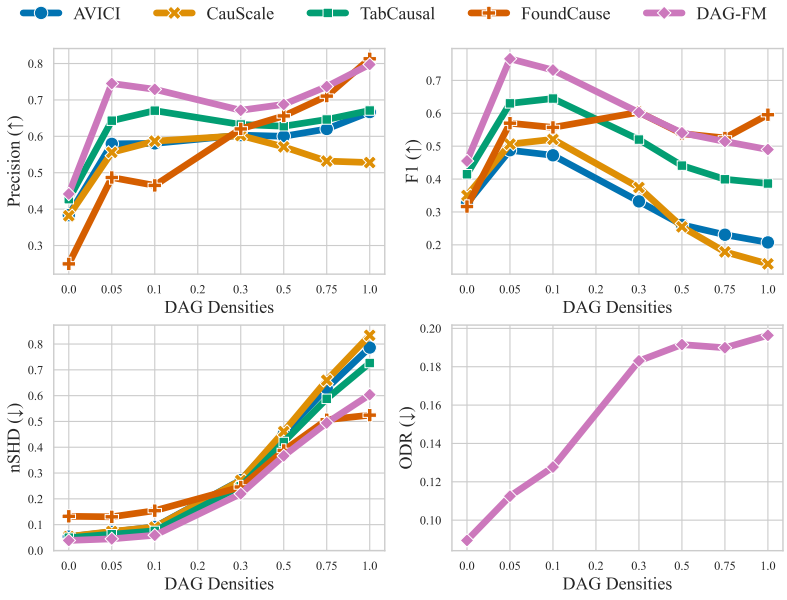


Figure 4: Performance of DAG identification (Precision, F1, nSHD) and order identification (ODR) under different densities.

Robustness to Warping Distortion Observational data collected in real-world environments often contains unknown distortions due to complex uncertainties, compromising compliance with underlying causal discovery assumptions. Figure 5 shows model performance as the proportion of distorted variables increases, where "proportion of warping" specifies the ratio of randomly warped

variables relative to the total number of variables. DAG-FM significantly outperforms all baseline methods, demonstrating robust resilience to warping distortions.

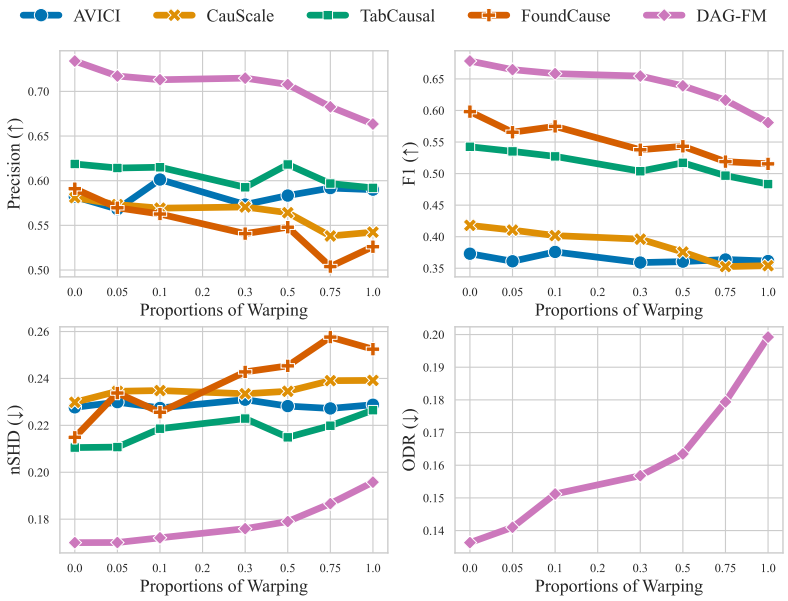


Figure 5: Performance of DAG identification (Precision, F1, nSHD) and order identification (ODR) under different proportions of warping.

Robustness to Discretization Another prevalent challenge in real-world datasets is mixed-type data, where features comprise both discrete and continuous variables. Figure 6 compares the performance of amortized methods as the fraction of discrete variables increases, where "proportion of discretization" indicates the ratio of discrete variables to all variables. DAG-FM outperforms competing methods across most settings, experiencing a slight performance drop only when virtually all variables are discrete, thereby confirming its robustness to discretization.

C.3 Generalization to Out-of-Distribution Data

Although amortized causal discovery algorithms enable zero-shot inference on unseen datasets, evaluating performance stability when test distributions deviate from the prior space remains crucial. Figure 7 compares the performance variations of amortized methods across four distinct OOD scenarios:

- **DAG**: The causal graphs of the test data are sampled from Stochastic Block Models, which are completely unseen during pre-training.
- **Mechanism**: The underlying causal mechanisms in the test SCMs follow PNL-HNM of the form $Y = h(f(\boldsymbol{X}) + g(\boldsymbol{X})U)$, where h is a non-linear invertible function. Although the theoretical identifiability of this functional form remains open, empirical evidence shows that the DAG structure can still be accurately identified.
- **Noise**: Exogenous noise variables in the test SCMs are sampled from Half-Normal and Weibull distributions, both of which are omitted during pre-training prior space construction.
- **Hetero**: PNL-HNM is mixed with other causal mechanisms within test datasets to systematically assess model capability under OOD heterogeneous mechanisms.

The evaluation results demonstrate that DAG-FM consistently outperforms baseline methods across all OOD scenarios, underscoring its strong generalization capabilities.

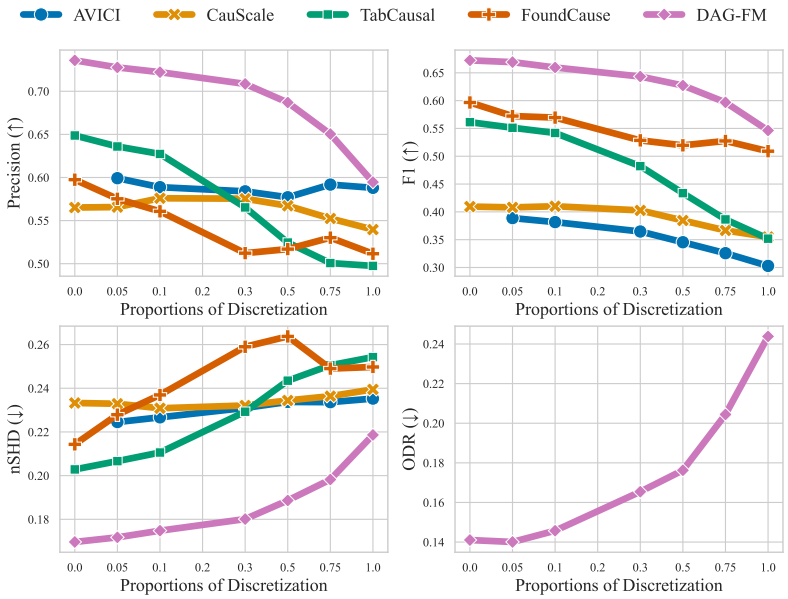


Figure 6: Performance of DAG identification (Precision, F1, nSHD) and order identification (ODR) under different proportions of discretization.

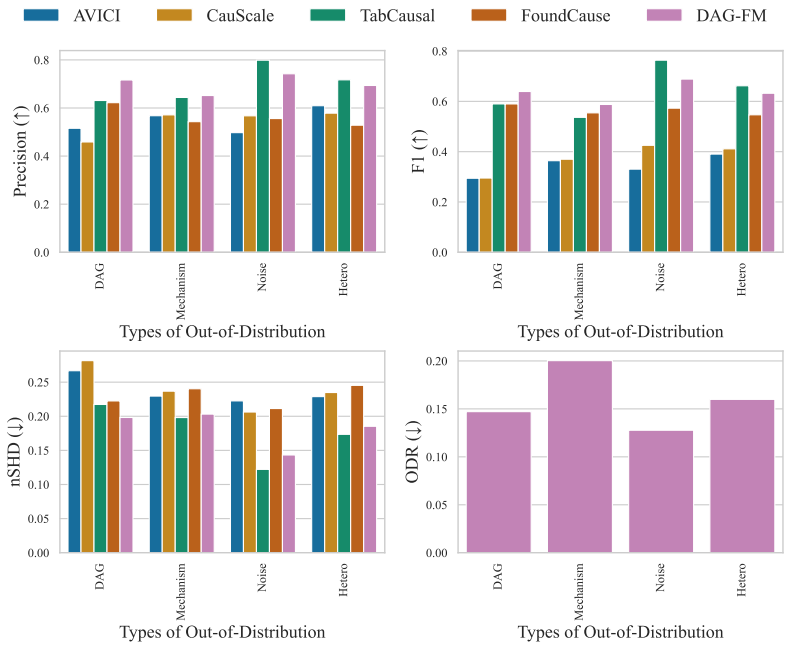


Figure 7: Performance of DAG identification (Precision, F1, nSHD) and order identification (ODR) under 4 types of OOD cases.

D ADDITIONAL EXPERIMENTS: ABLATION & FULL COMPARISON RESULTS

D.1 ABLATION STUDY ON MOLE

To demonstrate the performance gains brought by the Mixture of Causal Experts (MoLE) architecture in DAG-FM, we conduct an ablation study on the individual expert modules. Specifically, by modifying the routing logic, we isolate each individual expert by bypassing the router to exclude other experts, and compare its performance against the integrated MoLE model on both homogeneous and heterogeneous benchmarks. The results are summarized in Figure 8.

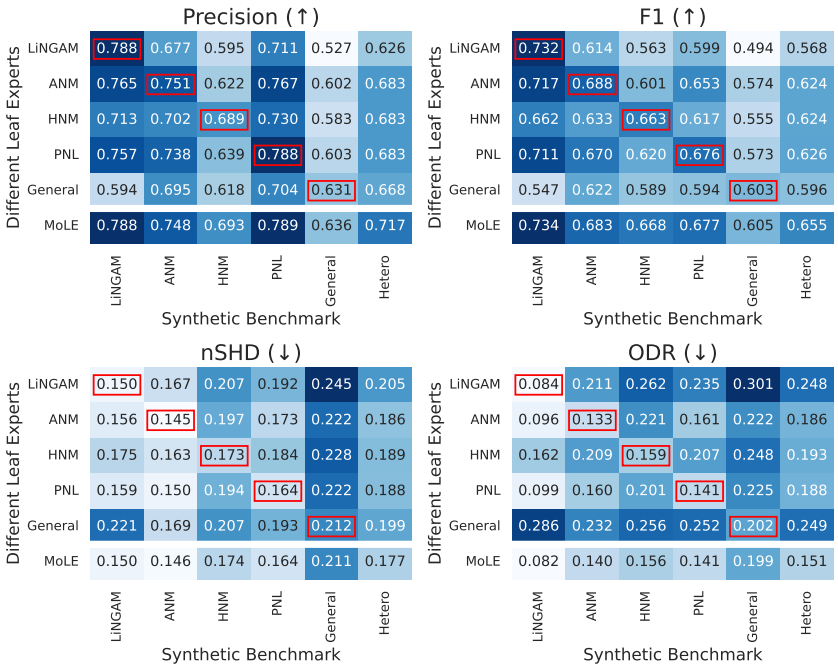


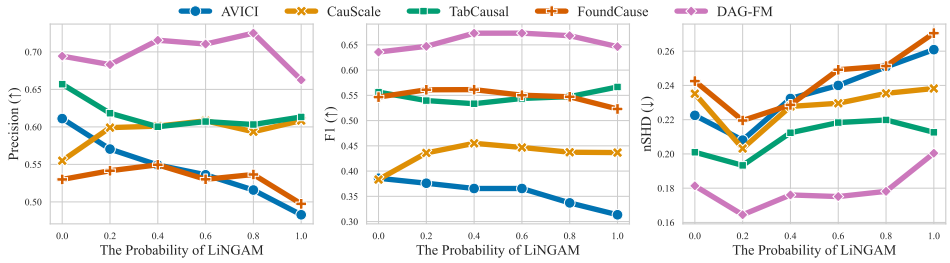
Figure 8: Performance of DAG identification (Precision, F1, nSHD) and order identification (ODR) across individual experts and the MoLE model on homogeneous and heterogeneous benchmarks.

The results yield three key insights: (i) In line with empirical expectations (highlighted along the diagonal red boxes), each domain-specific expert achieves the best performance among single-expert variants on its corresponding homogeneous mechanism benchmark; (ii) MoLE matches the performance of single experts under homogeneous mechanisms, confirming that the router successfully directs inputs to the appropriate experts; (iii) Under heterogeneous mechanism settings, MoLE enables DAG-FM to significantly outperform any individual expert trained on a single mechanism family.

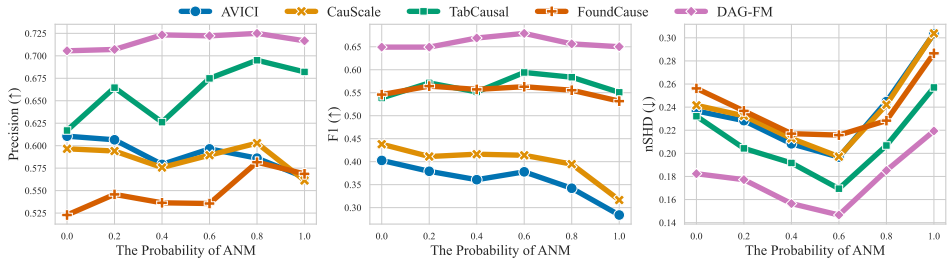
D.2 ABLATION ON OCCURRENCE PROBABILITIES OF HETEROGENEOUS MECHANISMS

In pre-training and the default **Hetero** synthetic benchmark, each mechanism family is sampled with equal probability. Here, we evaluate the performance of DAG-FM when mechanism families occur with non-uniform probabilities. As shown in Figure 9, from left to right, for a given target mechanism family $\mathcal{F}_i$, its occurrence probability $P(\mathcal{F}_i)$ increases from 0 to 1, while the remaining four mechanism families share the remaining probability equally at $(1 - P(\mathcal{F}_i))/4$.

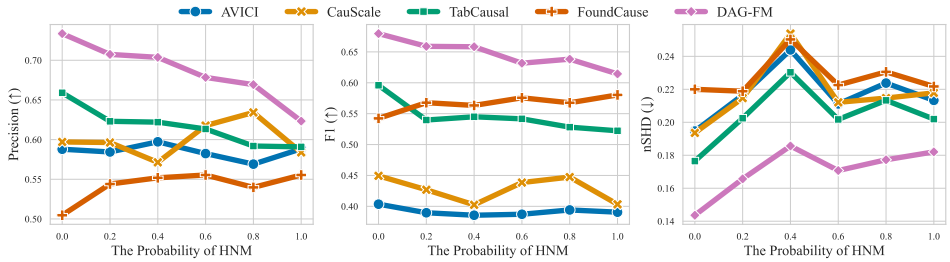
The results demonstrate that under provably identifiable settings (i.e., LiNGAM, ANM, HNM, and PNL), DAG-FM consistently outperforms baseline methods regardless of how the mechanism proportions vary, underscoring its robustness to non-uniform prior mechanism distributions. Performance degrades noticeably only under the General setting (where theoretical identifiability remains unproven), once the occurrence probability of General mechanisms exceeds 0.5.



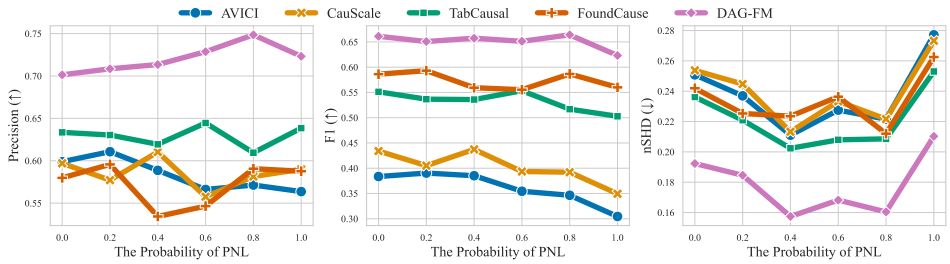
(a) LiNGAM



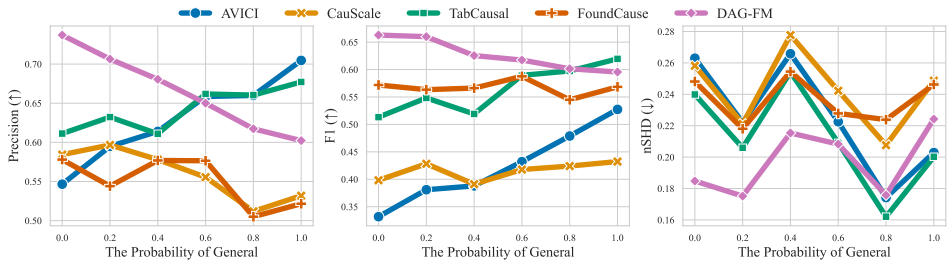
(b) ANM



(c) HNM



(d) PNL



(e) General

Figure 9: Performance of DAG identification (Precision, F1, nSHD) under non-uniform mechanism family occurrence probabilities: (a) LiNGAM, (b) ANM, (c) HNM, (d) PNL, and (e) General.

D.3 Scalability Limits under Reduced Precision

Table 5 presents the empirical computational limits of DAG-FM under a single 24GB VRAM environment during inference, including performance under BFloat16 mixed-precision.

Table 5: Maximum supported sample sizes (n) and feature dimensions (d) for DAG-FM inference under a single 24GB VRAM GPU footprint across different numerical precisions.

Precision	**Sample Size** (n)	**Feature Dimension** (d)
Float32	50,000	20
Float32	20,000	100
Float32	10,000	160
Float32	1,000	500
BFloat16	30,000	100
BFloat16	10,000	500
BFloat16	1,000	1,000

D.4 Full Comparison Results

Synthetic Benchmarks Tables 6 to 11 report the performance of all 24 baseline methods across the six synthetic benchmarks (**Hetero**, **LiNGAM**, **ANM**, **HNM**, **PNL** and **General**) described in Section B.2. DAG-FM consistently demonstrates a clear advantage across the vast majority of settings.

DAG Performance on Synthetic Heterogeneous Mechanism Benchmark ($n = 1000, d = 20$)

Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓
PC (Spirtes & Glymour, 1991)	$0.40_{0.13}$	$0.32_{0.14}$	$0.34_{0.11}$	$0.21_{0.08}$	$0.27_{0.21}$	$0.32_{0.24}$
FCI (Spirtes, 1995)	$0.40_{0.13}$	$0.32_{0.14}$	$0.33_{0.12}$	$0.21_{0.08}$	$0.27_{0.21}$	$0.57_{0.80}$
GES (Chickering, 2002)	$0.45_{0.15}$	$0.53_{0.19}$	$0.48_{0.15}$	$0.32_{0.13}$	$0.27_{0.21}$	$12.12_{7.17}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.28_{0.10}$	$0.36_{0.12}$	$0.31_{0.09}$	$0.19_{0.06}$	$0.36_{0.24}$	$0.37_{0.31}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.22_{0.10}$	$0.25_{0.13}$	$0.22_{0.10}$	$0.13_{0.06}$	$0.39_{0.29}$	$1.39_{0.04}$
CAM (Bühlmann et al., 2014)	$0.34_{0.13}$	$0.52_{0.15}$	$0.40_{0.12}$	$0.26_{0.09}$	$0.32_{0.21}$	$378.82_{26.83}$
RESIT (Peters et al., 2014)	$0.14_{0.08}$	$0.32_{0.12}$	$0.19_{0.09}$	$0.10_{0.05}$	$0.62_{0.34}$	$129.90_{7.84}$
RCD (Maeda & Shimizu, 2020)	$0.08_{0.12}$	$0.02_{0.03}$	$0.03_{0.05}$	$0.02_{0.02}$	$0.30_{0.23}$	$14.79_{11.21}$
SCORE (Rolland et al., 2022)	$0.38_{0.14}$	$0.53_{0.15}$	$0.43_{0.12}$	$0.28_{0.10}$	$0.29_{0.17}$	$13.44_{0.33}$
DAS (Montagna et al., 2023b)	$0.43_{0.16}$	$0.45_{0.15}$	$0.42_{0.13}$	$0.28_{0.10}$	$0.26_{0.18}$	$12.64_{0.64}$
NoGAM (Montagna et al., 2023a)	$0.39_{0.14}$	$0.55_{0.14}$	$0.45_{0.12}$	$0.30_{0.10}$	$0.28_{0.17}$	$46.40_{1.55}$
CaPS (Xu et al., 2024)	$0.38_{0.13}$	$0.54_{0.14}$	$0.43_{0.12}$	$0.28_{0.09}$	$0.29_{0.17}$	$14.76_{1.30}$
HOST (Duong & Nguyen, 2023)	$0.28_{0.13}$	$0.29_{0.12}$	$0.28_{0.11}$	$0.16_{0.08}$	$0.35_{0.26}$	$425.69_{180.88}$
ICDH (Yin et al., 2024)	$0.44_{0.16}$	$0.33_{0.16}$	$0.37_{0.14}$	$0.23_{0.11}$	$0.26_{0.21}$	$101.71_{67.40}$
SkewScore (Lin et al., 2025)	$0.28_{0.11}$	$0.42_{0.14}$	$0.33_{0.11}$	$0.20_{0.08}$	$0.39_{0.25}$	$14.75_{1.42}$
NOTEARS-Linear (Zheng et al., 2018)	$0.54_{0.20}$	$0.17_{0.12}$	$0.24_{0.14}$	$0.15_{0.10}$	$0.24_{0.21}$	$5.70_{4.36}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.44_{0.16}$	$0.30_{0.16}$	$0.35_{0.15}$	$0.22_{0.11}$	$0.26_{0.21}$	$13.35_{3.25}$
GOLEM (Ng et al., 2020)	$0.41_{0.18}$	$0.22_{0.13}$	$0.27_{0.14}$	$0.16_{0.10}$	$0.27_{0.22}$	$779.94_{423.28}$
GraNDAG (Lachapelle et al., 2020)	$0.39_{0.40}$	$0.05_{0.08}$	$0.08_{0.10}$	$0.04_{0.06}$	$0.26_{0.21}$	$242.65_{8.26}$
DAG-GNN (Yu et al., 2019)	$0.40_{0.20}$	$0.18_{0.14}$	$0.24_{0.16}$	$0.15_{0.12}$	$0.27_{0.22}$	$335.46_{36.78}$
DAGMA-Linear (Bello et al., 2022)	$0.43_{0.18}$	$0.19_{0.12}$	$0.25_{0.14}$	$0.15_{0.10}$	$0.26_{0.22}$	$2.22_{0.32}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.45_{0.15}$	$0.28_{0.13}$	$0.33_{0.13}$	$0.21_{0.10}$	$0.26_{0.20}$	$739.70_{250.37}$
AVICI (Lorch et al., 2022)	$0.58_{0.19}$	$0.30_{0.19}$	$0.37_{0.20}$	$0.25_{0.16}$	$0.23_{0.21}$	$0.06_{0.03}$
CauScale (Peng et al., 2026)	$0.57_{0.18}$	$0.34_{0.19}$	$0.41_{0.19}$	$0.27_{0.16}$	$0.23_{0.21}$	$0.07_{0.00}$
TabCausal (Li et al., 2026)	$\underline{0.63}_{0.18}$	$0.50_{0.19}$	$0.54_{0.17}$	$0.39_{0.17}$	$\underline{0.21}_{0.21}$	$0.01_{0.00}$
FoundCause (Blöbaum et al., 2026)	$0.57_{0.22}$	$\mathbf{0.66}_{0.18}$	$\underline{0.57}_{0.16}$	$\underline{0.42}_{0.16}$	$0.23_{0.17}$	$1.51_{0.01}$
DAG-FM (ours)	$\mathbf{0.72}_{0.15}$	$\underline{0.63}_{0.20}$	$\mathbf{0.66}_{0.16}$	$\mathbf{0.51}_{0.18}$	$\mathbf{0.18}_{0.17}$	$0.48_{0.01}$

Table 6: DAG identification performance on the synthetic **Hetero** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

Real-world Benchmarks Table 12 summarizes the DAG identification performance of all 24 baseline methods on two real-world benchmarks (**Sachs** and **Causal Chamber**). Table 13 reports the performance of all amortized algorithms on three additional real-world benchmarks (**Sockshop**, **Petshop**, and **CIPCAD**). DAG-FM outperforms the other methods on multiple metrics.

DAG Performance on Synthetic LiNGAM Benchmark ($n = 1000, d = 20$)					
Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓
PC (Spirtes & Glymour, 1991)	$0.45_{0.14}$	$0.34_{0.17}$	$0.37_{0.15}$	$0.23_{0.11}$	$0.26_{0.21}$
FCI (Spirtes, 1995)	$0.45_{0.15}$	$0.34_{0.17}$	$0.36_{0.15}$	$0.23_{0.11}$	$0.26_{0.21}$
GES (Chickering, 2002)	$0.53_{0.13}$	$\underline{0.71}_{0.21}$	$0.60_{0.14}$	$0.44_{0.14}$	$0.27_{0.24}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.12_{0.10}$	$0.14_{0.13}$	$0.13_{0.11}$	$0.07_{0.07}$	$0.43_{0.31}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.01_{0.02}$	$0.01_{0.02}$	$0.01_{0.02}$	$0.00_{0.01}$	$0.47_{0.38}$
CAM (Bühlmann et al., 2014)	$0.20_{0.12}$	$0.29_{0.15}$	$0.23_{0.13}$	$0.14_{0.08}$	$0.42_{0.25}$
RESIT (Peters et al., 2014)	$0.10_{0.08}$	$0.20_{0.15}$	$0.13_{0.10}$	$0.07_{0.06}$	$0.58_{0.40}$
RCD (Maeda & Shimizu, 2020)	$0.03_{0.04}$	$0.02_{0.03}$	$0.02_{0.03}$	$0.01_{0.02}$	$0.40_{0.29}$
SCORE (Rolland et al., 2022)	$0.38_{0.15}$	$0.51_{0.13}$	$0.43_{0.13}$	$0.28_{0.11}$	$0.30_{0.19}$
DAS (Montagna et al., 2023b)	$0.44_{0.17}$	$0.46_{0.14}$	$0.44_{0.14}$	$0.29_{0.12}$	$0.26_{0.19}$
NoGAM (Montagna et al., 2023a)	$0.40_{0.14}$	$0.54_{0.13}$	$0.45_{0.12}$	$0.30_{0.10}$	$0.29_{0.19}$
CaPS (Xu et al., 2024)	$0.39_{0.16}$	$0.52_{0.15}$	$0.43_{0.14}$	$0.29_{0.12}$	$0.30_{0.20}$
HOST (Duong & Nguyen, 2023)	$0.15_{0.10}$	$0.28_{0.12}$	$0.18_{0.10}$	$0.10_{0.06}$	$0.60_{0.32}$
ICDH (Yin et al., 2024)	$0.43_{0.19}$	$0.36_{0.17}$	$0.39_{0.17}$	$0.26_{0.14}$	$0.26_{0.21}$
SkewScore (Lin et al., 2025)	$0.23_{0.10}$	$0.40_{0.13}$	$0.28_{0.10}$	$0.17_{0.07}$	$0.44_{0.27}$
NOTEARS-Linear (Zheng et al., 2018)	$0.57_{0.19}$	$0.29_{0.18}$	$0.37_{0.19}$	$0.24_{0.16}$	$0.24_{0.22}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.42_{0.21}$	$0.30_{0.18}$	$0.34_{0.19}$	$0.22_{0.15}$	$0.26_{0.21}$
GOLEM (Ng et al., 2020)	$0.41_{0.17}$	$0.30_{0.15}$	$0.33_{0.14}$	$0.20_{0.11}$	$0.27_{0.20}$
GraNDAG (Lachapelle et al., 2020)	$0.23_{0.30}$	$0.03_{0.04}$	$0.04_{0.07}$	$0.02_{0.04}$	$0.26_{0.21}$
DAG-GNN (Yu et al., 2019)	$0.44_{0.20}$	$0.27_{0.15}$	$0.32_{0.16}$	$0.20_{0.13}$	$0.26_{0.21}$
DAGMA-Linear (Bello et al., 2022)	$0.52_{0.22}$	$0.33_{0.19}$	$0.39_{0.20}$	$0.27_{0.19}$	$0.25_{0.21}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.46_{0.14}$	$0.36_{0.16}$	$0.39_{0.14}$	$0.25_{0.13}$	$0.26_{0.21}$
AVICI (Lorch et al., 2022)	$0.53_{0.22}$	$0.29_{0.19}$	$0.35_{0.19}$	$0.23_{0.16}$	$0.23_{0.20}$
CauScale (Peng et al., 2026)	$0.62_{0.18}$	$0.40_{0.18}$	$0.47_{0.18}$	$0.33_{0.17}$	$0.22_{0.21}$
TabCausal (Li et al., 2026)	$\mathbf{0.82}_{0.18}$	$\mathbf{0.73}_{0.18}$	$\mathbf{0.77}_{0.17}$	$\mathbf{0.65}_{0.20}$	$\mathbf{0.12}_{0.14}$
FoundCause (Blöbaum et al., 2026)	$0.60_{0.21}$	$0.70_{0.23}$	$0.60_{0.17}$	$0.45_{0.19}$	$0.22_{0.17}$
DAG-FM (ours)	$\underline{0.79}_{0.15}$	$\underline{0.71}_{0.20}$	$\underline{0.73}_{0.16}$	$\underline{0.60}_{0.19}$	$\underline{0.15}_{0.16}$

Table 7: DAG identification performance on the synthetic **LiNGAM** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

DAG Performance on Synthetic ANM Benchmark ($n = 1000, d = 20$)					
Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓
PC (Spirtes & Glymour, 1991)	$0.39_{0.13}$	$0.33_{0.14}$	$0.34_{0.11}$	$0.21_{0.08}$	$0.25_{0.14}$
FCI (Spirtes, 1995)	$0.40_{0.13}$	$0.32_{0.15}$	$0.34_{0.11}$	$0.21_{0.08}$	$0.25_{0.14}$
GES (Chickering, 2002)	$0.45_{0.15}$	$0.48_{0.17}$	$0.44_{0.12}$	$0.29_{0.10}$	$0.25_{0.13}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.30_{0.10}$	$0.36_{0.13}$	$0.32_{0.09}$	$0.19_{0.07}$	$0.32_{0.18}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.24_{0.10}$	$0.27_{0.14}$	$0.25_{0.10}$	$0.14_{0.07}$	$0.34_{0.20}$
CAM (Bühlmann et al., 2014)	$0.46_{0.13}$	$0.62_{0.16}$	$0.51_{0.11}$	$0.35_{0.09}$	$0.23_{0.12}$
RESIT (Peters et al., 2014)	$0.10_{0.07}$	$0.13_{0.10}$	$0.11_{0.08}$	$0.06_{0.05}$	$0.43_{0.26}$
RCD (Maeda & Shimizu, 2020)	$0.05_{0.09}$	$0.01_{0.02}$	$0.02_{0.03}$	$0.01_{0.02}$	$0.27_{0.17}$
SCORE (Rolland et al., 2022)	$0.46_{0.13}$	$0.59_{0.16}$	$0.50_{0.12}$	$0.34_{0.10}$	$0.23_{0.13}$
DAS (Montagna et al., 2023b)	$0.50_{0.14}$	$0.49_{0.18}$	$0.48_{0.14}$	$0.33_{0.12}$	$0.22_{0.14}$
NoGAM (Montagna et al., 2023a)	$0.48_{0.13}$	$0.60_{0.16}$	$0.52_{0.12}$	$0.36_{0.11}$	$0.22_{0.12}$
CaPS (Xu et al., 2024)	$0.45_{0.13}$	$0.58_{0.16}$	$0.49_{0.12}$	$0.33_{0.10}$	$0.24_{0.13}$
HOST (Duong & Nguyen, 2023)	$0.29_{0.11}$	$0.27_{0.13}$	$0.27_{0.11}$	$0.16_{0.07}$	$0.30_{0.19}$
ICDH (Yin et al., 2024)	$0.51_{0.15}$	$0.35_{0.17}$	$0.40_{0.15}$	$0.26_{0.14}$	$0.22_{0.15}$
SkewScore (Lin et al., 2025)	$0.28_{0.13}$	$0.38_{0.15}$	$0.32_{0.12}$	$0.19_{0.09}$	$0.34_{0.17}$
NOTEARS-Linear (Zheng et al., 2018)	$0.61_{0.24}$	$0.16_{0.14}$	$0.23_{0.16}$	$0.14_{0.13}$	$0.22_{0.15}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.53_{0.15}$	$0.35_{0.19}$	$0.41_{0.17}$	$0.27_{0.15}$	$0.22_{0.16}$
GOLEM (Ng et al., 2020)	$0.47_{0.20}$	$0.20_{0.16}$	$0.26_{0.17}$	$0.16_{0.14}$	$0.23_{0.16}$
GraNDAG (Lachapelle et al., 2020)	$0.66_{0.34}$	$0.12_{0.15}$	$0.17_{0.16}$	$0.10_{0.12}$	$0.22_{0.16}$
DAG-GNN (Yu et al., 2019)	$0.44_{0.21}$	$0.16_{0.13}$	$0.23_{0.16}$	$0.14_{0.11}$	$0.23_{0.16}$
DAGMA-Linear (Bello et al., 2022)	$0.47_{0.23}$	$0.18_{0.14}$	$0.25_{0.17}$	$0.16_{0.13}$	$0.23_{0.16}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.48_{0.16}$	$0.25_{0.13}$	$0.31_{0.13}$	$0.19_{0.09}$	$0.23_{0.15}$
AVICI (Lorch et al., 2022)	$0.57_{0.19}$	$0.26_{0.18}$	$0.34_{0.19}$	$0.22_{0.16}$	$0.21_{0.16}$
CauScale (Peng et al., 2026)	$0.59_{0.21}$	$0.31_{0.19}$	$0.39_{0.20}$	$0.26_{0.17}$	$0.21_{0.16}$
TabCausal (Li et al., 2026)	$\underline{0.71}_{0.17}$	$0.57_{0.19}$	$\underline{0.62}_{0.17}$	$\underline{0.47}_{0.18}$	$\underline{0.17}_{0.14}$
FoundCause (Blöbaum et al., 2026)	$0.63_{0.19}$	$\mathbf{0.66}_{0.20}$	$0.61_{0.15}$	$0.46_{0.17}$	$0.19_{0.13}$
DAG-FM (ours)	$\mathbf{0.75}_{0.14}$	$\underline{0.65}_{0.18}$	$\mathbf{0.68}_{0.14}$	$\mathbf{0.54}_{0.16}$	$\mathbf{0.15}_{0.13}$

Table 8: DAG identification performance on the synthetic **ANM** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

DAG Performance on Synthetic HNM Benchmark ($n = 1000, d = 20$)					
Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓
PC (Spirtes & Glymour, 1991)	$0.39_{0.13}$	$0.33_{0.17}$	$0.32_{0.11}$	$0.20_{0.08}$	$0.27_{0.19}$
FCI (Spirtes, 1995)	$0.39_{0.13}$	$0.32_{0.17}$	$0.32_{0.11}$	$0.20_{0.08}$	$0.27_{0.19}$
GES (Chickering, 2002)	$0.43_{0.15}$	$0.49_{0.19}$	$0.43_{0.13}$	$0.29_{0.10}$	$0.27_{0.18}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.29_{0.13}$	$0.35_{0.15}$	$0.31_{0.13}$	$0.19_{0.11}$	$0.35_{0.22}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.23_{0.10}$	$0.26_{0.14}$	$0.23_{0.11}$	$0.14_{0.07}$	$0.37_{0.27}$
CAM (Bühlmann et al., 2014)	$0.29_{0.12}$	$0.48_{0.18}$	$0.35_{0.12}$	$0.22_{0.08}$	$0.37_{0.21}$
RESIT (Peters et al., 2014)	$0.14_{0.11}$	$0.43_{0.16}$	$0.19_{0.12}$	$0.11_{0.10}$	$0.80_{0.33}$
RCD (Maeda & Shimizu, 2020)	$0.05_{0.08}$	$0.01_{0.02}$	$0.02_{0.03}$	$0.01_{0.01}$	$0.29_{0.22}$
SCORE (Rolland et al., 2022)	$0.37_{0.14}$	$0.56_{0.16}$	$0.43_{0.12}$	$0.28_{0.10}$	$0.30_{0.16}$
DAS (Montagna et al., 2023b)	$0.41_{0.16}$	$0.46_{0.18}$	$0.42_{0.13}$	$0.27_{0.10}$	$0.27_{0.16}$
NoGAM (Montagna et al., 2023a)	$0.37_{0.14}$	$0.57_{0.15}$	$0.43_{0.12}$	$0.28_{0.09}$	$0.30_{0.15}$
CaPS (Xu et al., 2024)	$0.38_{0.14}$	$0.57_{0.16}$	$0.44_{0.12}$	$0.29_{0.10}$	$0.30_{0.16}$
HOST (Duong & Nguyen, 2023)	$0.33_{0.16}$	$0.31_{0.15}$	$0.31_{0.14}$	$0.19_{0.12}$	$0.34_{0.24}$
ICDH (Yin et al., 2024)	$0.47_{0.16}$	$0.33_{0.17}$	$0.38_{0.16}$	$0.25_{0.14}$	$0.25_{0.20}$
SkewScore (Lin et al., 2025)	$0.25_{0.12}$	$0.40_{0.17}$	$0.30_{0.12}$	$0.18_{0.08}$	$0.40_{0.23}$
NOTEARS-Linear (Zheng et al., 2018)	$0.55_{0.24}$	$0.16_{0.14}$	$0.24_{0.17}$	$0.15_{0.14}$	$0.24_{0.20}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.47_{0.18}$	$0.32_{0.20}$	$0.37_{0.19}$	$0.24_{0.16}$	$0.25_{0.21}$
GOLEM (Ng et al., 2020)	$0.44_{0.21}$	$0.21_{0.16}$	$0.27_{0.17}$	$0.17_{0.12}$	$0.26_{0.21}$
GraNDAG (Lachapelle et al., 2020)	$0.38_{0.35}$	$0.06_{0.12}$	$0.09_{0.11}$	$0.05_{0.07}$	$0.26_{0.20}$
DAG-GNN (Yu et al., 2019)	$0.45_{0.24}$	$0.17_{0.14}$	$0.24_{0.17}$	$0.15_{0.13}$	$0.26_{0.21}$
DAGMA-Linear (Bello et al., 2022)	$0.48_{0.23}$	$0.20_{0.17}$	$0.28_{0.19}$	$0.18_{0.16}$	$0.25_{0.21}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.46_{0.19}$	$0.26_{0.15}$	$0.32_{0.16}$	$0.20_{0.13}$	$0.26_{0.20}$
AVICI (Lorch et al., 2022)	$0.61_{0.20}$	$0.34_{0.18}$	$0.42_{0.19}$	$0.28_{0.16}$	$0.22_{0.19}$
CauScale (Peng et al., 2026)	$\underline{0.66}_{0.16}$	$0.40_{0.23}$	$0.47_{0.22}$	$0.34_{0.21}$	$0.22_{0.20}$
TabCausal (Li et al., 2026)	$0.64_{0.20}$	$0.54_{0.19}$	$0.57_{0.18}$	$0.42_{0.19}$	$\underline{0.19}_{0.16}$
FoundCause (Blöbaum et al., 2026)	$0.62_{0.21}$	$\mathbf{0.71}_{0.21}$	$\underline{0.62}_{0.16}$	$\underline{0.47}_{0.18}$	$0.21_{0.15}$
DAG-FM (ours)	$\mathbf{0.69}_{0.14}$	$\underline{0.67}_{0.18}$	$\mathbf{0.67}_{0.14}$	$\mathbf{0.52}_{0.17}$	$\mathbf{0.17}_{0.14}$

Table 9: DAG identification performance on the synthetic **HNM** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

DAG Performance on Synthetic PNL Benchmark ($n = 1000, d = 20$)					
Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓
PC (Spirtes & Glymour, 1991)	$0.39_{0.13}$	$0.30_{0.13}$	$0.32_{0.11}$	$0.20_{0.08}$	$0.28_{0.19}$
FCI (Spirtes, 1995)	$0.39_{0.13}$	$0.30_{0.13}$	$0.32_{0.11}$	$0.20_{0.08}$	$0.28_{0.19}$
GES (Chickering, 2002)	$0.45_{0.14}$	$0.50_{0.19}$	$0.45_{0.13}$	$0.30_{0.11}$	$0.27_{0.19}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.29_{0.09}$	$0.34_{0.11}$	$0.30_{0.08}$	$0.18_{0.06}$	$0.35_{0.23}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.23_{0.09}$	$0.25_{0.11}$	$0.23_{0.09}$	$0.14_{0.06}$	$0.37_{0.26}$
CAM (Bühlmann et al., 2014)	$0.35_{0.11}$	$0.51_{0.14}$	$0.40_{0.10}$	$0.26_{0.08}$	$0.33_{0.21}$
RESIT (Peters et al., 2014)	$0.14_{0.07}$	$0.29_{0.11}$	$0.18_{0.08}$	$0.10_{0.05}$	$0.61_{0.33}$
RCD (Maeda & Shimizu, 2020)	$0.09_{0.11}$	$0.02_{0.03}$	$0.04_{0.04}$	$0.02_{0.02}$	$0.30_{0.22}$
SCORE (Rolland et al., 2022)	$0.38_{0.12}$	$0.52_{0.14}$	$0.42_{0.11}$	$0.27_{0.09}$	$0.30_{0.18}$
DAS (Montagna et al., 2023b)	$0.42_{0.14}$	$0.44_{0.15}$	$0.41_{0.12}$	$0.27_{0.10}$	$0.28_{0.18}$
NoGAM (Montagna et al., 2023a)	$0.38_{0.12}$	$0.52_{0.15}$	$0.43_{0.11}$	$0.28_{0.09}$	$0.30_{0.18}$
CaPS (Xu et al., 2024)	$0.38_{0.12}$	$0.52_{0.14}$	$0.43_{0.11}$	$0.28_{0.09}$	$0.30_{0.18}$
HOST (Duong & Nguyen, 2023)	$0.37_{0.14}$	$0.33_{0.13}$	$0.33_{0.12}$	$0.21_{0.09}$	$0.31_{0.21}$
ICDH (Yin et al., 2024)	$0.46_{0.15}$	$0.32_{0.15}$	$0.36_{0.14}$	$0.23_{0.11}$	$0.26_{0.20}$
SkewScore (Lin et al., 2025)	$0.28_{0.12}$	$0.39_{0.12}$	$0.32_{0.11}$	$0.19_{0.08}$	$0.37_{0.20}$
NOTEARS-Linear (Zheng et al., 2018)	$0.59_{0.23}$	$0.16_{0.13}$	$0.24_{0.16}$	$0.15_{0.11}$	$0.24_{0.20}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.46_{0.16}$	$0.30_{0.17}$	$0.35_{0.16}$	$0.23_{0.12}$	$0.26_{0.20}$
GOLEM (Ng et al., 2020)	$0.47_{0.21}$	$0.21_{0.15}$	$0.28_{0.17}$	$0.17_{0.13}$	$0.26_{0.20}$
GraNDAG (Lachapelle et al., 2020)	$0.38_{0.35}$	$0.06_{0.08}$	$0.09_{0.12}$	$0.05_{0.07}$	$0.26_{0.21}$
DAG-GNN (Yu et al., 2019)	$0.47_{0.21}$	$0.18_{0.13}$	$0.25_{0.16}$	$0.15_{0.11}$	$0.26_{0.21}$
DAGMA-Linear (Bello et al., 2022)	$0.48_{0.21}$	$0.19_{0.14}$	$0.26_{0.17}$	$0.16_{0.13}$	$0.26_{0.21}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.46_{0.17}$	$0.25_{0.13}$	$0.31_{0.14}$	$0.19_{0.10}$	$0.26_{0.20}$
AVICI (Lorch et al., 2022)	$0.58_{0.18}$	$0.26_{0.15}$	$0.34_{0.16}$	$0.22_{0.13}$	$0.24_{0.19}$
CauScale (Peng et al., 2026)	$0.62_{0.16}$	$0.33_{0.18}$	$0.41_{0.19}$	$0.27_{0.15}$	$0.23_{0.20}$
TabCausal (Li et al., 2026)	$\underline{0.66}_{0.15}$	$0.50_{0.20}$	$0.55_{0.17}$	$0.40_{0.17}$	$\underline{0.21}_{0.19}$
FoundCause (Blöbaum et al., 2026)	$0.64_{0.21}$	$\mathbf{0.68}_{0.19}$	$\underline{0.62}_{0.15}$	$\underline{0.46}_{0.17}$	$\underline{0.21}_{0.16}$
DAG-FM (ours)	$\mathbf{0.79}_{0.11}$	$\underline{0.62}_{0.20}$	$\mathbf{0.68}_{0.15}$	$\mathbf{0.53}_{0.19}$	$\mathbf{0.16}_{0.16}$

Table 10: DAG identification performance on the synthetic **PNL** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

DAG Performance on Synthetic General Mechanism Benchmark ($n = 1000, d = 20$)					
Method	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓
PC (Spirtes & Glymour, 1991)	$0.35_{0.12}$	$0.28_{0.14}$	$0.29_{0.11}$	$0.18_{0.08}$	$0.30_{0.19}$
FCI (Spirtes, 1995)	$0.36_{0.12}$	$0.28_{0.14}$	$0.29_{0.11}$	$0.18_{0.08}$	$0.30_{0.19}$
GES (Chickering, 2002)	$0.37_{0.12}$	$0.51_{0.16}$	$0.41_{0.12}$	$0.27_{0.10}$	$0.34_{0.21}$
ICA-LiNGAM (Shimizu et al., 2006)	$0.25_{0.08}$	$0.39_{0.11}$	$0.30_{0.08}$	$0.18_{0.06}$	$0.46_{0.27}$
Direct-LiNGAM (Shimizu et al., 2011)	$0.26_{0.10}$	$0.39_{0.13}$	$0.30_{0.10}$	$0.18_{0.08}$	$0.43_{0.25}$
CAM (Bühlmann et al., 2014)	$0.42_{0.14}$	$\underline{0.68}_{0.12}$	$0.51_{0.13}$	$0.35_{0.11}$	$0.29_{0.16}$
RESIT (Peters et al., 2014)	$0.12_{0.07}$	$0.39_{0.13}$	$0.18_{0.08}$	$0.10_{0.05}$	$0.81_{0.27}$
RCD (Maeda & Shimizu, 2020)	$0.07_{0.12}$	$0.01_{0.01}$	$0.01_{0.02}$	$0.01_{0.01}$	$0.28_{0.20}$
SCORE (Rolland et al., 2022)	$0.35_{0.12}$	$0.59_{0.16}$	$0.44_{0.12}$	$0.29_{0.10}$	$0.36_{0.21}$
DAS (Montagna et al., 2023b)	$0.37_{0.12}$	$0.48_{0.17}$	$0.41_{0.11}$	$0.26_{0.09}$	$0.32_{0.20}$
NoGAM (Montagna et al., 2023a)	$0.37_{0.11}$	$0.62_{0.15}$	$0.45_{0.11}$	$0.30_{0.09}$	$0.34_{0.20}$
CaPS (Xu et al., 2024)	$0.35_{0.11}$	$0.60_{0.16}$	$0.44_{0.12}$	$0.29_{0.09}$	$0.35_{0.21}$
HOST (Duong & Nguyen, 2023)	$0.23_{0.11}$	$0.31_{0.13}$	$0.25_{0.10}$	$0.15_{0.07}$	$0.45_{0.27}$
ICDH (Yin et al., 2024)	$0.41_{0.13}$	$0.38_{0.14}$	$0.39_{0.12}$	$0.25_{0.11}$	$0.30_{0.21}$
SkewScore (Lin et al., 2025)	$0.26_{0.11}$	$0.45_{0.14}$	$0.32_{0.11}$	$0.20_{0.08}$	$0.44_{0.24}$
NOTEARS-Linear (Zheng et al., 2018)	$0.46_{0.17}$	$0.18_{0.13}$	$0.24_{0.14}$	$0.15_{0.12}$	$0.26_{0.20}$
NOTEARS-Nonlinear (Zheng et al., 2018)	$0.44_{0.14}$	$0.37_{0.17}$	$0.39_{0.15}$	$0.25_{0.13}$	$0.28_{0.20}$
GOLEM (Ng et al., 2020)	$0.32_{0.14}$	$0.23_{0.12}$	$0.26_{0.11}$	$0.15_{0.08}$	$0.32_{0.21}$
GraNDAG (Lachapelle et al., 2020)	$0.44_{0.41}$	$0.05_{0.08}$	$0.08_{0.12}$	$0.05_{0.08}$	$0.26_{0.20}$
DAG-GNN (Yu et al., 2019)	$0.35_{0.18}$	$0.21_{0.13}$	$0.25_{0.14}$	$0.15_{0.12}$	$0.30_{0.21}$
DAGMA-Linear (Bello et al., 2022)	$0.42_{0.18}$	$0.23_{0.14}$	$0.29_{0.15}$	$0.18_{0.13}$	$0.28_{0.21}$
DAGMA-Nonlinear (Bello et al., 2022)	$0.42_{0.12}$	$0.32_{0.12}$	$0.36_{0.10}$	$0.22_{0.08}$	$0.28_{0.20}$
AVICI (Lorch et al., 2022)	$\mathbf{0.69}_{0.14}$	$0.42_{0.18}$	$0.50_{0.16}$	$0.35_{0.17}$	$\mathbf{0.21}_{0.17}$
CauScale (Peng et al., 2026)	$0.51_{0.13}$	$0.38_{0.17}$	$0.42_{0.16}$	$0.28_{0.14}$	$0.26_{0.20}$
TabCausal (Li et al., 2026)	$\underline{0.65}_{0.13}$	$0.57_{0.17}$	$\underline{0.60}_{0.14}$	$\underline{0.44}_{0.15}$	$\mathbf{0.21}_{0.18}$
FoundCause (Blöbaum et al., 2026)	$0.51_{0.19}$	$\mathbf{0.69}_{0.17}$	$0.55_{0.13}$	$0.39_{0.13}$	$\underline{0.25}_{0.14}$
DAG-FM (ours)	$0.64_{0.11}$	$0.60_{0.17}$	$\mathbf{0.61}_{0.13}$	$\mathbf{0.45}_{0.14}$	$\mathbf{0.21}_{0.18}$

Table 11: DAG identification performance on the synthetic **General** benchmark (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines. Subscripts denote 95% confidence intervals.

DAG Performance on Real-world Benchmarks												
Method	**Sachs** ($n = 7466, d = 11$)						**Causal Chamber** ($n = 10\text{k}, d = 20$)					
	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓
PC (Spirtes & Glymour, 1991)	0.20	0.25	0.22	0.12	0.64	0.33	0.00	0.00	0.00	0.00	0.34	0.28
FCI (Spirtes, 1995)	0.21	0.25	0.23	0.13	0.62	0.83	0.00	0.00	0.00	0.00	0.32	0.45
GES (Chickering, 2002)	0.23	0.45	0.31	0.18	0.76	4.13	0.50	**0.62**	0.55	0.38	0.19	38.82
ICA-LiNGAM (Shimizu et al., 2006)	0.20	0.45	0.28	0.16	0.85	0.12	0.05	0.05	0.05	0.03	0.41	0.40
Direct-LiNGAM (Shimizu et al., 2011)	0.19	0.35	0.25	0.14	0.76	0.56	0.05	0.05	0.05	0.03	0.40	3.68
CAM (Bühlmann et al., 2014)	0.29	**0.75**	0.42	0.27	0.75	96.99			Out of Memory			
RESIT (Peters et al., 2014)	0.22	0.60	0.32	0.19	0.93	337.18			Out of Memory			
RCD (Maeda & Shimizu, 2020)	0.00	0.00	0.00	0.00	0.36	554.85			Out of Memory			
SCORE (Rolland et al., 2022)	0.20	0.45	0.28	0.16	0.85	222.86			Out of Memory			
DAS (Montagna et al., 2023b)	0.22	0.35	0.27	0.16	0.69	414.86			Out of Memory			
NoGAM (Montagna et al., 2023a)			Out of Memory						Out of Memory			
CaPS (Xu et al., 2024)	0.07	0.15	0.09	0.05	1.09	89.02	0.11	0.15	0.13	0.07	0.43	324.70
HOST (Duong & Nguyen, 2023)			Out of Memory						Out of Memory			
ICDH (Yin et al., 2024)	0.20	0.20	0.20	0.11	0.58	148.52	0.06	0.05	0.06	0.03	0.35	590.85
SkewScore (Lin et al., 2025)	0.23	0.50	0.31	0.19	0.80	95.78	0.07	0.13	0.09	0.05	0.53	362.79
NOTEARS-Linear (Zheng et al., 2018)	0.33	0.10	0.15	0.08	0.40	34.67	0.15	0.05	0.08	0.04	0.25	53.43
NOTEARS-Nonlinear (Zheng et al., 2018)	0.07	0.05	0.06	0.03	0.58	23.97	0.18	0.10	0.13	0.07	0.28	43.19
GOLEM (Ng et al., 2020)	0.10	0.10	0.10	0.05	0.65	364.46	0.07	0.05	0.06	0.03	0.34	377.49
GraNDAG (Lachapelle et al., 2020)	0.00	0.00	0.00	0.00	0.40	136.80	**1.00**	0.03	0.05	0.03	0.20	203.52
DAG-GNN (Yu et al., 2019)			Out of Memory						Out of Memory			
DAGMA-Linear (Bello et al., 2022)	0.20	0.10	0.13	0.07	0.47	2.22	0.04	0.03	0.03	0.02	0.32	3.22
DAGMA-Nonlinear (Bello et al., 2022)	0.25	0.20	0.22	0.12	0.51	230.47			Out of Memory			
AVICI (Lorch et al., 2022)			Out of Memory						Out of Memory			
CauScale (Peng et al., 2026)			Out of Memory						Out of Memory			
TabCausal (Li et al., 2026)	0.38	0.30	0.33	0.20	0.44	0.56	0.71	0.51	0.60	0.43	0.14	0.59
FoundCause (Blöbaum et al., 2026)	0.50	0.25	0.33	0.20	0.36	7.32	0.53	0.59	0.56	0.39	0.19	9.73
DAG-FM (ours)	**0.58**	0.35	**0.44**	**0.28**	**0.33**	1.19	0.80	0.51	**0.62**	**0.45**	**0.13**	1.81

Table 12: DAG identification performance on two real-world benchmarks (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with other baselines.

DAG Performance on Real-world Benchmarks																		
Method	**Sockshop** (50 Datasets)						**Petshop** (50 Datasets)						**CIPCAD** (3 Datasets)					
	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓	Prec. ↑	Rec. ↑	F1 ↑	Jac. ↑	nSHD ↓	Time ↓
AVICI (Lorch et al., 2022)	$0.05_{0.06}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.01_{0.01}$	$\underline{0.18}_{0.01}$	$6.51_{5.67}$	$\mathbf{0.26}_{0.18}$	$0.09_{0.06}$	$0.13_{0.09}$	$0.07_{0.05}$	$\mathbf{0.14}_{0.09}$	$3.19_{5.42}$	0.00	0.00	0.00	0.00	**0.08**	10.87
CauScale (Peng et al., 2026)	$\mathbf{0.10}_{0.04}$	$\underline{0.07}_{0.04}$	$\underline{0.08}_{0.04}$	$\underline{0.04}_{0.02}$	$0.26_{0.02}$	$0.14_{0.02}$	$0.15_{0.11}$	$0.23_{0.14}$	$\underline{0.18}_{0.12}$	$0.10_{0.07}$	$0.22_{0.11}$	$0.15_{0.04}$	0.10	$\underline{0.18}$	$\underline{0.13}$	$\underline{0.07}$	$\underline{0.17}$	0.19
TabCausal (Li et al., 2026)	$0.05_{0.10}$	$0.00_{0.01}$	$0.01_{0.01}$	$0.00_{0.01}$	$\mathbf{0.17}_{0.01}$	$0.02_{0.00}$	$0.17_{0.16}$	$0.19_{0.19}$	$\underline{0.18}_{0.17}$	$\underline{0.11}_{0.11}$	$\underline{0.15}_{0.08}$	$0.04_{0.04}$	$0.08_{0.03}$	$0.09_{0.03}$	$0.08_{0.01}$	$0.04_{0.01}$	$0.48_{0.39}$	$0.93_{0.76}$
FoundCause (Blöbaum et al., 2026)	$\underline{0.08}_{0.08}$	$0.02_{0.03}$	$0.02_{0.03}$	$0.01_{0.02}$	$0.19_{0.04}$	$1.50_{0.31}$	$0.10_{0.07}$	$\underline{0.35}_{0.22}$	$0.15_{0.10}$	$0.09_{0.05}$	$0.32_{0.06}$	$3.39_{2.52}$	$\underline{0.12}_{0.18}$	$0.14_{0.13}$	$0.11_{0.16}$	$\underline{0.07}_{0.09}$	$0.59_{0.21}$	$7.47_{1.10}$
DAG-FM (ours)	$\underline{0.08}_{0.03}$	$\mathbf{0.13}_{0.05}$	$\mathbf{0.10}_{0.04}$	$\mathbf{0.05}_{0.02}$	$0.37_{0.04}$	$0.78_{0.02}$	$\underline{0.24}_{0.07}$	$\mathbf{0.36}_{0.12}$	$\mathbf{0.28}_{0.09}$	$\mathbf{0.17}_{0.06}$	$0.22_{0.14}$	$0.65_{0.42}$	$\mathbf{0.18}_{0.25}$	$\mathbf{0.28}_{0.17}$	$\mathbf{0.19}_{0.23}$	$\mathbf{0.12}_{0.16}$	$0.60_{0.05}$	$1.87_{1.20}$

Table 13: DAG identification performance on 3 additional real-world benchmarks (higher is better for `Prec.`, `Rec.`, `F1`, and `Jac.`; lower is better for `nSHD`), compared with pretrained baselines.