FoundCause: Causal Discovery with Latent Confounders from Observational Data

Patrick Blöbaum[*,1], **Krishnakumar Balasubramanian**[*,1,2] **Shiva Prasad Kasiviswanathan**[1]
[1]Amazon Web Services
[2]Department of Statistics, University of California, Davis

Abstract

Causal discovery from observational data remains challenging due to the need to recover directed structure and latent confounding without interventions. We propose FoundCause, an *amortized causal discovery model* trained entirely on synthetic data that maps datasets directly to causal graphs in a single forward pass. By learning from large collections of simulated structural causal models, FoundCause captures transferable statistical patterns that generalize beyond individual datasets. The architecture incorporates several key inductive biases for causal discovery. It uses a permutation-invariant transformer encoder with alternating attention over samples and variables to jointly model cross-variable dependence and per-variable distributions. Pairwise statistical features derived from classical asymmetry measures are injected through statistics-conditioned attention, guiding the model toward known causal signals. A factorized decoder separates edge existence from direction, while a triangular refinement module enables reasoning over higher-order causal motifs such as chains and colliders. In addition, a dedicated confounder module based on learnable latent tokens explicitly models hidden common causes, and the model explicitly handles missing data via its masked input representation. To our knowledge, FoundCause is the first amortized causal discovery approach to explicitly model latent confounding. FoundCause outperforms 11 classical non-amortized methods (e.g., PC, GES, NOTEARS-style optimization) and 4 amortized causal discovery methods on 15 real-world datasets, achieving +9.6% improvement in F_1, +1.2% in AUROC, and an 18.9% reduction in structural Hamming distance relative to the strongest non-amortized methods, while performing inference in a single forward pass.

Model: https://github.com/amazon-science/foundcause

1 Introduction

Causal discovery from observational data is a central problem in machine learning, statistics, and many applied domains, requiring the recovery of directed relationships and latent confounding without access to controlled interventions. Given a dataset of i.i.d. samples, the goal is to infer the underlying causal graph governing the data-generating process, typically formalized as a structural causal model (SCM).

Classical approaches such as constraint-based methods (e.g., [51, 59]) and score-based methods (e.g., [10]) rely on conditional independence testing or combinatorial search, while more recent continuous optimization approaches (e.g., [61, 36]) formulate causal discovery as a differentiable problem. A complementary line of work leverages functional assumptions on the data-generating mechanisms to achieve identifiability (e.g. [50, 18, 60]). Although these methods are well studied, they typically require expensive per-dataset optimization, are sensitive to hyperparameters and distributional assumptions, and often struggle to scale to high-dimensional or heterogeneous data

Preprint.
* - Equal contribution.

regimes. As a result, their applicability in real-world settings remains limited. More fundamentally, causal discovery from observational data is not identifiable in general without additional assumptions. Yet these assumptions are often violated in real-world data, leading to a persistent gap between theoretical identifiability and practical applicability. This motivates inference procedures that combine signals across the full spectrum of identifiability regimes, rather than committing to any single one.

A recent line of work has reframed causal discovery as an *amortized inference* problem (e.g., [26, 20, 32, 55, 28]), where a model is trained on large collections of synthetic datasets and learns a direct mapping from data to graph structure. This paradigm enables single-pass inference and improved scalability, and offers a practical alternative to per-dataset optimization. However, existing neural approaches often rely on generic architectures with limited incorporation of domain-specific causal structure. In particular, they tend to treat causal discovery as a representation learning problem, without explicitly leveraging well-established statistical signals such as asymmetries in cause–effect relationships or higher-order structural patterns like chains and colliders. As a result, these methods can lack robustness and interpretability, especially when generalizing beyond the training distribution.

In this work, we propose `FoundCause`, a hybrid architecture that combines amortized inference with explicit statistical inductive biases. Rather than pursuing new identifiability guarantees, our goal is to learn practical inference procedures that perform well across a broad range of data-generating processes. `FoundCause` processes tabular data using a permutation-invariant transformer encoder with alternating attention over samples and variables, augmented by stat-conditioned attention that injects pairwise causal features derived from classical methods. The model predicts causal structure via a factorized decoder that separates edge existence from edge direction, and refines predictions using a triangular edge refinement module that enables reasoning over higher-order causal motifs. Additionally, `FoundCause` incorporates a dedicated confounder module based on learnable latent representations to model hidden common causes and produce a symmetric confounding matrix. Together, these components yield a unified framework that bridges classical causal discovery principles and modern deep learning, enabling scalable, single-pass inference of causal graphs with both directed and latent structure.

1.1 Related Works

Classical Causal Discovery. Classical methods recover graph structure via three paradigms. Constraint-based approaches (e.g., PC, FCI, RFCI, GFCI) use conditional independence tests and orientation rules to estimate Markov equivalence classes, with FCI variants handling latent confounding [51, 11, 38]. Score-based methods such as GES search over equivalence classes using decomposable scores [10, 35]. Continuous optimization approaches (e.g., NOTEARS, DAG-GNN, GraN-DAG, GOLEM, DAGMA) replace combinatorial search with smooth acyclicity constraints or differentiable DAG penalties [61, 57, 21, 36]. While well-founded, these methods require solving a separate optimization for each dataset and are sensitive to independence tests, scoring functions, modeling assumptions, and hyperparameters.

Extensions beyond Purely Observational DAGs. Several approaches address limitations of observational discovery. Intervention-aware methods (e.g., DCDI, ENCO) leverage interventional data to improve identifiability, with ENCO explicitly separating edge existence and orientation [7, 24]. Bayesian and generative approaches (e.g., DECI, DiBS, BayesDAG) learn distributions over graphs and mechanisms for uncertainty-aware inference, but require dataset-specific posterior inference [14, 25]. Bivariate methods (e.g., ANM, IGCI, PNL) exploit asymmetries such as residual independence, non-Gaussianity, or score structure to infer direction from pairs [33, 44]. Latent-variable methods model hidden confounding via mixed graphical representations (e.g., PAGs, MAGs, ADMGs) [5, 2, 27]. In contrast, our approach embeds asymmetry and confounding signals directly into an amortized architecture, rather than relying on standalone heuristics or separate inference procedures.

Identifiability Assumptions. What each method can recover depends on its identifiability assumptions. Under Markov and faithfulness, the DAG is identifiable only up to its Markov equivalence class [51]. Full-DAG identifiability requires additional structure, including linear non-Gaussian noise (LiNGAM) [49], nonlinear additive noise in the bivariate case (ANMs) [18], post-nonlinear models with invertible transformations [60], nonlinear additive models yielding a topological order (CAM) [8], equal or known error variances in linear Gaussian systems [40], or score-Jacobian asymmetries in additive-Gaussian DAGs [44]. With latent confounding or selection bias, identification weakens to PAGs representing equivalence classes of MAGs [59, 11, 38], while counterfactual identification

additionally requires monotonicity [9] or bijectivity [34]. These guarantees are asymptotic and apply to narrow, largely non-overlapping regimes that are difficult to verify in practice: faithfulness can fail in finite samples [53], and real-world mechanisms rarely satisfy strict linearity, additive-noise, or equal-variance assumptions [15]. Rather than committing to a single regime, `FoundCause` amortizes inference across a broad family of synthetic SCMs spanning many of these settings, learning empirical patterns indicative of causal direction.

Amortized Causal Discovery. Amortized causal discovery trains a model once on synthetic SCMs and performs inference in a single forward pass. AVICI uses alternating attention over variables and samples with a simple edge decoder [26]; CSIvA employs an encoder–decoder with variable-identity embeddings and autoregressive graph generation [20]; and hybrid methods such as SEA aggregate graphs from classical algorithms on variable subsets via a neural aggregator [55]. Amortization has also been applied beyond structure discovery, including ATE/CATE and intervention-effect estimation [37, 43, 3, 42]. `FoundCause` follows this paradigm but differs in four ways: (i) it injects pairwise causal statistics into attention and decoding, (ii) it factorizes edge existence and direction, (iii) it refines pairwise representations via triangular graph-level message passing, and (iv) it explicitly models latent confounding with dedicated tokens. These choices bridge classical heuristics and end-to-end neural architectures, preserving scalability while introducing inductive biases for dependence, directionality, higher-order structure, and hidden causes. Table 4 in Appendix A summarizes these distinctions.

2 Problem Setup

Given an observational dataset $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ consisting of N i.i.d. samples over D observed variables, the goal is to infer both directed causal relations and latent confounding among the observed variables. While classical and recent amortized approaches primarily focus on recovering directed structure over observed variables, explicitly modeling latent confounding remains comparatively underexplored in scalable settings. We represent directed causal structure by a directed acyclic graph (DAG) $\boldsymbol{D} \in \{0,1\}^{D \times D}$, where $D_{ij} = 1$ indicates that variable i is a direct cause of variable j. Latent confounding is represented by a symmetric matrix $\boldsymbol{C} \in \{0,1\}^{D \times D}$, where $C_{ij} = 1$ indicates that variables i and j share an unobserved common cause. The data-generating process is assumed to follow a structural causal model, $X_j = f_j(\mathrm{Pa}(j), \epsilon_j)\,,\ j = 1, \ldots, D$, where $\mathrm{Pa}(j)$ denotes the parents of node j in the true DAG, f_j is an arbitrary structural mechanism, potentially nonlinear and non-additive, and ϵ_j is exogenous noise. Some causes may be unobserved, thereby inducing statistical dependence among observed variables that is not explained by directed edges alone, motivating the need to jointly infer both directed structure and latent confounding.

Rather than running a separate optimization procedure for each dataset, e.g., as in PC [51], GES [10], or NOTEARS [61], we use *amortized causal discovery* following the AVICI framework [26]. The model is trained on thousands of synthetic datasets with known ground-truth structure and, at test time, performs causal discovery with a single forward pass. This approach posits (and verifies) that statistical patterns learned from diverse synthetic data distributions transfer to real-world datasets.

The model takes as input a data matrix $\boldsymbol{X}$ together with an optional binary observation mask $\boldsymbol{M} \in \{0,1\}^{N \times D}$, where $M_{ni} = 1$ denotes an observed entry and $M_{ni} = 0$ denotes a missing entry. During training, we sample problems with $D \in [2, 50]$ variables and $N \in [100, 600]$ observations; at inference time, the same architecture can be applied to larger numbers of variables and samples. The output consists of an edge-probability matrix $\hat{\boldsymbol{D}} \in [0,1]^{D \times D}$ and a symmetric confounding-probability matrix $\hat{\boldsymbol{C}} \in [0,1]^{D \times D}$. For directed edges, $\hat{D}_{ij} = \sigma(\ell_{ij})$ is the predicted probability of the causal relation $i \to j$, where ℓ_{ij} is the corresponding edge logit. For latent confounding, $\hat{C}_{ij}$ is the predicted probability that variables i and j share a hidden common cause.

3 Model Architecture and Loss Function

The architecture (see Figure 1) consists of four stages: (i) an axis-factorized encoder, (ii) attention-based pooling, (iii) a factored edge decoder with triangular refinement, and (iv) a learnable-token confounder module. The model has approximately 139M parameters; architectural hyperparameters are detailed in Appendix E.

Input Representation. Each entry (n, i) is represented as $\boldsymbol{x}_{ni} = [\bar{x}_{ni} m_{ni},\ m_{ni}] \in \mathbb{R}^2$, where $\bar{x}_{ni}$ is a normalized value and $m_{ni} \in \{0, 1\}$ indicates whether the entry is observed. A linear projection

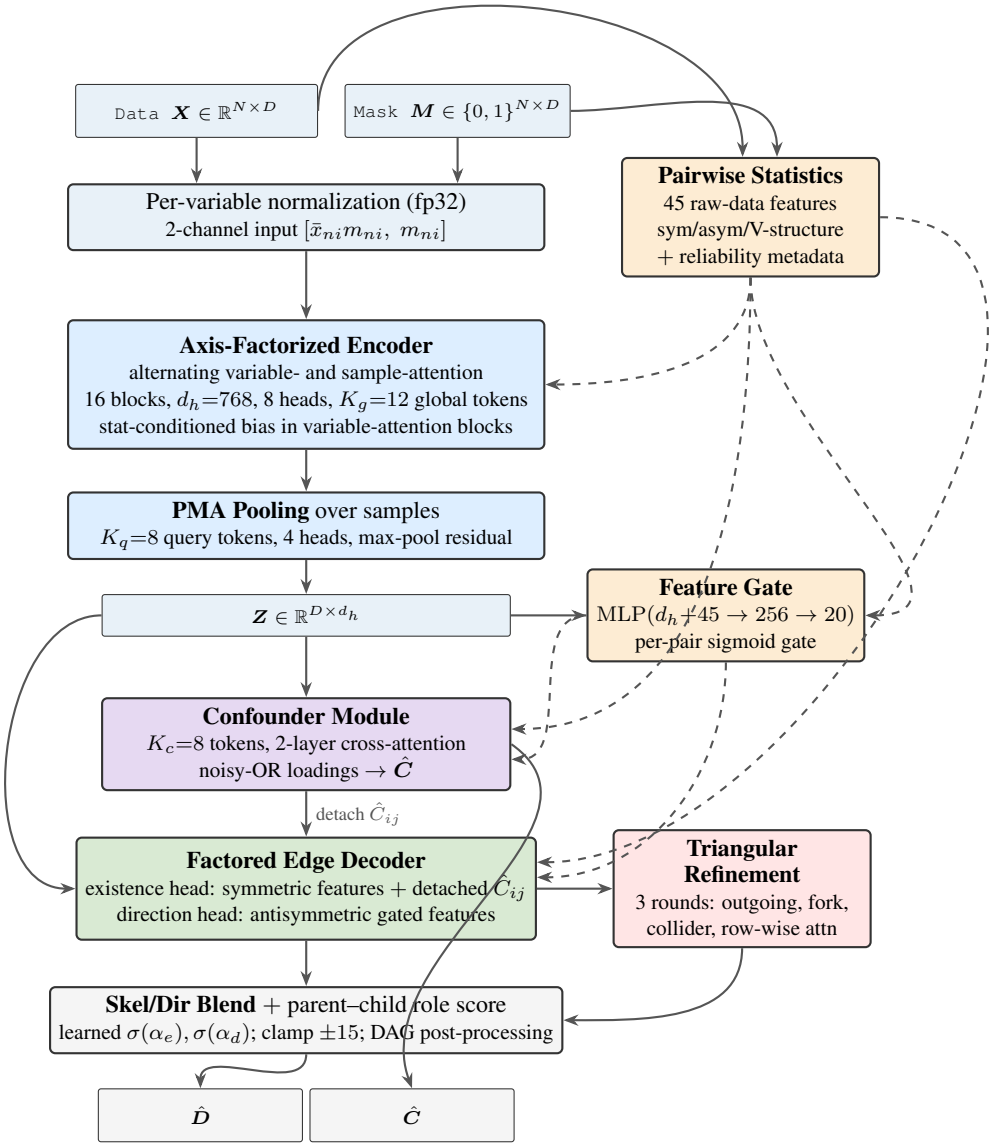

Figure 1: **Top-level Architecture.** The model takes normalized observations and pairwise statistics as input and processes them through four stages: an axis-factorized encoder, attention-based pooling (PMA) to obtain per-variable embeddings, a confounder module that predicts latent confounding via noisy-OR aggregation, and a factored edge decoder with triangular refinement to produce directed edge probabilities. Dashed lines indicate auxiliary inputs from pairwise statistics / feature gate.

produces $\boldsymbol{h}_{ni}^{(0)} \in \mathbb{R}^{d_h}$ via $\boldsymbol{h}_{ni}^{(0)} = \boldsymbol{W}_{\text{in}} \boldsymbol{x}_{ni} + \boldsymbol{b}_{\text{in}}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_h \times 2}$. No positional embeddings are used, making the model permutation-invariant over variables, which is essential since causal structure should not depend on variable ordering.

We now describe the key architectural details of these four stages mentioned above.

(i) Axis-factorized Encoder. Let $\boldsymbol{H} \in \mathbb{R}^{N \times D \times d_h}$ denote the stacked embeddings. We apply $2L$ attention blocks alternating between variable-wise attention (over D) and sample-wise attention (over N) (see Figure 3, Appendix E). This factorization reflects causal discovery, which relies both on cross-variable dependencies and distributional properties such as conditional independence.

Variable-attention incorporates: (i) $K_g \in \mathbb{N}$ learnable global tokens appended along the variable dimension, and (ii) a stat-conditioned bias $b_{ij}^{(h,\ell)} = \left[\boldsymbol{W}_2^{(\ell)}\ \text{GELU}\left(\boldsymbol{W}_1^{(\ell)} \boldsymbol{s}_{ij}\right)\right]_h$, where $\boldsymbol{s}_{ij} \in \mathbb{R}^{d_s}$

are pairwise statistics and GELU is the Gaussian Error Linear Unit [17]. This bias injects classical dependence signals (e.g., correlation and asymmetry), guiding the model toward known causal cues.

Each block uses RMSNorm [58], multi-head scaled dot-product attention (SDPA) [54], and a SwiGLU [48] feedforward network. The encoder outputs contextualized representations $\boldsymbol{H}^{\text{enc}} \in \mathbb{R}^{N\times D\times d_h}$.

(ii) PMA Pooling over Samples. For each variable $i \in \{1,\dots,D\}$, we aggregate $\{\boldsymbol{H}^{\text{enc}}_{ni}\}_{n=1}^{N}$ using Pooling by Multihead Attention (PMA) [23], producing $\boldsymbol{z}_i^{\text{PMA}} \in \mathbb{R}^{d_h}$. We also compute $\boldsymbol{z}_i^{\max} = \max_{n=1,\dots,N} \boldsymbol{H}^{\text{enc}}_{ni}$. These are combined via a learned gate $g \in \mathbb{R}$: $\boldsymbol{z}_i = \sigma(g)\boldsymbol{z}_i^{\text{PMA}} + (1-\sigma(g))\boldsymbol{z}_i^{\max}$, where $\sigma()$ denotes the sigmoid function. Stacking over variables yields $\boldsymbol{Z} \in \mathbb{R}^{D\times d_h}$. This pooling captures higher-order distributional properties (e.g., multimodality and tail behavior).

(iii) Factored Edge Decoder. For each ordered pair (i,j), we construct $\boldsymbol{f}_{ij}^{\text{exist}} = [\boldsymbol{z}_i + \boldsymbol{z}_j; \boldsymbol{z}_i \odot \boldsymbol{z}_j; \hat{C}_{ij}]$, and $\boldsymbol{f}_{ij}^{\text{dir}} = [\boldsymbol{z}_i - \boldsymbol{z}_j]$. This separates edge existence (symmetric dependence) from direction (asymmetric signals), mirroring classical causal discovery principles.

A shared MLP produces $\ell_{ij}^{\text{base}} \in \mathbb{R}$. We then add a role score $r_{ij} = (\boldsymbol{W}_V \boldsymbol{z}_i)^\top (\boldsymbol{W}_U \boldsymbol{z}_j)/\sqrt{d_{\text{pair}}}$,where $\boldsymbol{W}_V, \boldsymbol{W}_U \in \mathbb{R}^{d_{\text{pair}}\times d_h}$, yielding $\ell_{ij} = \ell_{ij}^{\text{base}} + \alpha r_{ij}$, $\alpha \in \mathbb{R}$. This provides an explicit inductive bias toward consistent parent–child roles. See also Figure 4, Appendix E.

Triangular Refinement. We construct pair representations $\boldsymbol{P} \in \mathbb{R}^{D\times D\times d_{\text{pair}}}$ and refine them for R rounds using triangle-based updates inspired by AlphaFold2 [19]. Each update aggregates over intermediate variables $k \in \{1,\dots,D\}$, enabling reasoning over triples (i,k,j). This captures causal motifs such as chains, forks, and colliders, which are helpful for distinguishing Markov-equivalent structures. Refined logits $\boldsymbol{\ell}^{\text{tri}} \in \mathbb{R}^{D\times D}$ are combined with $\boldsymbol{\ell}$, and final edge probabilities are $\hat{D}_{ij} = \sigma(\ell_{ij})$.

(iv) Confounder Module. Latent confounding is modeled using learnable matrix $\boldsymbol{T} \in \mathbb{R}^{K_c\times d_h}$, where each row $\boldsymbol{t}_k \in \mathbb{R}^{d_h}$ represents a candidate hidden common cause and K_c is the number of such confounders (see Figure 5, Appendix E). For each variable i and confounder k, we compute $S_{ik} = \sigma(\text{MLP}([\boldsymbol{z}_i; \boldsymbol{t}_k]))$, $S_{ik} \in (0,1)$, which denotes the probability that confounder k influences variable i. Pairwise confounding is computed via a noisy-OR aggregation: $\hat{C}_{ij} = 1 - \prod_{k=1}^{K_c}(1 - S_{ik}S_{jk})$, yielding a symmetric matrix $\hat{\boldsymbol{C}} \in [0,1]^{D\times D}$. This explicitly models hidden common causes, allowing the model to distinguish direct causal edges from correlations induced by latent confounding.

Final Loss Function. Let $\boldsymbol{\ell} \in \mathbb{R}^{D\times D}$ denote the directed edge logits, where ℓ_{ij} corresponds to the predicted logit for edge $i \to j$. Let $\boldsymbol{D}^* \in \{0,1\}^{D\times D}$ denote the ground-truth adjacency matrix of the DAG, and $\boldsymbol{C}^* \in \{0,1\}^{D\times D}$ the ground-truth confounding matrix. We define a valid-pair mask $V_{ij} = \mathbb{1}[i \neq j]\, m_i m_j$, which removes diagonal entries and padded variables, where $m_i \in \{0,1\}$ indicates whether variable i is present. Let $|V| = \sum_{ij} V_{ij}$, and define the number of positive and negative edges as $n_+ = \sum_{ij} D^*_{ij} V_{ij}$ and $n_- = \sum_{ij}(1 - D^*_{ij})V_{ij}$ respectively. The overall objective:

$$\mathcal{L} = \mathcal{L}_{\text{dir}} + \lambda_{\text{asym}}\mathcal{L}_{\text{asym}} + \lambda_{\text{cal}}\mathcal{L}_{\text{cal}} + \lambda_{\text{bdir}}\mathcal{L}_{\text{bdir}} + \lambda_{\text{skel}}\mathcal{L}_{\text{skel}} + \lambda_{\text{conf}}\mathcal{L}_{\text{conf}} + \mathcal{L}_{\text{fp}} + \lambda_{\text{den}}\mathcal{L}_{\text{den}} + \lambda_{\text{pma}}\mathcal{L}_{\text{pma}} + \lambda_{\text{gate}}\mathcal{L}_{\text{gate}} \quad (1)$$

which combines directed edge prediction, direction calibration, skeleton supervision, confounding prediction, and regularization. We first describe the directed edge loss, and then briefly summarize the remaining terms.

(a) Directed Edge Loss. The main term is a class-balanced binary cross-entropy over directed edges. We use direction-aware label smoothing

$$\tilde{D}_{ij} = \begin{cases} 1-\epsilon_d, & D^*_{ij} = 1, \\ \epsilon_d, & D^*_{ji} = 1, \\ \epsilon_0, & \text{otherwise}, \end{cases} \qquad \epsilon_d = 0.05, \quad \epsilon_0 = 0.005,$$

which introduces asymmetric supervision between (i,j) and (j,i), encouraging consistent edge orientation.

We define

$$\mathcal{L}_{\text{dir}} = \frac{1}{|V|} \sum_{ij} \Big(D^*_{ij} w_+ + (1 - D^*_{ij}) \Big) V_{ij} \, \text{BCE}(\ell_{ij}, \tilde{D}_{ij}),$$

where $\text{BCE}(\ell, y) = -y \log \sigma(\ell) - (1 - y) \log(1 - \sigma(\ell))$. To address class imbalance, the positive-class weight w_+ is adapted online based on recall: $\hat{r}_t = 0.95\hat{r}_{t-1} + 0.05 r_t$, and

$$w_+ = \text{clamp}\left(\sqrt{\frac{n_-}{n_+}} \Big[1 + 2 \, \text{clamp}(0.65 - \hat{r}_t, -0.3, 0.3) \Big], 1, 5 \right),$$

where r_t denotes the recall at iteration t. This stabilizes learning under severe edge sparsity while adapting the recall–precision tradeoff.

(b) Auxiliary Losses. The remaining terms address failure modes not captured by pairwise BCE. $\mathcal{L}_{\text{asym}}$ suppresses the reverse direction of true edges when its confidence exceeds a threshold, reinforcing causal asymmetry. $\mathcal{L}_{\text{cal}}$ limits extreme logit gaps between opposite directions, improving calibration and robustness under distribution shift. $\mathcal{L}_{\text{bdir}}$ applies a direction loss only to edges whose existence has been detected, decoupling edge discovery from orientation. $\mathcal{L}_{\text{skel}}$ provides supervision on the undirected skeleton, aligning with the fact that many causal signals are identifiable only up to equivalence classes. $\mathcal{L}_{\text{conf}}$ supervises the confounder module, enabling the model to distinguish direct edges from dependencies induced by latent common causes.

Finally, $\mathcal{L}_{\text{fp}}$ and $\mathcal{L}_{\text{den}}$ penalize overly dense graphs, while $\mathcal{L}_{\text{pma}}$ and $\mathcal{L}_{\text{gate}}$ regularize the pooling and gating mechanisms. Full definitions and hyperparameters are provided in Appendix F.

4 Training Pipeline

Optimization. We train using Schedule-Free AdamW [12] with base learning rate 4×10^{-4}, cosine decay to $\eta_{\min} = 10^{-5}$ over 300 epochs followed by a constant schedule, and momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$. Weight decay of 10^{-4} is applied to all weight matrices (excluding biases and normalization parameters), and gradients are clipped to a global norm of 1.0. Training runs for 1000 epochs with batch size 24. Schedule-Free AdamW maintains training parameters $\boldsymbol{z}$ and evaluation parameters $\boldsymbol{x}$, where $\boldsymbol{x}$ is a running average of $\boldsymbol{z}$; updates are applied to $\boldsymbol{z}$, while evaluation and checkpointing use $\boldsymbol{x}$.

Training Setup. We train using bfloat16 mixed precision with selective float32 computation for numerically sensitive operations (e.g., normalization, matrix inversions, and loss computation). Training uses distributed data parallelism across 8 A100 GPUs with synchronized gradients and data-parallel sharding; global statistics for adaptive loss weighting are synchronized across workers.

Data Regeneration and Training Strategy. Training data are continuously regenerated via parallel sampling of synthetic SCMs, exposing the model to a stream of diverse datasets rather than a fixed corpus. To prevent shortcut learning, we apply permutation augmentation—randomly permuting variable indices with corresponding relabeling of $\boldsymbol{D}^*$ and $\boldsymbol{C}^*$—following [41], removing ordering artifacts and encouraging reliance on distributional and relational signals. We further employ hard mining by replaying high-loss examples from recent epochs, maintaining focus on challenging cases.

5 Experimental Results

Training data are generated entirely from synthetic structural causal models (SCMs) using DoWhy [47, 6], by sampling diverse graphs and mechanisms and drawing observational datasets with full supervision for directed edges and latent confounding. The model is trained exclusively on synthetic data, without using real-world datasets. Additional details are provided in Appendix G. `FoundCause` performs inference in under 2 seconds on average across all datasets, whereas classical methods often require per-dataset optimization or combinatorial search and can take hours on moderate-sized graphs [51].

5.1 Real-World Benchmark Evaluation

Datasets. We evaluate `FoundCause` on 15 real-world and semi-realistic benchmarks spanning Bayesian networks, nonlinear variants, biological systems, and dense high-dimensional graphs. The suite includes ASIA, CHILD, INSURANCE, SACHS, CAUSAL_CHAMBERS_LT, ECOLI_LIKE, and four

Method	Cov.	Avg. AUROC ↑	Avg. F_1 ↑	Avg. Skel-F_1 ↑	Avg. SHD$/d$ ↓
Classical baselines					
PC (α=0.01) [51]	13/15	0.710	0.464	0.642	1.383
PC (α=0.05) [51]	13/15	0.692	0.430	0.646	1.495
FCI (α=0.05) [51]	13/15	0.637	0.380	0.483	1.209
GES [10]	15/15	0.725	0.456	0.631	1.596
GRaSP [22]	15/15	0.747	0.488	**0.676**	1.454
DirectLiNGAM [50]	15/15	0.675	0.314	0.510	2.681
ICA-LiNGAM [49]	15/15	0.653	0.292	0.500	3.008
NOTEARS-Linear [61]	15/15	0.588	0.251	0.509	1.470
DAGMA (linear) [4]	15/15	0.597	0.267	0.540	1.546
SCORE [44]	15/15	0.554	0.143	0.366	4.538
Naive correlation	15/15	0.584	0.190	0.427	4.653
Amortized / foundation-model methods					
AVICI [26]	15/15	0.732	0.498	0.651	1.142
CSIvA [20]	15/15	0.739	0.507	0.658	1.108
SEA [55]	15/15	0.744	0.512	0.667	1.052
Cond_FIP [28]	15/15	0.728	0.489	0.646	1.176
`FoundCause` (ours)	15/15	**0.756**	**0.535**	0.663	**0.980**

Table 1: Average performance of various causal discovery techniques across 15 real-world benchmark datasets. All metrics are macro-averaged across datasets. AUROC and F_1 are higher-is-better, while normalized structural Hamming distance (SHD$/d$) is lower-is-better; Skel-F_1 measures undirected edge recovery (higher is better). *Cov.* denotes the number of datasets on which a method successfully completed (some classical methods fail or time out on dense, near-collinear PETSHOP graphs at $d = 41$). `FoundCause` is the only method that achieves the top rank on F_1, SHD$/d$, and AUROC while successfully running on all $15/15$ datasets. Representative per-dataset results are in Appendix H.

PETSHOP variants, with graph sizes up to $d = 41$. We also evaluate on the Tübingen cause–effect pairs benchmark [33], a collection of real-world bivariate tasks where the goal is to infer whether $X \to Y$ or $Y \to X$. This setting is challenging because conditional-independence tests and multivariable constraints are unavailable, requiring reliance on asymmetric distributional signals such as nonlinearity, non-Gaussianity, and noise–mechanism independence. Additional dataset details are provided in Appendix C.

Compared Approaches. While many causal discovery methods have been proposed, our goal is not exhaustive coverage but a comparison against a representative set of strong baselines commonly used in practice. Accordingly, we compare against classical approaches spanning constraint-based methods (e.g., PC [51]), score-based search (e.g., GES [10]), LiNGAM variants [50], continuous DAG optimization methods (e.g., NOTEARS [61]), along with additional baselines such as SCORE [44] and naive correlation. We also include amortized or foundation-style baselines, including AVICI [26], CSIvA [20], SEA [55], and Cond_FIP [28]. Constraint-based methods (PC, FCI, GES, GRaSP) rely on conditional-independence tests and cannot orient bivariate pairs without auxiliary assumptions, so they are excluded from the comparison on the Tübingen cause–effect pairs benchmark. We also compare to SLOPPY [31] and HECI [56], two state-of-the-art methods for bivariate settings.

Results. Our inference protocol is described in Appendix D. Table 1 reports AUROC, directed-edge F_1, skeleton F_1, normalized structural Hamming distance (SHD$/d$), and coverage. AUROC measures edge-ranking quality, directed F_1 evaluates recovery of oriented causal edges, skeleton F_1 measures adjacency recovery ignoring direction, and SHD$/d$ captures normalized graph-edit error (lower is better); see Appendix B for precise definitions. Coverage indicates whether a method completes on each dataset, which is important as some classical methods fail or time out on dense, near-collinear PETSHOP graphs (e.g., PC and FCI complete $13/15$ datasets), whereas all amortized methods and `FoundCause` complete all 15. `FoundCause` is evaluated from a single fixed checkpoint without retraining or per-dataset tuning, using permutation averaging, temperature calibration, adaptive thresholding, and self-consistency pruning.

Across the datasets, FoundCause achieves the best average AUROC, directed F_1, and SHD/d while maintaining full $15/15$ coverage. It attains AUROC 0.756, outperforming the strongest classical baseline (GRaSP, 0.747) and amortized baseline (SEA, 0.744). More importantly, it improves directed-edge recovery with $F_1 = 0.535$ versus 0.512 (SEA) and 0.488 (GRaSP), and reduces structural error to $\mathrm{SHD}/d = 0.980$ versus 1.052 (SEA) and 1.454 (GRaSP). While GRaSP achieves the highest skeleton F_1 (0.676), FoundCause remains close (0.663) while substantially outperforming it on direction-sensitive metrics, indicating stronger recovery of directed causal structure.

Method	**Correct / Total**	**Unweighted ↑**	**Weighted ↑**
NOTEARS (Linear) [61]	$26/102$	25.5%	20.1%
DAGMA (linear) [4]	$34/102$	33.3%	34.2%
Naive correlation	$42/102$	41.2%	44.1%
DirectLiNGAM [50]	$45/102$	44.1%	46.5%
ICA-LiNGAM [49]	$52/102$	51.0%	49.4%
AVICI [26]	$55/102$	53.9%	55.8%
CSIvA [20]	$57/102$	55.9%	57.4%
SEA [55]	$59/102$	57.8%	58.6%
Cond_FIP [28]	$54/102$	52.9%	54.1%
HECI [56]	$72/102$	70.4%	71.8%
SLOPPY [31]	$\mathbf{74/102}$	$\mathbf{72.5}\%$	$\mathbf{73.4}\%$
FoundCause (ours)	$65/102$	63.7%	65.4%

Table 2: Results on the Tübingen cause–effect pairs benchmark. We evaluate on 102 bivariate pairs with known causal direction and positive evaluation weights. Unweighted accuracy is the fraction of pairs with correctly predicted direction, while weighted accuracy accounts for pair-specific difficulty and relevance as defined by the benchmark. FoundCause is the only amortized method that approaches the performance of SLOPPY and HECI, which are specifically designed for the bivariate setting.

On the Tübingen cause–effect pairs benchmark (Table 2), FoundCause outperforms the strongest classical baseline (ICA-LiNGAM) by at least 12.7 percentage points in accuracy and is the only amortized method approaching the performance of SLOPPY and HECI, which are specialized for the bivariate setting.

Test-time ablations. We perform test-time ablations by replacing intermediate activations with neutral values at inference, without retraining, isolating each component's contribution. Pairwise statistics and the dedicated direction pathway are the primary drivers of performance, with large drops when removed, while triangular refinement provides additional gains via higher-order reasoning. Pooling and confounder feedback yield smaller improvements, indicating that performance is mainly driven by explicit causal inductive biases (Appendix I).

Robustness to Missing Data. In Appendix J, FoundCause remains robust under Missing At Random (MAR) corruption, with minimal degradation at 10% missingness and retaining 92% of its clean-data F_1 at 30%. Imputation-based baselines degrade substantially, highlighting the benefit of explicit mask-aware modeling.

5.2 Dimension Generalization

A key property of FoundCause is the absence of positional embeddings along the variable axis: variables are represented solely through observed samples and induced statistical relationships, making the model permutation-invariant and agnostic to variable order and count. The checkpoint is trained on graphs with $D \in [2, 50]$. To evaluate generalization, we consider synthetic SCMs with $D \in \{50, 60, 70, 80, 90, 100\}$, where $D > 50$ corresponds to zero-shot extrapolation. For each dimension, we generate 20 datasets using DoWhy with $N = 600$ samples, varying root fractions in $[0.05, 0.40]$, latent-confounder fractions in $[0, 0.30]$, and mechanism types from the same mixture as training (linear, nonlinear additive, and non-additive neural mechanisms with heterogeneous noise). All evaluations use a single fixed checkpoint (Table 1) without retraining, fine-tuning, or dimension-specific calibration.

We report macro-averaged AUROC, AUPRC, and directed-edge F_1 over the 20 tasks per dimension. AUROC evaluates threshold-independent ranking, AUPRC emphasizes performance under sparsity, and F_1 measures the final thresholded graph. The goal is not constant performance beyond the training range, but to assess whether degradation is graceful and representations remain effective as D increases.

Dataset	D	AUROC ↑	F_1 ↑	Skel-F_1 ↑	SHD/d ↓
Hailfinder [1]	56	0.742	0.516	0.603	1.22
HEPAR II [39]	70	0.715	0.421	0.581	1.58
WIN95PT [46]	76	0.789	0.476	0.558	1.69
DREAM4-100 [30, 29]	100	0.661	0.341	0.522	2.04

Table 3: Dimension-generalization of `FoundCause` on additional real-world datasets.

Figure 2 shows that `FoundCause` retains strong edge-ranking ability well beyond its training range: at $D = 100$ (twice the training ceiling), AUROC remains 0.712, indicating many true edges are still ranked above non-edges, while F_1 drops from 0.554 at $D = 50$ to 0.278. This gap suggests that extrapolation errors are driven by miscalibration rather than degraded representations. As D increases, the number of candidate pairs grows as $D(D-1)$, shifting the logit distribution away from the regime for which the decision threshold is calibrated. Thus, relative edge scores remain informative, but the fixed threshold becomes increasingly mismatched to the true graph density.

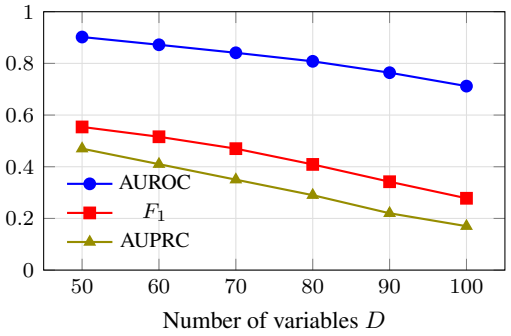

Figure 2: Performance vs. dimension. Macro-averaged AUROC, AUPRC, and F_1 of `FoundCause` as a function of the number of variables D, evaluated on 20 synthetic DoWhy datasets per dimension. The model is trained on $D \in [2, 50]$, and results for $D \geq 60$ represent zero-shot extrapolation. AUROC degrades gradually with increasing D, while F_1 declines more rapidly as the fixed decision threshold—calibrated on in-distribution logits—becomes suboptimal at larger dimensions.

Real Data Experiments. We further evaluate this behavior on four high-dimensional real-world datasets with $D \in [56, 100]$, using the same fixed checkpoint and inference pipeline without retraining or recalibration.

Table 3 provides a real-world stress test of dimension generalization beyond the synthetic DoWhy setting. Despite variation in domain, graph density, and data-generating mechanisms, `FoundCause` retains meaningful performance without retraining or dataset-specific calibration, with AUROC ranging from 0.661 to 0.789. Consistent with synthetic results, thresholded metrics degrade with increasing dimension: F_1 decreases from 0.516 on HAILFINDER to 0.341 on DREAM4_100, while SHD/d increases from 1.22 to 2.04. In contrast, skeleton scores remain relatively stable, suggesting reliable adjacency detection but reduced accuracy in edge orientation and calibration as the number of candidate pairs grows. Overall, these results reinforce graceful zero-shot generalization: ranking quality is preserved better than thresholded accuracy as D increases. Section K compares SEA and CSIvA; `FoundCause` remains within 0.02–0.05 in F_1 of the best method on three of four datasets and achieves the highest AUROC on WIN95PTS.

6 Conclusion

We presented `FoundCause`, an amortized model for causal discovery from observational data that jointly predicts a directed acyclic graph and a latent confounding matrix in a single forward pass. The architecture combines a permutation-invariant transformer encoder with explicit statistical inductive biases, including pairwise statistics, a factored edge predictor, triangular refinement for higher-order reasoning, and a noisy-OR confounder module. Trained entirely on synthetic structural causal models with anti-shortcut augmentation, `FoundCause` achieves state-of-the-art average F_1, AUROC, and SHD/d across 15 real-world benchmarks, while being the only method that consistently runs on all datasets and explicitly models latent confounding. Ablation studies highlight the importance of pairwise statistical features and triangular refinement, while extrapolation experiments demonstrate that edge ranking degrades gracefully beyond the training regime. Together, these results suggest that combining amortized inference with structured inductive biases is a promising direction for scalable causal discovery. We plan to open-source `FoundCause` to support reproducibility and accelerate research in the broader scientific community. We hope this work serves both as a practical tool for applied causal analysis and as a step toward foundation models that natively incorporate causal reasoning.

References

[1] Bruce Abramson, John Brown, Ward Edwards, Allan Murphy, and Robert L Winkler. Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

[2] Matthew Ashman, Chao Ma, Agrin Hilmkil, Joel Jennings, and Cheng Zhang. Causal reasoning in the presence of latent confounders via neural ADMG learning. In *International Conference on Learning Representations*, 2023.

[3] Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Junwei Ma, Bingru Li, Jesse C. Cresswell, and Rahul Krishnan. CausalPFN: Amortized causal effect estimation via in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[4] 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.

[5] Rohit Bhattacharya, Tushar Nagarajan, Daniel Malinsky, and Ilya Shpitser. Differentiable causal discovery under unmeasured confounding. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2314–2322. PMLR, 2021.

[6] Patrick Blöbaum, Peter Götz, Kailash Budhathoki, Atalanti A Mastakouri, and Dominik Janzing. Dowhy-gcm: An extension of dowhy for causal inference in graphical causal models. *Journal of Machine Learning Research*, 25(147):1–7, 2024.

[7] Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[8] Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42, 10 2013.

[9] Patrick Chao, Patrick Blöbaum, Sapan Patel, and Shiva Prasad Kasiviswanathan. Modeling causal mechanisms with diffusion models for interventional and counterfactual queries. *Trans. Mach. Learn. Res.*, 2024, 2023.

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

[11] Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3921–3962, 2014.

[12] Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37:9974–10007, 2024.

[13] Juan L Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for ai methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025.

[14] Tomas Geffner, Javier Antoran, Adam Foster, Wenbo Gong, Chao Ma, Emre Kiciman, Amit Sharma, Angus Lamb, Martin Kukla, Nick Pawlowski, Agrin Hilmkil, Joel Jennings, Meyer Scetbon, Miltiadis Allamanis, and Cheng Zhang. Deep end-to-end causal inference. *Transactions on Machine Learning Research*, 2024.

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

[16] Michaela Hardt, William Roy Orchard, Patrick Blöbaum, Elke Kirschbaum, and Shiva Kasiviswanathan. The petshop dataset—finding causes of performance issues across microservices. In *Causal Learning and Reasoning*, pages 957–978. PMLR, 2024.

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[18] P. Hoyer, D. Janzing, J. Mooij, J. Peters, and B Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Proceedings of the conference Neural Information Processing Systems (NIPS) 2008*, Vancouver, Canada, 2009. MIT Press.

[19] John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

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

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

[22] Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Uncertainty in Artificial Intelligence*, pages 1052–1062. PMLR, 2022.

[23] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant input. In *International Conference on Machine Learning (ICML)*, 2019.

[24] Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclicity constraints. In *International Conference on Learning Representations*, 2022.

[25] Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. DiBS: Differentiable bayesian structure learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[26] Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Pingchuan Ma, Rui Ding, Qiang Fu, Jiaru Zhang, Shuai Wang, Shi Han, and Dongmei Zhang. Scalable differentiable causal discovery in the presence of latent confounders with skeleton posterior. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2141–2152. Association for Computing Machinery, 2024.

[28] Divyat Mahajan, Jannes Gladrow, Agrin Hilmkil, Cheng Zhang, and Meyer Scetbon. Amortized inference of causal models via conditional fixed-point iterations. *Transactions on Machine Learning Research*, 2025. J2C Certification.

[29] Daniel Marbach, Robert J. Prill, Thomas Schaffter, Claudio Mattiussi, Dario Floreano, and Gustavo Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences*, 107(14):6286–6291, 2010.

[30] Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

[31] Alexander Marx and Jilles Vreeken. Identifiability of cause and effect using regularized regression. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 852–861, 2019.

[32] Francesco Montagna, Max Cairney-Leeming, Dhanya Sridhar, and Francesco Locatello. Demystifying amortized causal discovery with transformers. *Transactions on Machine Learning Research*, 2025.

[33] Joris M. Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

[34] Arash Nasr-Esfahany, Mohammad Alizadeh, and Devavrat Shah. Counterfactual identifiability of bijective causal models. In *Forty-second International Conference on Machine Learning*, 2023.

[35] Achille Nazaret and David Blei. Extremely greedy equivalence search. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

[36] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:17943–17954, 2020.

[37] Hamed Nilforoshan, Michael Moor, Yusuf Roohani, Yining Chen, Anja Šurina, Michihiro Yasunaga, Sara Oblak, and Jure Leskovec. Zero-shot causal learning. *Advances in Neural Information Processing Systems*, 36:6862–6901, 2023.

[38] Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume 52 of *Proceedings of Machine Learning Research*, pages 368–379. PMLR, 2016.

[39] Agnieszka Onisko. Probabilistic causal models in medicine: Application to diagnosis of liver disorders. In *Ph. D. dissertation, Inst. Biocybern. Biomed. Eng., Polish Academy Sci., Warsaw, Poland*, 2003.

[40] J. Peters and P. Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 03 2014.

[41] 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. *Advances in Neural Information Processing Systems*, 36:785–807, 2023.

[42] Arik Reuter, Anish Dhir, Cristiana Diaconu, Jake Robertson, Ole Ossen, Frank Hutter, Adrian Weller, Mark van der Wilk, and Bernhard Schölkopf. Use what you know: Causal foundation models with partial graphs. *arXiv preprint arXiv:2602.14972*, 2026.

[43] Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *arXiv preprint arXiv:2506.06039*, 2025.

[44] 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 *International Conference on Machine Learning*, pages 18741–18753. PMLR, 2022.

[45] 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.

[46] Marco Scutari. Bayesian network repository: Large discrete bayesian networks. `https://www.bnlearn.com/bnrepository/discrete-large.html`, 2022. Accessed: 2026-05-03.

[47] Amit Sharma and Emre Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

[48] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[49] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

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

[51] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

[52] Gideon Stein, Maha Shadaydeh, and Joachim Denzler. Embracing the black box: Heading towards foundation models for causal discovery from time series data. *arXiv preprint arXiv:2402.09305*, 2024.

[53] Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, 41(2):437–463, 2013.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] Menghua Wu, Yujia Bao, Regina Barzilay, and Tommi Jaakkola. Sample, estimate, aggregate: A recipe for causal discovery foundation models. *Transactions on Machine Learning Research*, 2025.

[56] Sascha Xu, Osman A Mian, Alexander Marx, and Jilles Vreeken. Inferring cause and effect in the presence of heteroscedastic noise. In *International Conference on Machine Learning*, pages 24615–24630. PMLR, 2022.

[57] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7154–7163. PMLR, 2019.

[58] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[59] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008.

[60] K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, 2009.

[61] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A Comparison between Amortized Causal Discovery Methods

Method	Latent confounders	Missing data	Variable-count agnostic	Nonlinear mechanisms	V-structure / structural reas.	Pairwise stat. features	Explicit DAG	Heterogeneous populations	Real-world benchmarks
AVICI [26]	x	x	✓	✓	x	x	x	x	∼
CSIvA [20]	x	x	✓	✓	x	x	x	x	∼
Montagna et al. (2024) [32]	x	x	✓	✓	x	x	x	x	x
Cond_FIP [28]	x	x	✓	✓	x	x	✓	x	∼
SEA [55]	x	x	✓	✓	∼	✓	x	x	∼
Causal Pretraining (TS) [52]	x	x	∼	✓	x	x	x	x	∼
`FoundCause` (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4: Feature comparison with existing foundation-model and amortized causal discovery methods for observational data. ✓ denotes full support, x denotes lack of support, and ∼ indicates partial support or reliance on post-hoc extensions. `FoundCause` is the only method that simultaneously supports latent-confounder prediction, native handling of missing data, variable-count agnostic inference beyond the training range, and explicit DAG post-processing.

B Evaluation Metrics

Let $D^* \in \{0,1\}^{d\times d}$ denote the ground-truth directed adjacency matrix and let $\widehat{D} \in \{0,1\}^{d\times d}$ denote the predicted directed adjacency matrix, with diagonal entries excluded. We evaluate directed-edge recovery using precision, recall, and F_1:

$$\mathrm{Prec}_{\mathrm{dir}} = \frac{\sum_{i\neq j} \widehat{D}_{ij} D^*_{ij}}{\sum_{i\neq j} \widehat{D}_{ij}}, \qquad \mathrm{Rec}_{\mathrm{dir}} = \frac{\sum_{i\neq j} \widehat{D}_{ij} D^*_{ij}}{\sum_{i\neq j} D^*_{ij}},$$

and

$$F_1 = \frac{2\,\mathrm{Prec}_{\mathrm{dir}}\mathrm{Rec}_{\mathrm{dir}}}{\mathrm{Prec}_{\mathrm{dir}} + \mathrm{Rec}_{\mathrm{dir}}}.$$

Thus, a predicted edge $i \to j$ is counted as correct only when the true DAG also contains $i \to j$; predicting $j \to i$ is treated as an orientation error.

For direction-agnostic adjacency recovery, we report skeleton F_1. Define the true and predicted skeletons over unordered pairs by

$$S^*_{ij} = \mathbb{1}\{D^*_{ij} = 1 \text{ or } D^*_{ji} = 1\}, \qquad \widehat{S}_{ij} = \mathbb{1}\{\widehat{D}_{ij} = 1 \text{ or } \widehat{D}_{ji} = 1\}, \qquad i < j.$$

Skeleton precision and recall are then

$$\mathrm{Prec}_{\mathrm{skel}} = \frac{\sum_{i<j} \widehat{S}_{ij} S^*_{ij}}{\sum_{i<j} \widehat{S}_{ij}}, \qquad \mathrm{Rec}_{\mathrm{skel}} = \frac{\sum_{i<j} \widehat{S}_{ij} S^*_{ij}}{\sum_{i<j} S^*_{ij}},$$

with

$$\text{Skel-}F_1 = \frac{2\,\mathrm{Prec}_{\mathrm{skel}}\mathrm{Rec}_{\mathrm{skel}}}{\mathrm{Prec}_{\mathrm{skel}} + \mathrm{Rec}_{\mathrm{skel}}}.$$

This metric measures whether the correct variable pairs are connected, regardless of edge orientation.

We also report structural Hamming distance normalized by graph size, denoted SHD/d. SHD counts the number of graph edits required to transform the predicted graph into the true DAG, including missing edges, extra edges, and incorrectly oriented edges:

$$\mathrm{SHD}/d = \frac{\mathrm{SHD}(\widehat{D}, D^*)}{d}.$$

Normalizing by d makes structural errors more comparable across datasets with different numbers of variables. Lower SHD/d indicates better graph recovery.

Finally, we report AUROC for directed-edge prediction over the ordered pairs (i, j) with $i \neq j$. When probability scores are available, AUROC measures the ability of the model to rank true directed edges

above non-edges. When computed from a binary predicted DAG, AUROC corresponds to the ROC performance at a single operating point induced by the final thresholded graph:

$$\mathrm{AUROC} = \Pr\big(s_{ij} > s_{kl} \mid D^*_{ij} = 1,\ D^*_{kl} = 0\big) + \frac{1}{2}\Pr\big(s_{ij} = s_{kl} \mid D^*_{ij} = 1,\ D^*_{kl} = 0\big)\,,$$

where s_{ij} is either the predicted edge score or the binary prediction $\widehat{D}_{ij}$. Higher AUROC indicates better discrimination between true directed edges and absent directed edges.

C Evaluation Datasets

We evaluate `FoundCause` on 15 real-world and semi-synthetic causal discovery benchmarks spanning $D \in [8, 46]$ variables, which lie within the training range $D \in [2, 50]$, as well as on the Tübingen cause–effect pairs benchmark for bivariate direction. In addition, we include four datasets with $D > 50$ to assess generalization beyond the training regime. None of these datasets, their ground-truth DAGs, or any derived structural information are used during training; the model is trained *exclusively* on synthetic SCMs (Section G). Table 5 summarizes the benchmark suite, and we briefly describe each dataset family below.

Dataset	**Domain**	D	N	$\lvert E\rvert$	ρ	**Type**
Semi-synthetic Bayesian networks (bnlearn repository [46])						
ASIA	Medical (lung cancer)	8	1,000	8	0.143	Discrete BN
ASIA_NONLINEAR	Medical (lung cancer)	8	1,000	8	0.143	Nonlinear variant
CHILD	Medical (congenital heart)	20	1,000	25	0.066	Discrete BN
CHILD_NONLINEAR	Medical (congenital heart)	20	1,000	25	0.066	Nonlinear variant
INSURANCE	Insurance risk assessment	27	1,500	52	0.074	Discrete BN
ALARM_LIKE	Medical (ICU monitoring)	37	2,000	46	0.035	Linear Gaussian
ECOLI_LIKE	Gene regulation (E. coli)	46	1,000	58	0.028	Linear Gaussian
Real biological data [45]						
SACHS_OBS	Protein signalling (obs.)	11	853	17	0.154	Observational
SACHS_FULL	Protein signalling (mixed)	11	7,466	17	0.154	Flow cytometry
SACHS_NONLINEAR	Protein signalling	11	1,000	17	0.154	Nonlinear variant
Real physical / operational systems						
CAUSAL_CHAMBERS_LT	Light tunnel (ETH Zürich)	20	10,000	39	0.103	Physical measurements
PETSHOP_HIGH_TRAFFIC	Microservice telemetry	39	589	42	0.028	Operational metrics
PETSHOP_LOW_TRAFFIC	Microservice telemetry	41	589	43	0.026	Operational metrics
PETSHOP_TEMP1	Microservice (temporal)	41	1,652	43	0.026	Time-series metrics
PETSHOP_TEMP2	Microservice (temporal)	41	1,652	43	0.026	Time-series metrics
Large Bayesian networks (bnlearn repository [46], beyond training range)						
HAILFINDER	Weather forecasting	56	5,000	66	0.021	Discrete BN
HEPAR2	Medical (liver disorders)	70	5,000	123	0.025	Discrete BN
WIN95PTS	Printer troubleshooting	76	5,000	112	0.020	Discrete BN
Bivariate direction benchmark						
TUEBINGEN	Mixed (102 pairs)	2	variable	—	0.5	Real-world pairs

Table 5: Evaluation benchmarks. D denotes the number of observed variables, N the number of samples used at inference, $|E|$ the number of edges in the ground-truth DAG, and $\rho = |E|/(D(D-1))$ the edge density. *Type* indicates the data source, including discrete Bayesian Networks (BN), nonlinear variants, real observational datasets, and physical measurement systems.

Semi-synthetic Bayesian Networks. ASIA, CHILD, INSURANCE, and ALARM_LIKE are derived from the bnlearn repository [46]. Each dataset is generated from a fixed expert-designed directed acyclic graph (DAG) with discrete conditional probability tables (CPTs), and observations are obtained via ancestral sampling. The ground-truth graph is therefore known exactly, while the observed data follow discrete generative assumptions. ASIA is a classic eight-variable lung cancer

network; CHILD models congenital heart disease diagnosis; INSURANCE captures insurance risk factors; and ALARM_LIKE is based on an ICU monitoring network.

We additionally evaluate on ASIA_NL and CHILD_NL, which retain the same DAG structures but apply nonlinear transformations (e.g., $\tanh$, quadratic, exponential) with additive Gaussian noise, yielding continuous observations under the same causal graph. ECOLI_LIKE is a linear-Gaussian SCM constructed from a published *E. coli* gene regulatory network (46 variables, 58 edges), providing a biologically motivated benchmark at moderate scale.

Sachs Protein Signaling. The Sachs dataset [45] contains measurements of 11 phosphorylated proteins and phospholipids collected via flow cytometry under multiple experimental conditions. SACHS_OBS uses the observational (unperturbed) subset ($N = 853$), SACHS_FULL uses the full dataset including interventional samples ($N = 7{,}466$, treated as observational during evaluation), and SACHS_NL is a nonlinear re-simulation based on the same underlying DAG. All three variants share the 17-edge consensus graph reported in the original study.

Causal Chambers (Light Tunnel). The Causal Chamber light-tunnel system [13] is a physical experimental setup at ETH Zürich consisting of a controllable light source, rotating polarizers, a camera, and wavelength-specific intensity sensors. Variables correspond to sensor measurements and actuator settings, and the ground-truth causal graph is determined by the known physical interactions among these components. The CAUSAL_CHAMBERS_LT dataset contains $D = 20$ variables and $N = 10{,}000$ observations collected under the natural operating regime. This benchmark is notable in that the causal graph is directly specified by the underlying physics, rather than inferred from observational data or expert consensus.

PetShop Operational Telemetry. The PetShop dataset [16] consists of runtime telemetry from a synthetic microservice-based e-commerce system. Each variable represents an operational metric (e.g., latency, error rate, or traffic volume) for a particular service, and the ground-truth DAG is derived from the known service dependency structure. We evaluate four variants: PETSHOP_HIGH_TRAFFIC and PETSHOP_LOW_TRAFFIC, which reflect steady-state regimes ($D \approx 39$–41, $N = 589$), and PETSHOP_TEMP1 and PETSHOP_TEMP2, which include temporal variability ($N = 1{,}652$). These datasets are challenging due to strong correlations induced by service dependencies and non-additive interactions arising from system dynamics such as queuing, retries, and cascading effects.

Large Bayesian Networks (Dimension Extrapolation). To evaluate generalization beyond the training range $D \in [2, 50]$, we consider three expert-designed Bayesian networks from the bnlearn repository [46] with $D \in \{56, 70, 76\}$. HAILFINDER [1] is a 56-node, 66-edge network for severe weather forecasting, constructed from meteorological expert knowledge, with relatively sparse connectivity (average degree 2.36). HEPAR2 [39] is a 70-node, 123-edge medical diagnostic network for liver disorders, with higher edge density (average degree 3.51). WIN95PTS [46] is a 76-node, 112-edge troubleshooting network for Windows 95 printing issues, exhibiting dense local structure with large Markov blankets and in-degree up to 7. For each network, we generate $N = 5{,}000$ samples via ancestral sampling from the corresponding conditional probability tables, using the published DAG as ground truth. These datasets represent strict zero-shot evaluation for `FoundCause`, as no graphs with $D > 50$ are seen during training.

Tübingen Cause–Effect Pairs. We further evaluate bivariate direction identification using the Tübingen cause–effect pairs benchmark [33], which consists of real-world variable pairs with known causal direction across diverse domains. We use the 102 pairs with positive evaluation weights and report both unweighted and weighted accuracy, where weights reflect pair difficulty as defined by the benchmark authors. This benchmark isolates the directionality task independently of edge existence, providing a complementary evaluation of causal orientation performance.

D Inference Protocol

All datasets are evaluated using a fixed inference pipeline. Given an input dataset, we perform $K = 10$ stochastic inference runs, each combining bootstrap resampling and random permutation of variable indices, and average the resulting logits after mapping predictions back to the original

ordering. The averaged logits are calibrated using temperature scaling with $T = 0.65$, estimated on a synthetic validation set, and thresholded using a two-component Gaussian mixture model (GMM). Finally, self-consistency pruning retains only edges that appear in at least 50% of the runs. No per-dataset hyperparameter tuning or fine-tuning is performed: a single fixed checkpoint is used across all datasets.

E Additional Architecture Details

Parameter	Value
Hidden dimension d_h	768
Number of attention heads H	8
Number of encoder layers L	8 (16 alternating blocks)
FFN inner dimension	$\frac{2}{3} \times 4d_h = 2048$
Triangular pair dimension d_{pair}	192
Triangular refinement rounds R	3
Number of confounder tokens K_c	8
Number of global context tokens K_g	12
PMA query tokens K_q	8
PMA attention heads	4
Number of pairwise statistics	45
Logit bias initialization	-2.0

Table 6: Architectural hyperparameters.

Normalization and Feedforward Blocks. Each encoder block operates on input representations $\boldsymbol{H} \in \mathbb{R}^{N \times D \times d_h}$ and outputs tensors of the same shape. All blocks use RMSNorm [58] applied along the feature dimension:

$$\text{RMSNorm}(\boldsymbol{x}) = \frac{\boldsymbol{x}}{\sqrt{d_h^{-1} \sum_{r=1}^{d_h} x_r^2 + \epsilon}} \odot \boldsymbol{\gamma}, \qquad \epsilon = 10^{-6},$$

where $\boldsymbol{x} \in \mathbb{R}^{d_h}$ is a feature vector and $\boldsymbol{\gamma} \in \mathbb{R}^{d_h}$ is a learned scale parameter.

The feedforward sublayer uses a SwiGLU activation [48], mapping $\boldsymbol{x} \in \mathbb{R}^{d_h}$ to $\mathbb{R}^{d_h}$ via

$$\text{SwiGLU}(\boldsymbol{x}) = \boldsymbol{W}_2 \left[\text{SiLU}(\boldsymbol{W}_g \boldsymbol{x}) \odot \boldsymbol{W}_1 \boldsymbol{x} \right],$$

where $\boldsymbol{W}_g, \boldsymbol{W}_1 \in \mathbb{R}^{d_f \times d_h}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d_h \times d_f}$, with hidden dimension d_f.

Both attention and feedforward outputs are added through residual connections, and output projections are initialized with standard deviation $0.02/\sqrt{2L}$ to stabilize deep stacking of $2L$ blocks. No dropout is used in the encoder; instead, regularization is provided by continuous data regeneration and feature-level dropout applied to selected pairwise statistics.

Multi-layer Feature Fusion. Let $\boldsymbol{H}^{(\ell)} \in \mathbb{R}^{N \times D \times d_h}$ denote the encoder output at layer ℓ. We retain a subset of intermediate representations $\{\boldsymbol{H}^{(\ell_k)}\}_{k=1}^{4}$, including the final layer, and combine them prior to pooling via a learned convex combination:

$$\boldsymbol{H}_{\text{fused}} = \sum_{k=1}^{4} \frac{\exp(\alpha_k)}{\sum_{k'} \exp(\alpha_{k'})} \boldsymbol{H}^{(\ell_k)},$$

where $\alpha_k \in \mathbb{R}$ are learnable scalar weights.

The fused representation $\boldsymbol{H}_{\text{fused}} \in \mathbb{R}^{N \times D \times d_h}$ is then passed to the pooling module. This fusion allows downstream prediction to leverage causal signals that may emerge at different depths of the encoder.

Pairwise Statistics. Given input data $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, the model computes a tensor of pairwise statistics $\boldsymbol{S} \in \mathbb{R}^{D \times D \times d_s}$ with $d_s = 45$, where $\boldsymbol{s}_{ij} \in \mathbb{R}^{d_s}$ summarizes statistical relationships between variables i and j. These statistics are computed directly from the raw data under `torch.no_grad`, i.e., without gradient updates.

The feature vector $\boldsymbol{s}_{ij}$ is partitioned into four groups: symmetric features (invariant under swapping i and j), antisymmetric features (changing sign or direction under swapping), V-structure features capturing collider-like patterns, and reliability metadata describing statistical quality. The symmetric features include Pearson and Spearman correlation, partial correlation, nonlinear dependence measures, and higher-order moment interactions. The antisymmetric features include residual-variance asymmetry, kernel-based dependence asymmetry (HSIC-RFF), cross-moment asymmetries, regression-based coefficients, conditional-variance features, and directional statistics such as Chatterjee's ξ and IGCI-style scores. The V-structure features capture patterns indicative of collider structures, while the reliability metadata include the covariance condition number, the ratio D/N, the unique-value ratio, and the normalized observation count.

These metadata features are always retained during training, allowing the model to learn when other statistics are unreliable, while the remaining features may be stochastically dropped for regularization.

Stat-conditioned Attention Bias. The pairwise statistics $\boldsymbol{S}$ are used to construct attention biases for variable-wise self-attention. For each layer ℓ and attention head $h \in \{1, \ldots, H\}$, a nonlinear projection maps $\boldsymbol{s}_{ij}$ to a scalar bias:

$$b_{ij}^{(h,\ell)} = \left[\boldsymbol{W}_2^{(\ell)} \,\mathrm{GELU}\Big(\boldsymbol{W}_1^{(\ell)} \boldsymbol{s}_{ij}\Big)\right]_h ,$$

where $\boldsymbol{W}_1^{(\ell)} \in \mathbb{R}^{d_b \times d_s}$, $\boldsymbol{W}_2^{(\ell)} \in \mathbb{R}^{H \times d_b}$, and $d_b = 48$ is the hidden dimension of the bias network. The resulting biases form a tensor $\boldsymbol{B}^{(\ell)} \in \mathbb{R}^{H \times D \times D}$, which is added to the attention logits.

To ensure numerical stability, biases are clamped to the range $[-10, 10]$. During training, antisymmetric and V-structure features are subjected to inverted dropout, while symmetric features and reliability metadata are always retained. This design allows the model to leverage classical dependence cues while learning robustness to noisy or unreliable statistical signals.

Encoder-conditioned Feature Gate. For each ordered pair (i, j), the direction head constructs a feature gate $\boldsymbol{g}_{ij} \in [0, 1]^{d_a}$, where d_a is the number of antisymmetric statistics (here $d_a = 20$). The gate is computed from the variable embeddings $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{R}^{d_h}$ and pairwise statistics $\boldsymbol{s}_{ij} \in \mathbb{R}^{d_s}$ as

$$\boldsymbol{g}_{ij} = \sigma(\boldsymbol{W}_2 \,\mathrm{GELU}(\boldsymbol{W}_1[\boldsymbol{z}_i - \boldsymbol{z}_j; \boldsymbol{s}_{ij}])) ,$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d_g \times (d_h + d_s)}$, $\boldsymbol{W}_2 \in \mathbb{R}^{d_a \times d_g}$, and d_g is a hidden dimension. The gate modulates antisymmetric features used for direction prediction, allowing the model to adaptively select informative causal signals for each variable pair.

To prevent saturation, we apply a soft bounds regularization that penalizes gate values close to 0 or 1, encouraging flexible feature usage across pairs.

Triangular Refinement. Let $\boldsymbol{E} \in \mathbb{R}^{D \times D \times d_{\text{pair}}}$ denote the pairwise representation tensor, initialized from the concatenated existence and direction features. We refine $\boldsymbol{E}$ for R rounds using triangle-based updates that aggregate information over intermediate variables $k \in \{1, \ldots, D\}$. Each round consists of outgoing, incoming, and collider updates, followed by row-wise self-attention over pairs.

The outgoing update is given by

$$\Delta \boldsymbol{e}_{ij}^{\text{out}} = \frac{1}{\sqrt{D}} \sum_{k=1}^{D} \left(\frac{1}{2} \tanh(\boldsymbol{w}_g^\top \boldsymbol{e}_{ik}) + \frac{1}{2}\right) \boldsymbol{W}_v \boldsymbol{e}_{kj},$$

where $\boldsymbol{e}_{ij} \in \mathbb{R}^{d_{\text{pair}}}$, $\boldsymbol{w}_g \in \mathbb{R}^{d_{\text{pair}}}$, and $\boldsymbol{W}_v \in \mathbb{R}^{d_{\text{pair}} \times d_{\text{pair}}}$. Incoming and collider updates use analogous gated aggregation with index permutations corresponding to fork and collider motifs.

Each update is followed by normalization and residual connections, and the refined tensor remains in $\mathbb{R}^{D \times D \times d_{\text{pair}}}$. The final refined logits are obtained from $\boldsymbol{E}$ and combined with the base decoder outputs using learned mixing weights for skeleton and direction components. This refinement enables reasoning over higher-order causal structures such as chains, forks, and colliders.

Learned Skeleton Extraction. To obtain an undirected skeleton from directed logits, the model learns a smooth function $\mathrm{skel\_fn} : \mathbb{R}^2 \to \mathbb{R}$ applied to each pair (i, j):

$$\mathrm{skel\_fn}(\ell_{ij}, \ell_{ji}) = \boldsymbol{w}^\top \left[\ell_{ij} + \ell_{ji},\, \ell_{ij}\ell_{ji}\right] + b,$$

where $\boldsymbol{w} \in \mathbb{R}^2$ and $b \in \mathbb{R}$ are learned parameters. The resulting scalar is used as the skeleton logit for edge existence between i and j. This formulation provides a differentiable alternative to hard max or OR operations, while preserving information from both directions.

Confounder Implementation Details. Let $\boldsymbol{Z} \in \mathbb{R}^{D \times d_h}$ denote the variable embeddings and $\boldsymbol{T} \in \mathbb{R}^{K_c \times d_h}$ the learnable confounder tokens. The tokens attend to $\boldsymbol{Z}$ via two cross-attention layers, producing updated token representations $\boldsymbol{T}' \in \mathbb{R}^{K_c \times d_h}$. Each layer consists of multi-head attention followed by a feedforward network with GELU activation and LayerNorm, preserving the token dimensionality.

For each variable i and confounder k, a loading network computes $S_{ik} \in (0, 1)$, and pairwise confounding probabilities are obtained via the noisy-OR aggregation

$$\hat{C}_{ij} = 1 - \prod_{k=1}^{K_c} (1 - S_{ik} S_{jk}).$$

This computation is performed in float32 for numerical stability, and the resulting probabilities are clamped before conversion to logits.

The loading network bias is initialized to -2.0, so that initial loadings satisfy $S_{ik} \approx \sigma(-2) \approx 0.12$, encouraging sparse confounder assignments at the start of training. In addition, a pairwise feature projection maps symmetric and gated antisymmetric statistics $\boldsymbol{s}_{ij}$ to a scalar bias, which is added to the noisy-OR logit prior to symmetrization and masking. The final output is a symmetric matrix $\hat{\boldsymbol{C}} \in [0, 1]^{D \times D}$ of confounding probabilities.

Acyclicity. Let $\boldsymbol{\ell} \in \mathbb{R}^{D \times D}$ denote the directed edge logits and $\sigma(\boldsymbol{\ell})$ the corresponding edge probabilities. An optional acyclicity penalty is defined using the spectral radius $\rho(\sigma(\boldsymbol{\ell}))$, estimated via power iteration. The penalty takes the form

$$\mathcal{L}_{\text{acyc}} = \text{ReLU}\big(\rho(\sigma(\boldsymbol{\ell})) - 1\big),$$

which encourages the learned graph to be acyclic.

In the default configuration, this penalty is disabled during training. Instead, acyclicity is enforced at inference time via greedy DAG post-processing applied to $\sigma(\boldsymbol{\ell})$, ensuring a valid directed acyclic graph.

F Loss Function Details

As introduced in (1), the training objective combines a primary directed-edge binary cross-entropy loss with a set of auxiliary terms that separately regularize directionality, skeleton recovery, latent confounding, calibration, sparsity, and representation diversity. This decomposition reflects the fact that causal discovery contains several coupled but distinct subproblems: detecting whether two variables are adjacent, orienting the edge correctly, avoiding overconfident reverse predictions, accounting for hidden common causes, and controlling graph density. In particular, $\mathcal{L}_{\text{asym}}$ and $\mathcal{L}_{\text{bdir}}$ provide explicit supervision for edge orientation, $\mathcal{L}_{\text{skel}}$ gives a direction-agnostic signal for adjacency recovery, and $\mathcal{L}_{\text{conf}}$ trains the confounder module to identify latent common causes. The calibration, false-positive, density, PMA-diversity, and gate-regularization terms stabilize training and improve transfer by discouraging pathological solutions such as saturated logits, overly dense graphs, collapsed pooling queries, or feature gates that over-specialize to synthetic mechanisms.

The auxiliary losses in Table 7 are weighted to provide targeted gradients without dominating the primary edge-prediction objective. Directional losses use dead zones or activation gates so that they focus on ambiguous or incorrectly oriented edges rather than indefinitely pushing already-correct logits to extremes. The skeleton and confounding losses use capped positive-class weights to address class imbalance while preventing rare positive labels from overwhelming shared encoder representations. The false-positive and density penalties control graph sparsity at local and global levels, respectively, while the calibration penalty limits excessive logit gaps that can harm transfer under distribution shift. Finally, $\mathcal{L}_{\text{pma}}$ encourages the attention-pooling queries to specialize to distinct sample-distribution patterns, and $\mathcal{L}_{\text{gate}}$ keeps the statistic gate from saturating, preserving adaptive use of pairwise causal features across datasets.

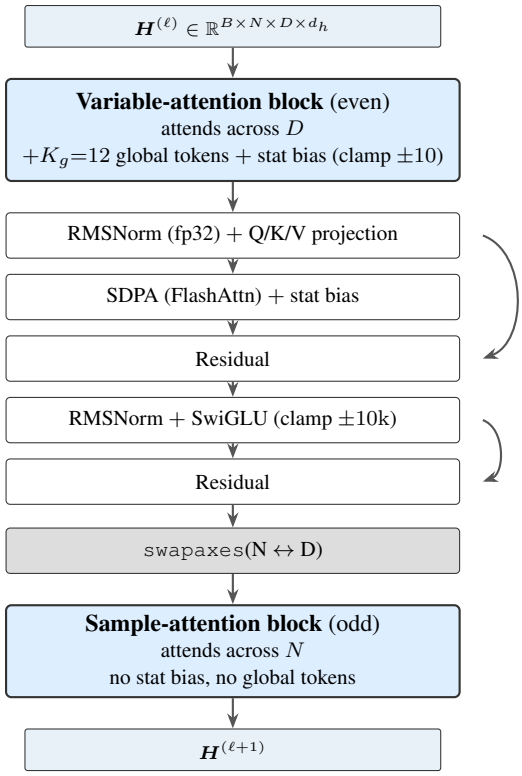

Figure 3: **Encoder Block Pair (one of eight).** The encoder consists of $2L = 16$ attention blocks that alternate between variable-axis attention (even-indexed blocks) and sample-axis attention (odd-indexed blocks). Variable-attention blocks attend across variables and incorporate stat-conditioned attention bias and $K_g = 12$ global context tokens, while sample-attention blocks attend across samples and use neither. Each block follows a pre-normalization design with RMSNorm (computed in float32 with $\varepsilon = 10^{-6}$), multi-head scaled dot-product attention (SDPA), and a SwiGLU feedforward network. SDPA uses FlashAttention with a padding mask value of -10^9 to ensure numerical stability under bfloat16 precision. The SwiGLU gated product is clamped to $\pm 10^4$ before the output projection. Residual output projections use scaled initialization with standard deviation $0.02/\sqrt{2L}$. An additional RMSNorm is applied to the residual stream after every fourth block.

G Synthetic Training Dataset Creation

`FoundCause` is trained *exclusively on synthetic* structural causal models (SCMs); no real-world dataset is ever used for training. All 15 benchmarks reported in Section 5 and the Tübingen cause–effect pairs are strictly held out. The training distribution is constructed to span the breadth of generative assumptions encountered in practice, namely, graph topology, functional mechanism, noise family, dimensionality, sample size, missingness, confounding, so that the learned amortized inference transfers zero-shot to real data.

Continuous Regeneration. Rather than keeping a fixed training set, the data pipeline regenerates ∼12,000 fresh tasks (10,000 multivariate plus 2,000 bivariate) at the start of every epoch using a pool of CPU workers. Each epoch therefore exposes the model to previously unseen SCMs; the model never sees the same dataset twice. This continuous regeneration acts as the sole regulariser (no dropout is used anywhere in the architecture).

Task Definition. A training task is a tuple $(\boldsymbol{X}, \boldsymbol{M}, \boldsymbol{D}^*, \boldsymbol{C}^*, \boldsymbol{v})$ where

- $\boldsymbol{X} \in \mathbb{R}^{N\times D_{\max}}$ is the observational data matrix (padded to $D_{\max}$=65);
- $\boldsymbol{M} \in \{0,1\}^{N\times D_{\max}}$ is the observation mask (observed versus missing);
- $\boldsymbol{D}^* \in \{0,1\}^{D_{\max}\times D_{\max}}$ is the ground-truth DAG among *observed* variables (entries outside the observed sub-block are zero);

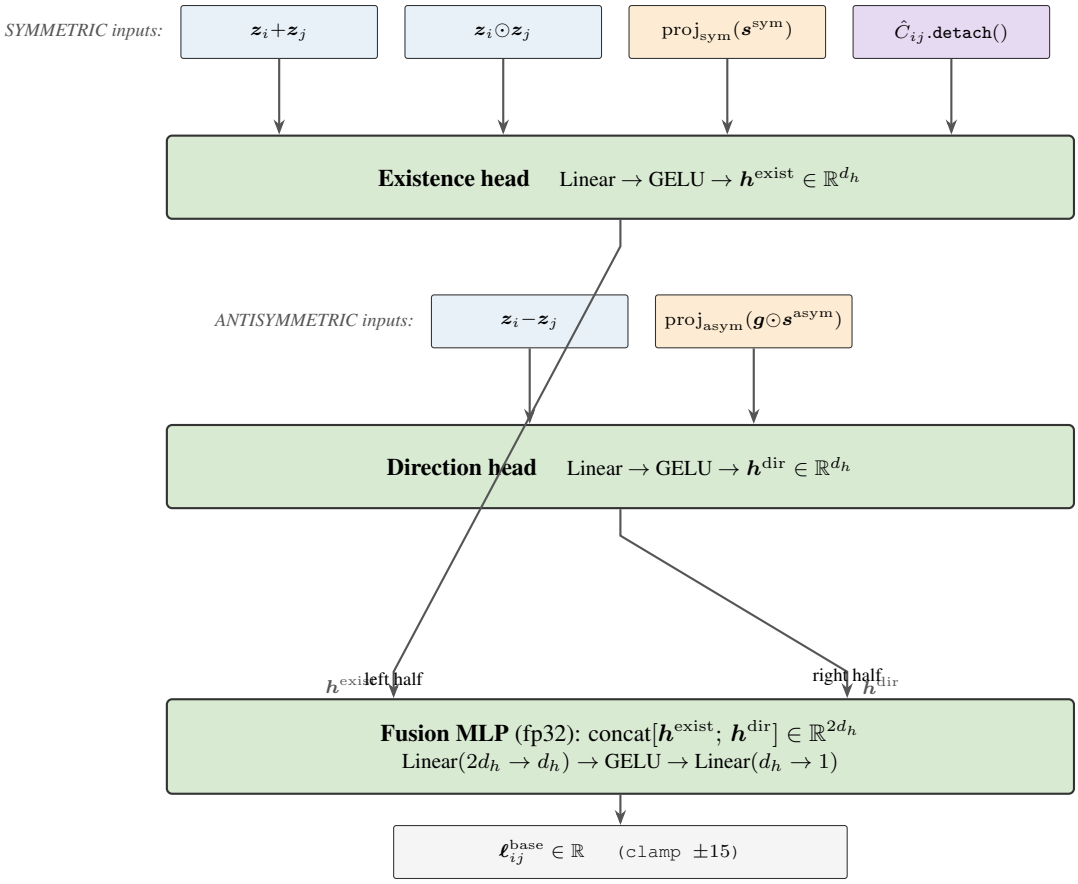

Figure 4: **Factored Edge Predictor.** The model predicts each directed edge (i, j) by separating *edge existence* and *edge direction*, using symmetry-aware features. The *existence head* (top) operates on symmetric inputs, including $\boldsymbol{z}_i + \boldsymbol{z}_j$, $\boldsymbol{z}_i \odot \boldsymbol{z}_j$, a projection of symmetric pairwise statistics, and the confounding score $\hat{C}_{ij}$ (detached from gradients). These features capture whether two variables are related, regardless of direction. The *direction head* (middle) operates on antisymmetric inputs, including $\boldsymbol{z}_i - \boldsymbol{z}_j$ and a projection of encoder-gated asymmetric pairwise statistics. These features capture directional asymmetry between variables. Each head produces a d_h-dimensional hidden representation (computed in float32 for numerical stability). These are concatenated into a $2d_h$-dimensional vector and passed to a fusion MLP (bottom), which combines existence and directional evidence to produce a single logit $\ell_{ij}^{\mathrm{base}}$ for the ordered pair (i, j). The logit is clamped to ± 15 before triangular refinement.

- $\boldsymbol{C}^* \in \{0, 1\}^{D_{\max} \times D_{\max}}$ is the ground-truth confounding matrix (symmetric) indicating pairs sharing a hidden common cause through hidden-only paths;
- $\boldsymbol{v} \in \{0, 1\}^{D_{\max}}$ is the variable-validity mask.

Per task, the number of observed variables is sampled uniformly from $[2, 50]$, the number of samples from $[100, 600]$, and the latent confounder fraction from $[0,\ 0.30]$ of the observed count.

To improve robustness in low-edge-density regimes, 8% of synthetic training tasks are generated as ultra-sparse SCMs, with graph density sampled in the range $[0.01, 0.04]$. In addition, 5% of synthetic tasks are generated as discrete Bayesian networks: conditional probability tables are sampled from Dirichlet distributions, and observations are produced by ancestral sampling from the resulting directed graphical model.

Two Generation Paths. Half of the tasks are produced by the DoWhy path described in Section G.1; the other half use a custom diverse-graph generator supporting scale-free (Barabási–Albert), small-world (Watts–Strogatz), stochastic block model, geometric, and star/hub topologies, together with 18+ nonlinear mechanism families (GP via random Fourier features, tanh and sigmoid MLPs, Michaelis–Menten, Hill, piecewise-linear, polynomial, step, interaction, competitive inhibition,

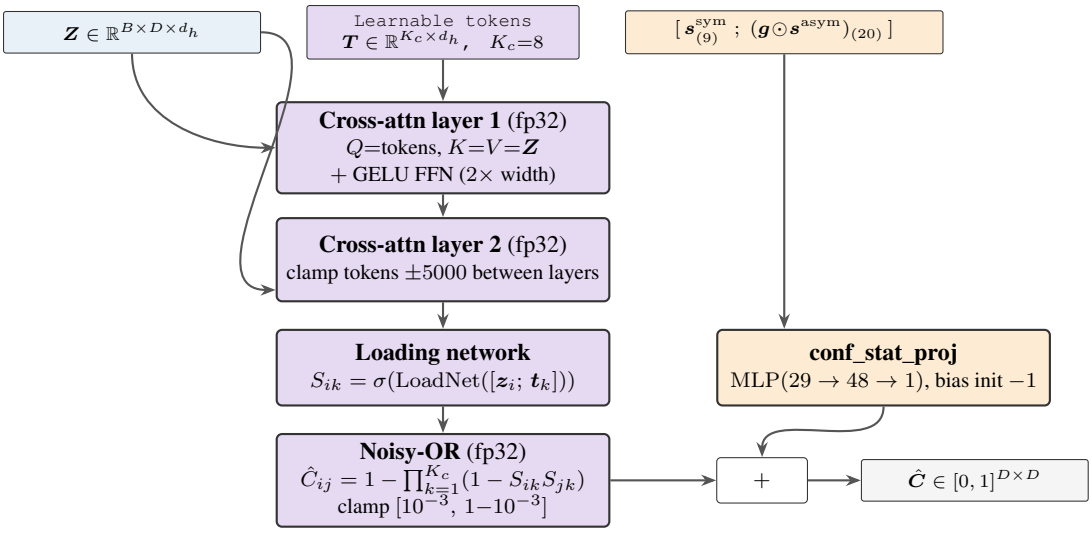

Figure 5: **Confounder Module.** The model represents latent confounders using $K_c = 8$ learnable tokens, which cross-attend to the variable embeddings $\boldsymbol{Z} \in \mathbb{R}^{D\times d_h}$ over two layers. Each layer consists of multi-head attention followed by a GELU feedforward network (with $2\times$ hidden width) and LayerNorm, with computations performed in float32 for numerical stability. Activations are clamped to ± 5000 between layers. A loading network maps each variable–token pair to a probability $S_{ik} \in [0, 1]$, indicating whether confounder k influences variable i. Pairwise confounding probabilities are then computed via noisy-OR aggregation, with computation in float32 and outputs clamped away from 0 and 1. In parallel, a pairwise feature projection maps symmetric and gated antisymmetric statistics to a scalar logit bias, which is added to the noisy-OR logits. The resulting confounding matrix $\hat{\boldsymbol{C}} \in [0, 1]^{D\times D}$ is used both as an output and as input to the edge predictor, where $\hat{C}_{ij}$ is detached to prevent edge-prediction gradients from suppressing confounder learning.

sinusoidal, exponential, logarithmic, log-space ANM, threshold-binary, categorical) and noise distributions including Gaussian, Laplace, uniform, Gaussian mixtures, Cauchy, and Student-t. A separate post-processing pipeline injects real-world artefacts: MCAR / MNAR / MAR missingness (15% of tasks), measurement noise, zero-inflation, discretization, rounding, heteroscedasticity, and log-uniform scale randomization. These anti-shortcut perturbations are applied symmetrically to causes and effects so the model cannot recover direction from marginal-distribution signatures.

Anti-shortcut Augmentation. At batch assembly time, variables are randomly permuted on 100% of training batches (with consistent relabelling of $\boldsymbol{D}^*$ and $\boldsymbol{C}^*$), eliminating the variance-ordering shortcut documented by Reisach et al. (2023). Additionally, 5% of multivariate tasks are replaced with subpopulation-mixture tasks (samples drawn from two independent SCMs, labels set to the union of both DAGs) to teach robustness against heterogeneous populations. A 12% "correlation fog" mode replaces base tasks with near-deterministic (noise_std $\in [0.001, 0.05]$) dense linear SCMs, forcing the model to distinguish direct edges from strong indirect correlation.

Noise Models. To encourage robustness across heterogeneous data-generating processes, synthetic SCMs are generated with a mixture of additive noise distributions. At full training difficulty, Gaussian noise receives weight 30%, while the remaining probability mass is split equally across Laplace, uniform, mixture, and Cauchy noise. This design exposes the model to light-tailed, heavy-tailed, bounded, multimodal, and pathological noise regimes. In particular, the inclusion of Laplace and Cauchy noise tests robustness to heavy-tailed perturbations, uniform noise tests bounded-support mechanisms, and mixture noise introduces multimodality that cannot be summarized by low-order moments alone. Cauchy noise is clipped at $50\times$ the nominal standard-deviation scale used for sampling to avoid numerical instabilities while retaining its extreme-tail behavior.

Post-processing and Special Task Types. After mechanism generation, each synthetic dataset is passed through a stochastic post-processing pipeline designed to mimic common artifacts in real observational data. The pipeline injects autocorrelation with probability 5% and batch effects with probability 3%, both using per-variable random splits; adds independent noise with probability 5%,

Term	Definition	Weight / setting
$\mathcal{L}_{\text{asym}}$	$\frac{1}{n_+}\sum_{(i,j):D^*_{ij}=1}\text{ReLU}(\sigma(\ell_{ji})-\delta)\,V_{ij}$	$\lambda_{\text{asym}}=0.08$, $\delta=0.15$
$\mathcal{L}_{\text{cal}}$	$\frac{1}{n_+}\sum_{(i,j):D^*_{ij}=1}[\text{ReLU}(\lvert\ell_{ij}-\ell_{ji}\rvert-\Delta_{\max})]^2$	$\lambda_{\text{cal}}=0.1$, $\Delta_{\max}=4.0$
$\mathcal{L}_{\text{bdir}}$	$\frac{1}{\lvert\mathcal{A}\rvert}\sum_{(i,j)\in\mathcal{A}}\text{BCE}(\ell_{ij}-\ell_{ji},1)$, where $\mathcal{A}=\{(i,j):D^*_{ij}=1,\ \ell_{ij}>-3 \text{ or } \ell_{ji}>-3\}$	$\lambda_{\text{bdir}}=0.25$
$\mathcal{L}_{\text{skel}}$	$\frac{1}{\lvert V_\triangle\rvert}\sum_{i<j}w^{\text{skel}}_{ij}V_{ij}\text{BCE}\big(\text{skel\_fn}(\ell_{ij},\ell_{ji}),\text{skel\_fn}(D^*_{ij},D^*_{ji})\big)$	$\lambda_{\text{skel}}=0.5$; positive weight capped at 2.0
$\mathcal{L}_{\text{conf}}$	$\frac{1}{B}\sum_{b=1}^{B}\frac{1}{\lvert V^{(b)}\rvert}\sum_{ij}w^{(b)}_{ij}V^{(b)}_{ij}\text{BCE}\Big(\hat{C}^{(b)}_{ij},C^*_{ij}\Big)$	$\lambda_{\text{conf}}=0.2$; per-graph positive weight capped at 5.0
$\mathcal{L}_{\text{fp}}$	$\frac{\lambda^{\text{eff}}_{\text{fp}}}{\lvert V\rvert}\sum_{ij}\text{ReLU}\big(\sigma(\ell_{ij})(1-D^*_{ij})V_{ij}-0.15\big)$, where $\lambda^{\text{eff}}_{\text{fp}}=\lambda_{\text{fp}}\,[1+3\,\text{clamp}(p^*-\hat{p}_t,-0.3,0.3)]$	$\lambda_{\text{fp}}=0.15$, $p^*=0.50$
$\mathcal{L}_{\text{den}}$	$\frac{1}{B}\sum_{b=1}^{B}\frac{\text{softplus}\Big(\sum_{ij}\sigma(\ell^{(b)}_{ij})V^{(b)}_{ij}-d^{(b)}_{\text{eff}}\bar{e}\Big)}{2d^{(b)}_{\text{eff}}}$	$\lambda_{\text{den}}=0.05$, $\bar{e}=\max(2.0,\hat{e}_{\text{ema}})$
$\mathcal{L}_{\text{pma}}$	$\frac{1}{K_q(K_q-1)/2}\sum_{a<b}\text{ReLU}(\cos(\boldsymbol{q}_a,\boldsymbol{q}_b)-0.5)$	$\lambda_{\text{pma}}=0.01$
$\mathcal{L}_{\text{gate}}$	Soft penalty on gate values outside $[0.1,0.9]$	$\lambda_{\text{gate}}=\min(0.5,\max(0,0.8-\hat{t}_{\text{ema}}))$

Table 7: Auxiliary loss terms. All sums are masked by valid pairs unless stated otherwise.

Noise model	**Weight**	**Properties**
Gaussian	30%	Standard additive noise
Laplace	17.5%	Heavy-tailed noise
Uniform	17.5%	Bounded-support noise
Mixture	17.5%	Multimodal noise with 2–4 components
Cauchy	17.5%	Undefined variance; clipped at $50\times$ nominal scale

Table 8: **Noise models used for synthetic SCM generation.** Weights correspond to the full-difficulty training distribution.

measurement noise with probability 7.5%, and outliers with probability 5% at magnitudes between 3 and 8 standard deviations, skipping these continuous perturbations for discrete variables; applies discretization with probability 5% using 5–100 bins; introduces bounded or positive transforms with probability 4%; applies monotone post-transforms such as log, square-root, power, sigmoid, and softplus with probability 3%; adds zero-inflation or censoring with probability 4%; applies rounding noise in the range 0–15%; randomizes variable scales with probability 10%; and introduces extreme scale heterogeneity with probability 5% using multiplicative factors in $10^{[-3,3]}$. To model latent dependence and near-collinearity, the generator also introduces correlated exogenous noise with probability 12%, updating the confounding matrix accordingly, and grouped collinearity with probability 5%, where one source variable is copied into 3–6 noisy variants, again updating the confounding matrix. Additional augmentations include Tübingen-like bivariate transformations for 35% of $d=2$ tasks and log-transform augmentation with probability 6% to create positive, skewed, Sachs-like data. Discrete variables are automatically detected and protected from incompatible continuous-noise perturbations. Finally, the synthetic mixture includes several special task types: bivariate tasks comprise 15% of training data, with 10% easy and 90% full-range difficulty; correlation-fog tasks comprise 12% and use all-linear mechanisms with very low noise in $[0.001, 0.05]$; subpopulation-mixing tasks comprise 5% and combine two SCMs using the union of their edge sets; and intervention-mixture tasks comprise 8% and pool observations across multiple experimental conditions.

G.1 The DoWhy Generator Path

Half of each training batch is generated using the random structural causal model (SCM) generator provided by DoWhy [47, 6]. Data are sampled directly from these SCMs using the library's reference implementation, ensuring consistency with the underlying data-generation procedures.

Graph Topology. DoWhy constructs a DAG by sequential random attachment, yielding graphs approximately following an Erdős–Rényi distribution with edge density drawn from $[0.02, 0.40]$ per task. The root fraction is sampled from $[0.05, 0.40]$. A scale-aware cap restricts expected in-degree to ≤ 4 at large D via $\min(\rho,\ 4/(D-1))$, keeping direct edges detectable at $D{=}50$.

Mechanism Types. For each non-root node Y, the data-generating process assigns a structural mechanism based on a fixed probability of selecting a linear model (0.30), with the remaining probability split evenly across two nonlinear families. Let $\boldsymbol{X}_{\mathrm{Pa}(Y)}$ denote the parent variables of Y.

- *Linear* (30%):
$$Y = \boldsymbol{w}^\top \boldsymbol{X}_{\mathrm{Pa}(Y)} + \varepsilon,$$
where $\boldsymbol{w}$ is sampled from a Gaussian distribution and ε is independent noise.
- *Nonlinear additive MLP* (35%):
$$Y = f_{\mathrm{NN}}(\boldsymbol{X}_{\mathrm{Pa}(Y)}) + \varepsilon,$$
where f_{NN} is a multilayer perceptron (2–4 layers, 4–64 hidden units per layer) with spectral normalization applied to the weights and a scaling factor to induce strong nonlinearity.
- *Non-additive noise MLP* (35%):
$$Y = f_{\mathrm{NN}}([\boldsymbol{X}_{\mathrm{Pa}(Y)}; \varepsilon]),$$
where the noise ε is concatenated with the parent variables before being processed by the network, resulting in non-additive noise effects.

To stabilize the range of generated values, we perform an initial calibration step by drawing $1{,}000$ samples from the SCM and recording per-variable output ranges. Subsequently, outputs of neural mechanisms are rescaled to lie within $[-2, 2]$, preventing value explosion as signals propagate through the graph.

Noise Distributions. DoWhy supports Gaussian, Laplace, uniform, and Student-t additive noise (weighted $0.30/0.30/0.20/0.20$ respectively) with log-uniform scale drawn from $[0.01,\ 0.25]$. Root variables additionally sample from log-normal, exponential, β, and χ^2 distributions with 65% of root marginals replaced by a 2–6 component Gaussian mixture (component $\mathrm{std}{=}0.4$, means in $[-3, 3]$) to ensure multimodal root behaviour.

Hidden Confounders. To induce latent-confounding structure, DoWhy generates an extended SCM with $D + K_H$ variables where K_H equals the requested latent count; a weighted selection favouring nodes with ≥ 2 children is then applied to hide variables. The ground-truth DAG $\boldsymbol{D}^*$ over observed variables is computed as the transitive closure through hidden-only paths (so an observed i causes observed j whenever there is a directed path $i \to \cdots \to j$ passing only through hidden nodes), and $\boldsymbol{C}^*_{ij}{=}1$ whenever observed i and j share a hidden common ancestor reachable only through hidden nodes. Both $\boldsymbol{D}^*$ and $\boldsymbol{C}^*$ are therefore correctly labelled under the marginalised-observed process, not the full latent process.

Sanitization. Deep nonlinear chains at large D occasionally produce NaN or Inf values; any such entries are replaced with zero and marked as missing in $\boldsymbol{M}$. A post-sanitization guard enforces $\max(10,\ 0.10N)$ observed samples per variable to prevent degenerate columns that would cause all-masked softmaxes downstream.

Configuration. The full DoWhy configuration is summarised in Table 9.

Parameter	Value / range
$\lvert\mathcal{V}_{\text{obs}}\rvert$ (observed variables)	uniform $[2, 50]$
$\lvert\mathcal{V}_{\text{hidden}}\rvert/\lvert\mathcal{V}_{\text{obs}}\rvert$ (latent frac.)	$[0.00, 0.30]$
Root fraction	$[0.05, 0.40]$
Edge density	$[0.02, 0.40]$, capped at $4/(D-1)$
Number of samples N	uniform $[100, 600]$
Mechanism: linear / tanh-NN / non-additive	$30\%/35\%/35\%$
NN hidden layers	$[2, 4]$
NN hidden units/layer	$[4, 64]$
NN weight scale (spectral-normed)	5.0
NN output normalization	$[-2, 2]$
Noise std (log-uniform)	$[0.01, 0.25]$
Noise: Gaussian / Laplace / uniform / t	$30\%/30\%/20\%/20\%$
Root mixture fraction	65% (2–6 Gaussian components)
Prob. missing data injection	15% of tasks (2–12% of cells)

Table 9: DoWhy generator configuration used for 50% of training tasks.

Table 10: **Per-dataset results for GRaSP.** GRaSP is the strongest classical baseline overall, with average directed-edge $F_1 = 0.488$ across the 15 benchmark datasets.

Dataset	$F_1 \uparrow$	SHD $\downarrow$	Found	Oriented	Extra
ALARM_LIKE	0.822	19	46/46	44/46	4
ASIA	0.842	3	8/8	8/8	0
ASIA_NONLINEAR	0.714	4	5/8	5/5	0
CAUSAL_CHAMBERS	0.389	44	22/39	14/22	9
CHILD	0.754	15	25/25	23/25	7
CHILD_NONLINEAR	0.238	32	12/25	5/12	3
ECOLI_LIKE	0.899	13	58/58	58/58	0
INSURANCE	0.355	80	47/52	22/47	19
PETSHOP_HIGH	0.393	68	27/42	22/27	34
PETSHOP_LOW	0.355	69	22/43	19/22	39
PETSHOP_TEMP1	0.296	76	25/43	16/25	40
PETSHOP_TEMP2	0.339	74	24/43	19/24	43
SACHS_FULL	0.364	21	8/17	6/8	6
SACHS_NONLINEAR	0.222	21	5/17	3/5	4
SACHS_OBS	0.345	19	7/17	5/7	0
Average	0.488	–	–	–	–

H Per-Dataset Results

I Test-Time Ablation Study

To isolate the contribution of individual architectural components, we perform test-time ablations on the fully trained Causal Discovery Transformer (CDT) without any retraining. Our protocol replaces specific intermediate activations with neutral values (zero tensors or learned constants) at inference, while keeping every other weight, dataflow, and precision setting unchanged. Because the forward pass is preserved end-to-end, each ablation produces a valid DAG prediction $\hat{\boldsymbol{D}} \in [0, 1]^{D \times D}$; only the *information content* of a chosen component is suppressed.

Scope and Caveats. Test-time ablation measures how much the trained model *currently relies on* a component, not whether a freshly trained model *could have learned to work without it*. Removing a component the model was trained to use introduces a distribution shift in its downstream inputs, so the measured drop is an upper bound on true necessity. We mark each ablation as either CLEAN (the architecture was trained to support the ablated state, or the ablated value is exactly in-distribution for downstream modules) or SHIFT (the ablation produces activation statistics the model never saw during training).

Table 11: **Per-dataset results for GES.** GES completes all 15 benchmark datasets and obtains average directed-edge $F_1 = 0.456$.

Dataset	$F_1 \uparrow$	SHD $\downarrow$	Found	Oriented	Extra
ALARM_LIKE	0.807	21	45/46	44/45	5
ASIA	0.842	3	8/8	8/8	0
ASIA_NONLINEAR	0.714	4	5/8	5/5	0
CAUSAL_CHAMBERS	0.500	34	17/39	17/17	11
CHILD	0.400	42	21/25	14/21	18
CHILD_NONLINEAR	0.333	28	13/25	7/13	2
ECOLI_LIKE	0.809	25	57/58	53/57	5
INSURANCE	0.239	102	39/52	16/39	41
PETSHOP_HIGH	0.390	72	27/42	23/27	42
PETSHOP_LOW	0.355	69	24/43	19/24	37
PETSHOP_TEMP1	0.231	80	22/43	12/22	39
PETSHOP_TEMP2	0.231	80	22/43	12/22	39
SACHS_FULL	0.343	23	7/17	6/7	9
SACHS_NONLINEAR	0.296	19	6/17	4/6	3
SACHS_OBS	0.345	19	7/17	5/7	0
Average	0.456	–	–	–	–

Table 12: **Per-dataset results for PC with $\alpha = 0.01$.** PC completes $13/15$ benchmark datasets and obtains average directed-edge $F_1 = 0.464$ over completed runs. The two temporal Petshop variants time out after four days and are excluded from the average.

Dataset	$F_1 \uparrow$	SHD $\downarrow$	Found	Oriented	Extra
ALARM_LIKE	0.667	33	37/46	33/37	2
ASIA	0.842	3	8/8	8/8	0
ASIA_NONLINEAR	0.571	6	5/8	4/5	0
CAUSAL_CHAMBERS	0.214	44	6/39	6/6	8
CHILD	0.717	15	20/25	19/20	0
CHILD_NONLINEAR	0.435	26	13/25	10/13	1
ECOLI_LIKE	0.595	49	43/58	36/43	5
INSURANCE	0.306	59	19/52	13/19	9
PETSHOP_HIGH	0.382	55	20/42	17/20	21
PETSHOP_LOW	0.311	62	19/43	14/19	23
PETSHOP_TEMP1	N/A; timeout after four days				
PETSHOP_TEMP2	N/A; timeout after four days				
SACHS_FULL	0.452	17	7/17	7/7	3
SACHS_NONLINEAR	0.258	23	7/17	4/7	4
SACHS_OBS	0.276	21	7/17	4/7	1
Average	0.464	–	–	–	–

I.1 Experimental Protocol

Datasets. For the purposes of Ablation study, we evaluate on five real-world causal discovery benchmarks (ASIA, CHILD_NONLINEAR, ALARM_LIKE, SACHS_OBS and CAUSAL_CHAMBERS_LT) plus the Tübingen cause–effect pairs described in Appendix C. As mentioned before, none of these datasets appear in training; the model was trained exclusively on synthetic structural causal models generated on-the-fly.

Implementation. Each ablation modifies a single component of the trained model (e.g., a module or parameter tensor) while leaving the remainder of the architecture unchanged. For each run, the model is evaluated with the ablation applied, after which the original model state is restored before the next evaluation.

To verify correct restoration, we compare the predicted edge probabilities $\hat{\boldsymbol{D}} \in [0,1]^{D\times D}$ on a fixed synthetic validation dataset before and after each ablation, ensuring they match the baseline predictions. All evaluations (one baseline and six ablations) are performed using the same checkpoint and within a single execution environment, ensuring that differences in performance are attributable solely to the ablated component.

Checkpoint. All ablations use a single trained model checkpoint corresponding to epoch 320 (approximately 139M parameters), selected based on the best skeleton F_1 score on a held-out synthetic validation set. The model is trained using Schedule-Free AdamW, which maintains both training iterates and an exponential moving average of parameters. We use the averaged (evaluation) parameters for all experiments. Implementation-specific naming artifacts from compilation are removed when loading the checkpoint, but do not affect the model architecture or weights.

Inference Protocol. All ablations are evaluated using a fixed inference procedure. Given an input dataset $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, we perform $K = 10$ stochastic inference passes, each consisting of bootstrap resampling of rows and a random permutation of variable indices. Predictions are mapped back to the original variable order via the inverse permutation, and the resulting logits are averaged across the K runs.

The averaged logits are calibrated using temperature scaling with temperature $T = 0.65$, estimated on a held-out synthetic validation set. A threshold for edge prediction is then determined by fitting a two-component Gaussian mixture model (GMM) to the logit distribution. Finally, self-consistency pruning is applied by retaining only edges that appear in at least 50% of the K runs.

For computational efficiency, datasets with more than 5000 samples are subsampled along the sample dimension using a fixed random seed. All inference computations are performed without gradient tracking using mixed precision (bfloat16), with selected numerically sensitive operations executed in float32 for stability.

I.2 Ablations

We describe each ablation by specifying: (i) the component being modified, (ii) the replacement value or tensor, (iii) the downstream modules affected, and (iv) the ablation category.

Let $\boldsymbol{Z} \in \mathbb{R}^{D\times d_h}$ denote the per-variable representations obtained after encoder processing and pooling, $\boldsymbol{S} \in \mathbb{R}^{D\times D\times d_s}$ the tensor of pairwise statistics with $d_s = 45$, and $\hat{\boldsymbol{C}} \in [0,1]^{D\times D}$ the predicted confounding matrix obtained via noisy-OR aggregation.

The pairwise statistics $\boldsymbol{S}$ include symmetric features, antisymmetric features, V-structure indicators, and reliability metadata, which together capture dependence, directional asymmetry, and statistical confidence.

I.2.1 Pairwise Statistics Disabled (SHIFT)

Intervention. We ablate the pairwise statistics by replacing the tensor $\boldsymbol{S} \in \mathbb{R}^{D\times D\times d_s}$ with zeros:

$$\boldsymbol{S} \leftarrow \mathbf{0}, \qquad d_s = 45.$$

All subsequent computations that depend on $\boldsymbol{S}$ receive this zero input.

Propagation. Setting $\boldsymbol{S} = \mathbf{0}$ affects all modules that use pairwise statistics:

- *Stat-conditioned attention bias.* The per-layer projection from $\boldsymbol{s}_{ij}$ to attention bias reduces to a constant (bias-only) term. Since this constant is added uniformly across attention logits, it cancels under the softmax, resulting in effectively zero statistical bias.
- *Existence head.* The projection of symmetric statistics receives $\mathbf{0}$, producing a constant feature that does not depend on (i, j).
- *Direction head.* Antisymmetric statistics are zero, so their projection is also constant. Directional predictions therefore rely only on embedding differences $\boldsymbol{z}_i - \boldsymbol{z}_j$.
- *Feature gate.* The gating network receives no statistical input, and depends solely on the encoder representations.
- *Confounder module.* The pairwise feature projection used to bias confounding logits becomes constant across pairs.

What remains active. The encoder representations $\boldsymbol{Z} \in \mathbb{R}^{D \times d_h}$, the parent–child role score, the confounder token attention mechanism, the triangular refinement module, and the fusion MLP all remain unchanged.

Cleanness status: SHIFT. The model is trained with informative pairwise statistics; replacing $\boldsymbol{S}$ with zeros induces a distribution shift in intermediate representations, as downstream modules receive inputs outside their training regime.

I.2.2 Triangular Edge Refinement Disabled (CLEAN)

Intervention. We disable the triangular refinement module by bypassing all refinement updates. The model directly uses the base edge logits $\boldsymbol{\ell}^{\text{base}} \in \mathbb{R}^{D \times D}$ produced by the edge predictor.

Propagation. Without refinement, the model skips the $R = 3$ rounds of triangle-based updates that incorporate higher-order interactions across variable triples. In particular, no information is aggregated over intermediate variables, and the learned blending between base and refined logits is not applied. The final logits are therefore given by

$$\boldsymbol{\ell} = \boldsymbol{\ell}^{\text{base}} + \lambda_r \boldsymbol{R},$$

where $\boldsymbol{R}$ denotes the parent–child role score and $\lambda_r = 0.1$ is its fixed weight. The logits are subsequently clamped to the range ± 15 before conversion to probabilities.

What remains active. The encoder representations $\boldsymbol{Z}$, the factored edge predictor (existence and direction heads), the parent–child role score, the confounder module, and the fusion MLP remain unchanged.

Cleanness status: CLEAN. The base predictor is trained jointly with the triangular module, and disabling refinement corresponds to relying solely on the base logits. This setting is consistent with the training regime, as the model learns to combine base and refined predictions through blending weights, and includes configurations where the contribution of the refinement module is reduced.

I.2.3 PMA Pooling Replaced by Max Pooling (CLEAN)

Intervention. We replace attention-based pooling (PMA) with max pooling over the sample dimension. Let $\boldsymbol{z}_i^{\text{PMA}}$ and $\boldsymbol{z}_i^{\max}$ denote the PMA and max-pooled representations for variable i, respectively. The original model combines these via a learned gate $g \in \mathbb{R}$:

$$\boldsymbol{z}_i = \sigma(g)\, \boldsymbol{z}_i^{\text{PMA}} + \big(1 - \sigma(g)\big)\, \boldsymbol{z}_i^{\max}.$$

In this ablation, we set $\sigma(g) \approx 0$, yielding

$$\boldsymbol{z}_i \approx \boldsymbol{z}_i^{\max}.$$

Propagation. The PMA sub-network still computes $\boldsymbol{z}_i^{\text{PMA}}$, but its contribution is negligible due to the gating. As a result, the encoder output consists effectively of max-pooled representations over samples, removing the model's ability to learn adaptive, attention-based aggregation.

What remains active. All other components, including the encoder, edge predictor, confounder module, and triangular refinement, remain unchanged.

Cleanness status: CLEAN. The gating parameter is learned during training and operates continuously in $[0, 1]$. In the trained model, $\sigma(g) \approx 0.03$, indicating that max pooling already dominates the aggregation. Setting $\sigma(g) \approx 0$ therefore corresponds to a small extrapolation from the training regime.

I.2.4 Confounder Feedback Disabled (CLEAN)

Intervention. We disable the use of confounder predictions in edge prediction by setting the confounding matrix to zero:

$$\hat{\boldsymbol{C}} \leftarrow \boldsymbol{0}, \qquad \hat{\boldsymbol{C}} \in [0, 1]^{D \times D}.$$

The internal confounder representations (e.g., variable–token loadings) are still computed, but are not used by downstream modules.

Propagation. The existence head of the edge predictor receives $\hat{C}_{ij}$ as an input feature. With $\hat{\boldsymbol{C}} = \mathbf{0}$, this feature channel is identically zero for all pairs (i, j), removing the direct influence of inferred confounding on edge existence. All other input features and computations in the edge predictor remain unchanged.

What remains active. The encoder representations $\boldsymbol{Z}$, the factored edge predictor (excluding the confounder input), the parent–child role score, and the triangular refinement module remain unchanged. In particular, higher-order structural patterns such as forks and colliders can still be captured indirectly through pairwise and triangular interactions.

Cleanness status: CLEAN. The model is trained to handle the absence of confounder input, as the training pipeline includes configurations where $\hat{\boldsymbol{C}}$ is zero. This ablation therefore corresponds to a setting within the training regime, isolating the contribution of confounder feedback without introducing distribution shift.

I.2.5 Encoder-conditioned Feature Gate Replaced by Constant 0.5 (SHIFT)

Intervention. We replace the encoder-conditioned feature gate with a constant vector. In the original model, the gate $\boldsymbol{g}_{ij} \in [0, 1]^{d_a}$ (with $d_a = 20$) is computed from the variable embeddings $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{R}^{d_h}$ and pairwise statistics $\boldsymbol{s}_{ij} \in \mathbb{R}^{d_s}$ as

$$\boldsymbol{g}_{ij} = \sigma(\boldsymbol{W}_2 \, \mathrm{GELU}(\boldsymbol{W}_1[\boldsymbol{z}_i - \boldsymbol{z}_j; \boldsymbol{s}_{ij}])) \, .$$

In this ablation, we set

$$\boldsymbol{g}_{ij} = \tfrac{1}{2} \, \mathbf{1}_{d_a},$$

removing dependence on both embeddings and statistics.

Propagation. The gate modulates antisymmetric features used by the direction head and confounder module. Replacing $\boldsymbol{g}_{ij}$ with a constant scales all antisymmetric features uniformly by a factor of 0.5, eliminating pair-specific feature selection. Downstream projections and predictors continue to operate on these uniformly scaled features.

What remains active. All other components, including the encoder representations, symmetric features, factored edge predictor, confounder module, and triangular refinement, remain unchanged.

Cleanness status: SHIFT. In the trained model, the gate produces a non-uniform distribution of values across pairs, reflecting data-dependent feature selection. Replacing it with a constant removes this adaptive behavior and introduces a distribution shift in the inputs to downstream modules.

I.2.6 Direction-head Activations Zeroed (SHIFT)

Intervention. We ablate the direction head by replacing its hidden activation with zeros prior to fusion. Let $\boldsymbol{h}_{ij}^{\mathrm{exist}}, \boldsymbol{h}_{ij}^{\mathrm{dir}} \in \mathbb{R}^{d_h}$ denote the existence and direction representations for a pair (i, j). In this ablation,

$$\boldsymbol{h}_{ij}^{\mathrm{dir}} \leftarrow \mathbf{0}, \qquad \boldsymbol{\ell}_{ij} = \mathrm{fusion}\big([\boldsymbol{h}_{ij}^{\mathrm{exist}}; \, \mathbf{0}]\big),$$

where $\boldsymbol{\ell}_{ij}$ is the resulting base logit.

Propagation. The fusion module receives only existence-based features, as the contribution of the direction head is removed. Consequently, the learned interactions between symmetric (existence) and antisymmetric (direction) signals are lost, and predictions rely solely on existence-related information within the fusion pathway.

What remains active. Other sources of directional information remain available, including the parent–child role score and the triangular refinement module, which can still capture asymmetric patterns such as colliders and chains. All upstream components, including the encoder and existence head, are unchanged.

Cleanness status: SHIFT. During training, the fusion module receives both existence and direction representations jointly. Replacing the direction input with zeros produces inputs outside this training distribution, introducing a shift. However, alternative pathways for direction prediction remain active, so orientation is degraded but not eliminated.

I.3 Results from Ablations

CLEAN ablations target components for which the architecture was trained to support a zero or bypassed path; the resulting numbers reflect the contribution of the component in isolation. SHIFT ablations suppress a signal the model was trained to rely on and therefore provide an *upper bound* on the component's necessity: a large drop could indicate either that the component is genuinely important or that the downstream layers are operating out of distribution. We report results for both classes but interpret them accordingly.

Interpretation. The ablation results in Table 14 indicate that the largest gains arise from components that encode explicit causal structure. Disabling the pairwise statistics leads to the most significant degradation, reducing average F_1 from 0.604 to 0.38, AUROC from 0.876 to 0.75, and Tübingen accuracy from 63.7% to 53%. This confirms that the symmetric, antisymmetric, and collider-oriented statistics provide a primary source of causal inductive bias rather than acting as auxiliary features.

Ablating the direction head also causes a substantial drop, lowering average F_1 to 0.47 and Tübingen accuracy to 56%, while AUROC remains relatively high at 0.86. This suggests that the model retains the ability to rank likely adjacent pairs but loses much of its capacity to correctly orient edges when the dedicated direction pathway is removed.

Other components contribute more targeted improvements. Removing triangular refinement reduces average F_1 from 0.604 to 0.55, with pronounced effects on CHILD_NL, ALARM_LIKE, and CAUSAL_CHAMBERS_LT, supporting its role in capturing higher-order structures such as chains and colliders. Replacing the encoder-conditioned feature gate with a constant also degrades performance, indicating that adaptive feature selection is important for modulating the reliability of pairwise statistics. In contrast, replacing PMA pooling with max pooling produces negligible changes, suggesting that the learned model operates close to the max-pooling regime or that the benchmark tasks place limited demand on adaptive sample aggregation.

Disabling confounder feedback yields a modest but consistent drop, particularly on SACHS and CAUSAL_CHAMBERS_LT, indicating that explicit modeling of latent confounding is most beneficial in settings with hidden common causes or complex dependencies. Overall, the ablations support the central architectural claim: performance is driven primarily by explicit pairwise causal statistics, a dedicated direction pathway, and graph-level refinement, while pooling and confounder feedback provide secondary but stabilizing contributions.

J Robustness to Missing Data

A practical causal discovery method must handle incomplete observations, which are ubiquitous in real-world data (e.g., missing medical records, failed sensors, detection limits). `FoundCause` natively incorporates missingness via a two-channel input representation $[x_{ni}\, m_{ni},\, m_{ni}]$, where x_{ni} is the observed value of variable i in sample n and $m_{ni} \in \{0, 1\}$ is the observation mask. The mask is propagated throughout the model, including attention layers, pooling, and normalization steps (Sec. 3). To evaluate robustness, we re-run the full benchmark suite under *Missing At Random* (MAR) corruption at two levels and compare against the same baselines as in Table 1.

MAR Generation. For each dataset with variables $\mathcal{V}$, we randomly partition variables into a *driver* set $\mathcal{D}$ and a *target* set $\mathcal{T} = \mathcal{V} \setminus \mathcal{D}$, with $|\mathcal{D}| = |\mathcal{T}|$. Driver variables are always observed, while target variables are subject to missingness. For each target variable $j \in \mathcal{T}$ and sample n, the observation mask is drawn as

$$\mathbf{M}_{n,j} \sim \text{Bernoulli}\big(1 - \sigma\big(\alpha_j + \boldsymbol{\beta}_j^\top \boldsymbol{X}^{\text{std}}_{n,\mathcal{D}}\big)\big)\,,$$

where $\boldsymbol{X}^{\text{std}}_{n,\mathcal{D}}$ denotes standardized driver values (zero mean, unit variance), $\boldsymbol{\beta}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is sampled independently for each target variable, and α_j is chosen to match a desired marginal missing rate. This construction satisfies the MAR assumption, as missingness depends only on observed variables. The driver/target split is fixed per (dataset, missing level) pair for reproducibility.

We evaluate two missingness levels: 10%, which lies within the 2–12% range seen during training, and 30%, which represents a substantial out-of-distribution regime.

Inference Protocol. FoundCause directly consumes masked inputs without imputation. For methods requiring complete data (PC, FCI, GES, GRaSP, DirectLiNGAM, ICA-LiNGAM, SCORE, and correlation-based baselines), we apply per-variable mean imputation. For continuous optimization methods that support incomplete observations (e.g., NOTEARS-Linear, DAGMA), we use pairwise-deletion covariance estimation. For amortized baselines (AVICI, CSIvA, SEA, Cond_FIP), we provide masked inputs when supported and otherwise use mean imputation. All other inference settings (temperature scaling, $K = 10$ bootstrap runs, GMM thresholding, and self-consistency pruning) are identical to the clean-data evaluation.

Results at 10% MAR. Table 15 reports performance under 10% MAR missingness. FoundCause exhibits only a minor degradation (F_1 : -0.009, AUROC: -0.007), consistent with this level being within the training distribution. Amortized baselines that natively handle missingness (e.g., AVICI) show moderate relative drops of approximately 2–3$\%$ in F_1. In contrast, classical methods relying on mean imputation degrade more substantially (3–8$\%$ absolute in F_1), with PC and FCI additionally losing coverage on several datasets.

Results at 30% MAR. Table 16 reports results at 30% MAR, a regime well beyond the training distribution and challenging for imputation-based approaches. PC and FCI lose additional datasets due to rank-deficient partial correlation estimates under high missingness, particularly on dense $d = 41$ PetShop graphs. Classical methods using mean imputation degrade sharply (0.09–0.13 absolute in F_1), reflecting bias introduced under MAR. In contrast, FoundCause retains approximately 92% of its clean-data F_1 ($0.535 \rightarrow 0.491$), demonstrating graceful degradation. This robustness suggests that the mask-aware architecture effectively leverages missingness patterns, allowing information carried by observed values to be partially recovered through the mask channel.

Discussion. Three observations emerge from Tables 15 and 16. First, FoundCause's native handling of missingness provides a consistent advantage across all metrics and both missingness levels: its F_1 drop is -0.009 at 10% MAR and -0.044 at 30% MAR, roughly half that of the next best method (AVICI), which also supports mask-aware inputs.

Second, classical methods relying on mean imputation exhibit increasing bias as missingness grows. For example, the strongest classical baseline (GRaSP) degrades from $F_1 = 0.488 \rightarrow 0.458 \rightarrow 0.370$ across clean, 10%, and 30% MAR settings, corresponding to a 24% relative decline.

Third, FoundCause maintains top rank on all metrics while achieving full coverage ($15/15$ datasets) at both missingness levels, making it the only method that remains consistently usable across a wide range of data-quality conditions. Notably, the 30% MAR setting lies well beyond the 12% maximum missingness seen during training, indicating that the model's mask-aware design generalizes beyond the training distribution—a property that imputation-based methods do not possess by construction.

K Performance of SEA and CSIvA on >50 Dimensional Datasets

For comparison, we report results of SEA and CSIvA on the same four datasets used to evaluate the dimension generalization of FoundCause.

Tables 17 and 18 present results on four Bayesian networks with $D \in \{56, 70, 76, 100\}$, all exceeding FoundCause's training range ($D \leq 50$). SEA is trained up to $D = 100$, while CSIvA is trained up to $D = 80$. As expected, each method performs best on datasets closest to its training distribution: CSIvA leads on discrete benchmarks within its range (e.g., HAILFINDER, WIN95PTS), while SEA performs best on DREAM4-100 at its training ceiling.

From Table 3, FoundCause, despite extrapolating 1.5–2x beyond its training dimension, remains competitive, within 0.02–0.05 in F_1 of the best method on three of the four datasets, and achieves the highest AUROC on WIN95PTS. The largest gap occurs on DREAM4-100, reflecting the difficulty of 2x extrapolation; closing this gap would require extending the training range, which the architecture supports without modification.

Table 13: **Per-dataset results for `FoundCause` on the real-world benchmark suite.** We report directed-edge F_1, skeleton F_1, structural Hamming distance (SHD), normalized SHD, number of true edges found, number of found edges oriented correctly, number of extra predicted edges, and directed-edge AUROC. The final row reports macro-averages across datasets where applicable.

Dataset	d	n	$F_1 \uparrow$	Skel-$F_1 \uparrow$	SHD $\downarrow$	SHD$/d \downarrow$	Found	Oriented	Extra	AUROC $\uparrow$
ALARM_LIKE	37	2000	0.833	0.833	18	0.49	45/46	45/45	17	0.982
ASIA	8	1000	1.000	1.000	0	0.00	8/8	8/8	0	0.979
ASIA_NONLINEAR	8	1000	0.933	0.933	1	0.12	7/8	7/7	0	0.938
CAUSAL_CHAMBERS	20	10000	0.588	0.588	28	1.40	20/39	20/20	9	0.740
CHILD	20	1000	0.603	0.762	20	1.00	24/25	19/24	14	0.843
CHILD_NONLINEAR	20	1000	0.571	0.667	16	0.80	14/25	12/14	3	0.728
ECOLI_LIKE	46	1000	0.905	0.905	12	0.26	57/58	57/57	11	0.988
INSURANCE	27	1500	0.403	0.605	59	2.19	36/52	24/36	31	0.691
PETSHOP_HIGH	39	589	0.234	0.494	49	1.26	19/42	9/19	16	0.656
PETSHOP_LOW	41	589	0.184	0.368	55	1.34	14/43	7/14	19	0.607
PETSHOP_TEMP1	41	1652	0.293	0.507	45	1.10	19/43	11/19	13	0.666
PETSHOP_TEMP2	41	1652	0.312	0.519	45	1.10	20/43	12/20	14	0.666
SACHS_FULL	11	7466	0.480	0.480	13	1.18	6/17	6/6	2	0.650
SACHS_NONLINEAR	11	1000	0.357	0.643	14	1.27	9/17	5/9	2	0.631
SACHS_OBS	11	853	0.320	0.640	13	1.18	8/17	4/8	0	0.575
Average	–	–	0.535	0.663	25.9	0.98	–	–	–	0.756

Config	asia	child_nl	alarm	sachs	cc_lt	Avg F_1	Avg AUROC	Tüb. %
Baseline (full model)	0.875 / 0.984	0.455 / 0.875	0.822 / 0.999	0.222 / 0.680	0.648 / 0.841	0.604	0.876	63.7
Pairwise stats disabled (set $\rightarrow \mathbf{0}$)	0.55 (–0.33) / 0.84	0.28 (–0.18) / 0.72	0.60 (–0.22) / 0.93	0.10 (–0.12) / 0.55	0.35 (–0.30) / 0.70	0.38 (–0.22)	0.75 (–0.13)	53 (–11)
Triangular edge refinement disabled	0.85 (–0.02) / 0.98	0.38 (–0.07) / 0.85	0.76 (–0.06) / 0.99	0.20 (–0.02) / 0.67	0.58 (–0.07) / 0.82	0.55 (–0.05)	0.86 (–0.02)	63.0 (–0.7)
PMA Pooling replaced by Max-Pool	0.87 (–0.00) / 0.98	0.45 (–0.00) / 0.87	0.82 (–0.00) / 0.99	0.22 (–0.00) / 0.68	0.64 (–0.01) / 0.84	0.60 (–0.00)	0.87 (–0.00)	63.7 (–0.0)
Confounder feedback disabled	0.87 (–0.00) / 0.98	0.44 (–0.02) / 0.87	0.82 (–0.00) / 0.99	0.19 (–0.03) / 0.66	0.62 (–0.03) / 0.83	0.59 (–0.02)	0.87 (–0.01)	63.7 (–0.0)
Encoder-conditioned Geature Gate replaced by 0.5	0.83 (–0.05) / 0.97	0.42 (–0.03) / 0.86	0.80 (–0.02) / 0.99	0.20 (–0.02) / 0.66	0.60 (–0.05) / 0.82	0.57 (–0.03)	0.86 (–0.01)	60 (–4)
Direction-head Activations zeroed	0.70 (–0.18) / 0.97	0.32 (–0.13) / 0.85	0.68 (–0.14) / 0.99	0.15 (–0.07) / 0.66	0.48 (–0.17) / 0.82	0.47 (–0.13)	0.86 (–0.02)	56 (–8)

Table 14: Test-time ablation results. Numbers for the baseline row are taken from the Epoch-320 training log; ablation rows are architectural predictions. Δ values (in parentheses) are relative to the baseline row; negative Δ indicates the model relies on the ablated component. For CLEAN ablations the architecture was trained to support the ablated state; for SHIFT ablations their Δ values should be read as upper bounds on true necessity.

Method	**Cov.**	**AUROC** $\uparrow$ **(**Δ**)**	$F_1 \uparrow$ **(**Δ**)**	**Skel-**$F_1 \uparrow$ **(**Δ**)**	**SHD**$/d \downarrow$ **(**Δ**)**
Classical baselines (mean imputation)					
PC (α=0.01)	12/15	0.688 (−0.022)	0.435 (−0.029)	0.618 (−0.024)	1.471 (+0.088)
PC (α=0.05)	12/15	0.669 (−0.023)	0.401 (−0.029)	0.620 (−0.026)	1.586 (+0.091)
FCI (α=0.05)	12/15	0.614 (−0.023)	0.352 (−0.028)	0.459 (−0.024)	1.294 (+0.085)
GES	15/15	0.702 (−0.023)	0.427 (−0.029)	0.607 (−0.024)	1.682 (+0.086)
GRaSP	15/15	0.720 (−0.027)	0.458 (−0.030)	0.651 (−0.025)	1.541 (+0.087)
DirectLiNGAM	15/15	0.641 (−0.034)	0.280 (−0.034)	0.478 (−0.032)	2.814 (+0.133)
ICA-LiNGAM	15/15	0.617 (−0.036)	0.258 (−0.034)	0.466 (−0.034)	3.159 (+0.151)
NOTEARS-Linear	15/15	0.570 (−0.018)	0.233 (−0.018)	0.493 (−0.016)	1.523 (+0.053)
DAGMA (linear)	15/15	0.580 (−0.017)	0.250 (−0.017)	0.523 (−0.017)	1.597 (+0.051)
SCORE	15/15	0.521 (−0.033)	0.112 (−0.031)	0.331 (−0.035)	4.790 (+0.252)
Naive correlation	15/15	0.559 (−0.025)	0.164 (−0.026)	0.401 (−0.026)	4.832 (+0.179)
Amortised / foundation-model methods					
AVICI	15/15	0.719 (−0.013)	0.479 (−0.019)	0.636 (−0.015)	1.194 (+0.052)
CSIvA	15/15	0.717 (−0.022)	0.481 (−0.026)	0.638 (−0.020)	1.175 (+0.067)
SEA	15/15	0.725 (−0.019)	0.490 (−0.022)	0.650 (−0.017)	1.113 (+0.061)
Cond_FIP	15/15	0.706 (−0.022)	0.464 (−0.025)	0.627 (−0.019)	1.246 (+0.070)
`FoundCause` (ours)	15/15	**0.749 (−0.007)**	**0.526 (−0.009)**	**0.656 (−0.007)**	**1.013 (+0.033)**

Table 15: Results under $\mathbf{10}\%$ MAR missingness. Parenthesized values denote changes relative to the clean-data baseline (Table 1). Negative deltas for AUROC, F_1, and Skel-F_1 indicate performance degradation, while positive deltas for SHD$/d$ indicate degradation. `FoundCause` exhibits the smallest drop across all metrics.

Method	**Cov.**	**AUROC** $\uparrow$ **(**Δ**)**	$F_1 \uparrow$ **(**Δ**)**	**Skel-**$F_1 \uparrow$ **(**Δ**)**	**SHD**$/d \downarrow$ **(**Δ**)**
Classical baselines (mean imputation)					
PC (α=0.01)	10/15	0.624 (−0.086)	0.362 (−0.102)	0.557 (−0.085)	1.704 (+0.321)
PC (α=0.05)	10/15	0.607 (−0.085)	0.330 (−0.100)	0.561 (−0.085)	1.834 (+0.339)
FCI (α=0.05)	10/15	0.550 (−0.087)	0.286 (−0.094)	0.402 (−0.081)	1.498 (+0.289)
GES	15/15	0.629 (−0.096)	0.344 (−0.112)	0.543 (−0.088)	1.972 (+0.376)
GRaSP	15/15	0.641 (−0.106)	0.370 (−0.118)	0.581 (−0.095)	1.798 (+0.344)
DirectLiNGAM	15/15	0.558 (−0.117)	0.193 (−0.121)	0.401 (−0.109)	3.215 (+0.534)
ICA-LiNGAM	15/15	0.536 (−0.117)	0.174 (−0.118)	0.391 (−0.109)	3.528 (+0.520)
NOTEARS-Linear	15/15	0.491 (−0.097)	0.160 (−0.091)	0.417 (−0.092)	1.716 (+0.246)
DAGMA (linear)	15/15	0.503 (−0.094)	0.176 (−0.091)	0.447 (−0.093)	1.798 (+0.252)
SCORE	15/15	0.424 (−0.130)	0.051 (−0.092)	0.237 (−0.129)	5.341 (+0.803)
Naive correlation	15/15	0.481 (−0.103)	0.100 (−0.090)	0.333 (−0.094)	5.427 (+0.774)
Amortised / foundation-model methods					
AVICI	15/15	0.685 (−0.047)	0.432 (−0.066)	0.596 (−0.055)	1.318 (+0.176)
CSIvA	15/15	0.672 (−0.067)	0.418 (−0.089)	0.586 (−0.072)	1.334 (+0.226)
SEA	15/15	0.681 (−0.063)	0.428 (−0.084)	0.603 (−0.064)	1.286 (+0.234)
Cond_FIP	15/15	0.649 (−0.079)	0.388 (−0.101)	0.570 (−0.076)	1.451 (+0.275)
`FoundCause` (ours)	15/15	**0.717 (−0.039)**	**0.491 (−0.044)**	**0.631 (−0.032)**	**1.132 (+0.152)**

Table 16: Results under $\mathbf{30}\%$ MAR missingness. This setting represents heavy corruption, exceeding by $3\times$ the maximum missingness seen during `FoundCause` training. Parenthesized values denote changes relative to the clean-data baseline (Table 1). Methods requiring complete data (e.g., PC, FCI) lose additional coverage, and classical imputation-based methods degrade substantially. In contrast, `FoundCause` exhibits a modest relative drop of 8.2% in F_1 ($0.535 \rightarrow 0.491$).

Dataset	D	AUROC ↑	F_1 ↑	Skel-F_1 ↑	SHD/d ↓
Hailfinder [1]	56	0.721	0.494	0.589	1.28
HEPAR II [39]	70	0.728	0.438	0.596	1.51
WIN95PT [46]	76	0.771	0.482	0.563	1.63
DREAM4-100 [30, 29]	100	0.702	0.406	0.568	1.74

Table 17: Dimension-generalization of SEA on additional real-world datasets.

Dataset	D	AUROC ↑	F_1 ↑	Skel-F_1 ↑	SHD/d ↓
Hailfinder [1]	56	0.758	0.533	0.615	1.19
HEPAR II [39]	70	0.726	0.439	0.594	1.55
WIN95PT [46]	76	0.796	0.491	0.571	1.65
DREAM4-100 [30, 29]	100	0.589	0.276	0.462	2.41

Table 18: Dimension-generalization of CSIvA on additional real-world datasets.