

A Supplementary Material

In this supplementary material, we provide the information for understanding SciNO with detailed explanations. The supplementary material is organized as follows: Section A.1 introduces related works in ordering based causal discovery, score matching methods, and LLM-assisted causal discovery. Section A.2 provides theoretical derivations and proofs for preliminaries and the proposed method. Section A.3 explains a detailed description of the architecture of SciNO with layer-by-layer formulations on how to implement it. Section A.4 describes the experimental setup, datasets, evaluation metrics and implementation details of empirical results in Sections 4.1 and 4.2. Section A.5 presents experimental details in Section 4.3 and additional results on controlling LLMs for causal ordering.

A.1 Related Works

A.1.1 Ordering-based Causal Discovery

Ordering-based causal discovery methods are typically categorized into score matching based and non-score matching based approaches. SCORE [31], DiffAN [33], and CaPS [42] are score matching based methods that estimate the Hessian of the log density to identify leaf nodes and construct causal order. Non-score matching methods such as CAM [4] and GSPo[3] recover the causal order using functional assumptions and conditional independence tests, respectively. In this work, we focus on the score matching based approaches [31, 33, 42], which estimate gradients of the log-density for causal ordering.

A.1.2 Score Matching and Score-based Generative Models

A common approach to estimating the score function of a data distribution is to use a gradient estimator based on the Stein identity [21]. Stein kernel-based methods allow score estimation without an explicit density and have been used in SCORE [31] and CaPS [42]. However, these methods require computing the inverse of a kernel matrix, leading to high computational and memory costs. In recent work, Diffusion models [12, 37] have emerged as a key method for score estimation in generative modeling. However, the commonly used Denoising Diffusion Model [12, 37] is defined over finite-dimensional spaces and thus struggle to handle functional data such as derivatives. Functional diffusion models [23, 24] address this issue by defining the diffusion process over a Hilbert space and learning it directly via score-based modeling. [23, 24] utilize the Neural Operators [15, 22] to learn mappings between functions, allowing the model to capture both the functional form and its derivatives.

A.1.3 Causal Discovery with Large Language Models

Recent studies leverage domain knowledge inferred from LLM as priors or constraints to enhance data-driven causal discovery [2, 7, 18, 19, 25, 39]. Most methods adopt pairwise prompts to construct causal graphs via repeated queries on variable pairs [1, 7, 14, 18, 25]. To address inability of LLMs to distinguish between direct and indirect effects, [39] introduces a triplet prompt that queries relationships among three variables simultaneously, deriving causal order rather than graph structures. Existing methods treat LLMs as a black-box and directly convert generated tokens into binary decisions. This obscures uncertainty and limits the scientific explanatory power of causal inference. [7] partially addresses this by extracting pairwise edge-direction probabilities from the LLM and combining them with mutual information, but this approach remains limited, as it does not directly reflect causal relationships, limiting its explanatory power.

A.2 Proof of Theorems

A.2.1 Derivation of the Hessian diagonal approximation

In this section, we derive approximations (5) and (6), which are suggested by DiffAN [33], for the sake of completeness. Under the ANM assumption (2), the score function $\mathbf{S}(\mathbf{x})$ can be derived as:

$$\mathbf{S}_j(\mathbf{x}) = \partial_{x_j} \log P(\mathbf{x}) = \partial_{x_j} \log \prod_{i=1}^D P(x_i | \text{Pa}(x_i)) = \partial_{x_j} \sum_{i=1}^D \log P(x_i | \text{Pa}(x_i)), \quad j \in \mathcal{V}, \quad (13)$$

792 where D denotes the number of nodes $|\mathcal{V}|$. Let $\epsilon_i = x_i - f_i$. By the change of variable, it holds that

$$\partial_{x_j} \sum_{i=1}^D \log P(x_i | \text{Pa}(x_i)) = \partial_{x_j} \sum_{i=1}^D \log p_i(x_i - f_i) \quad (14)$$

$$= \frac{\partial \log p_j(x_j - f_j)}{\partial x_j} - \sum_{i \in \text{Ch}(x_j)} \frac{\partial f_i}{\partial x_j} \frac{\partial \log p_i(x_i - f_i)}{\partial \epsilon_i}. \quad (15)$$

793 When a leaf node x_l is marginalized out, the resulting score $\mathbf{S}_j(\mathbf{x}_{-l})$ for $x_j \in \text{Pa}(x_l)$ becomes:

$$\mathbf{S}_j(\mathbf{x}_{-l}) = \mathbf{S}_j(\mathbf{x}) + \frac{\partial f_l}{\partial x_j} \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l}. \quad (16)$$

794 Let $\delta_{j,l}$ denote the residue term added to the score of x_j upon marginalizing out x_l :

$$\delta_{j,l} := \frac{\partial f_l}{\partial x_j} \cdot \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l}. \quad (17)$$

795 Note that the residue term $\delta_{j,l}$ vanishes if $x_j \notin \text{Pa}(x_l)$. For a leaf node x_l , it holds that:

$$\mathbf{S}_l(\mathbf{x}) = \frac{\partial \log p_l(x_l - f_l)}{\partial x_l} = \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l} \cdot \frac{\partial (x_l - f_l)}{\partial x_l} = \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l}. \quad (18)$$

796 The partial derivatives of the score $\mathbf{S}_l(\mathbf{x})$ with respect to x_j for any $j \in \mathcal{V}$ are given by:

$$\partial_{x_j} \mathbf{S}_l(\mathbf{x}) = \frac{\partial^2 \log p_l(x_l - f_l)}{\partial \epsilon_l^2} \cdot \frac{\partial f_l}{\partial x_j}. \quad (19)$$

797 In particular, when $j = l$, this reduces to:

$$\partial_{x_l} \mathbf{S}_l(\mathbf{x}) = \frac{\partial^2 \log p_l(x_l - f_l)}{\partial \epsilon_l^2}. \quad (20)$$

798 Thus, $\delta_{j,l}$ can be rewritten as:

$$\delta_{j,l} = \frac{\partial f_l}{\partial x_j} \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l} \quad (21)$$

$$= \frac{\partial^2 \log p_l(x_l - f_l)}{\partial \epsilon_l^2} \cdot \frac{\partial f_l}{\partial x_j} \cdot \frac{\partial \log p_l(x_l - f_l)}{\partial \epsilon_l} \bigg/ \frac{\partial^2 \log p_l(x_l - f_l)}{\partial \epsilon_l^2} \quad (22)$$

$$= \partial_{x_j} \mathbf{S}_l(\mathbf{x}) \cdot \frac{\mathbf{S}_l(\mathbf{x})}{\partial_{x_l} \mathbf{S}_l(\mathbf{x})} \quad (23)$$

799 Let $\pi_k \subset \pi$ denote the set of selected leaf nodes up to step k , and $-\pi_k = \mathcal{V} \setminus \pi_k$ denote the remaining
800 ones. The decoupled score on the remaining variables $\mathbf{S}(\mathbf{x}_{-\pi_k}) \in \mathbb{R}^{D-|\pi_k|}$ are approximated by:

$$\mathbf{S}_j(\mathbf{x}_{-\pi_k}) = \mathbf{S}_j(\mathbf{x}) + \sum_{l \in \pi_k} \left(\partial_{x_j} \mathbf{S}_l(\mathbf{x}) \cdot \frac{\mathbf{S}_l(\mathbf{x})}{\partial_{x_l} \mathbf{S}_l(\mathbf{x})} \right), \quad j \in \mathcal{V} \setminus \pi_k. \quad (24)$$

801 Hence we obtain (5). Finally, the Hessian diagonal $\mathcal{D}(\mathbf{x}_{-\pi_k}) = (\partial_{x_j} \mathbf{S}_j(\mathbf{x}_{-\pi_k}) : j \in \mathcal{V} \setminus \pi_k)$ is:

$$\mathcal{D}_j(\mathbf{x}_{-\pi_k}) = \partial_{x_j} \mathbf{S}_j(\mathbf{x}) + \sum_{l \in \pi_k} \partial_{x_j} \left(\partial_{x_j} \mathbf{S}_l(\mathbf{x}) \cdot \frac{\mathbf{S}_l(\mathbf{x})}{\partial_{x_l} \mathbf{S}_l(\mathbf{x})} \right) \quad (25)$$

$$\approx \partial_{x_j} \hat{\mathbf{S}}_j^\theta(t, \mathbf{x}) + \sum_{l \in \pi_k} \partial_{x_j} \left(\partial_{x_j} \hat{\mathbf{S}}_l^\theta(t, \mathbf{x}) \cdot \frac{\hat{\mathbf{S}}_l^\theta(t, \mathbf{x})}{\partial_{x_l} \hat{\mathbf{S}}_l^\theta(t, \mathbf{x})} \right), \quad j \in \mathcal{V} \setminus \pi_k. \quad (26)$$

802 Thus, we obtain the approximation (6).

803 **Remark A.1.** Note that (25) requires the second-order derivatives of $\mathbf{S}(\mathbf{x})$ and so does approximation
804 (6) replacing the score function by the score model $\hat{\mathbf{S}}^\theta(t, \mathbf{x})$.

A.2.2 Approximation Power of Neural Operators

Score-based Generative Modeling in Function Spaces For readers who are not familiar with functional diffusion modeling, we briefly introduce a time-reversal theory of score-based generative modeling in Hilbert spaces [24].

Consider the following stochastic evolution equations in the Hilbert space \mathcal{H} :

$$d\mathbf{X}_t = \mathbf{B}_t(\mathbf{X}_t)dt + \mathbf{G}_t d\mathbf{W}_t, \quad d\hat{\mathbf{X}}_t = \hat{\mathbf{B}}_t(\hat{\mathbf{X}}_t)dt + \hat{\mathbf{G}}_t d\mathbf{W}_t, \quad t \in [0, T], \quad (27)$$

where $(\mathbf{X}_t, \hat{\mathbf{X}}_t)$ is a pair of forward and reverse stochastic processes with coefficients $(\mathbf{B}_t, \mathbf{G}_t)_{t \in [0, T]}$ and $(\hat{\mathbf{B}}_t, \hat{\mathbf{G}}_t)_{t \in [0, T]}$, and \mathbf{W}_t is an \mathcal{H} -valued Wiener process. [24, Theorem 2.1] provides the time-reversal formula which connects pairs $(\mathbf{X}_t, \hat{\mathbf{X}}_t)$ to have the same marginal distribution:

$$\hat{\mathbf{B}}_t(u) = -\mathbf{B}_{T-t}(u) + \mathbf{S}_{T-t}(u), \quad \hat{\mathbf{G}}_t = \mathbf{G}_{T-t}, \quad t \in [0, T]. \quad (28)$$

Here \mathbf{S}_t denotes the score operator for each marginal probability measures $(\mu_t)_{t \in [0, T]}$ of the solution to the forward equation. Therefore, we can generate samples from $\hat{\mathbf{X}}_T \sim \mathbb{P}_{\text{data}}$ and approximate the score operator of \mathbb{P}_{data} in function spaces by training the score model $\hat{\mathbf{S}}^\theta(t, \cdot)$ and running the reverse equation with replacing the score operator $\mathbf{S}(t, \cdot)$ in (28) by the trained score model $\hat{\mathbf{S}}^\theta(t, \cdot)$ for each $t \in [0, T]$. Therefore, the learning objective of score-based generative modeling in function spaces is equivalent to the score matching between $\mathbf{S}(t, \cdot)$ and $\hat{\mathbf{S}}^\theta(t, \cdot)$ in the Hilbert space \mathcal{H} [23, 24].

Proof of Theorem 3.1 To explain the statement in Theorem 3.1 precisely, let us introduce necessary definitions and theorems. We first introduce Sobolev spaces and Sobolev embedding theorem [16], which are main elements for understanding a fundamental theory in neural operators [15].

Definition A.1 ([16, Chapter 1.3]). Let $W_2^k = W_2^k(\mathcal{X})$ be the Sobolev space with the norm:

$$\|f\|_{W_2^k} := \sum_{|\alpha| \leq k} \|\partial_{\mathbf{x}}^\alpha f\|_{L_2(\mathcal{X})} \quad (29)$$

with the inner-product:

$$\langle f, g \rangle_{W_2^k} := \sum_{|\alpha| \leq k} \langle \partial_{\mathbf{x}}^\alpha f, \partial_{\mathbf{x}}^\alpha g \rangle_{L_2}. \quad (30)$$

Remark A.2. Note that the inner-product (30) induces the norm which is equivalent to the Sobolev norm (29). Due to the Sobolev embedding theorem [16, Chapter 10], it holds that

$$W_2^k \subset C^{k - \frac{D}{2}}, \quad (31)$$

where $C^{k - \frac{D}{2}}$ denotes the Hölder space with the following Hölder-norm:

$$\|f\|_{C^{k - \frac{D}{2}}} := \max_{|\alpha| \leq m} \sup_{\mathbf{x} \in \mathcal{X}} \|\partial_{\mathbf{x}}^\alpha f(\mathbf{x})\| + \max_{|\beta| = m} \|\partial_{\mathbf{x}}^\beta f\|_{C^{k - \frac{D}{2} - m}}, \quad m = \lfloor k - \frac{D}{2} \rfloor. \quad (32)$$

Sobolev embedding (31) implies that for any $f \in W_2^k$, it holds

$$\|f\|_{C^{k - \frac{D}{2}}} \lesssim \|f\|_{W_2^k}, \quad (33)$$

where the notation \lesssim means the inequality \leq holds up to constant factors. Therefore, we can approximate a function including its derivatives by (33) if we can control the Sobolev norm (29).

Let us set $a := (\mathbf{B}_t, \mathbf{G}_t)_{t \in [0, T]}$ as *smooth* coefficients defined on $[0, T]$ and consider their function class as $\mathcal{A} = C([0, T])$, which determines the probability measure μ_T by simulating the forward stochastic evolution equation \mathbf{X}_t in (27). For a given $a \in \mathcal{A}$, let us define an operator $\mathcal{G}^\dagger(a) = \mathbf{S}_*$ which maps smooth coefficients a to the target score function $\mathbf{S}_* = \nabla_{\mathbf{x}} \log P_{\text{data}}$ of the data distribution \mathbb{P}_{data} on the data space \mathcal{X} . In this paper, we assume the target score function satisfies $\mathbf{S}_* \in W_2^k(\Omega)$ for any Lipschitz domain $\Omega \subset \mathcal{X}$ where $k > 2 + \frac{D}{2}$. This is a natural assumption for many probability densities which are continuously differentiable on Lipschitz subsets of \mathcal{X} .

Theorem A.1. Suppose the map $\mathcal{G}^\dagger : C([0, T]) \rightarrow W_2^k(\Omega)$ is continuous for any Lipschitz subset $\Omega \subset \mathcal{X}$ such that $\mathcal{G}^\dagger(a) = \mathbf{S}_*$. Then for any $\varepsilon > 0$ and compact set $\bar{A} \subset \mathcal{A}$, there exists a neural operator $\mathcal{G} : C([0, T]) \rightarrow W_2^k(\Omega)$ such that

$$\sup_{a \in \bar{A}} \|\mathcal{G}^\dagger(a) - \mathcal{G}(a)\|_{W_2^k(\Omega)} \leq \varepsilon. \quad (34)$$

Furthermore, the neural operator $\mathcal{G} : C^\infty([0, T]) \rightarrow W_2^k(\Omega)$ is discretization-invariant such that

$$\sup_{a \in \mathcal{A}} \|\mathcal{G}(a)\| \lesssim \sup_{a \in \mathcal{A}} \|\mathcal{G}^\dagger(a)\|, \quad (35)$$

where the notation \lesssim means the inequality \leq holds up to constant factors.

Proof. Due to [22, Theorem 8] and Remark A.2, the neural operator \mathcal{G} is discretization-invariant since $W_2^k(\Omega)$ is continuously embedded in $C(\bar{\Omega})$. We apply [22, Theorem 11] to show (34) and (35). Note that Assumption 9 holds since we set $\mathcal{A} = C([0, T])$ and so does Assumption 10 since we set $\mathcal{U} = W_2^k(\Omega)$. Therefore, for a given $\varepsilon > 0$ and a compact set $\bar{A} \subset \mathcal{A}$, by [22, Theorem 11], there exists a neural operator \mathcal{G} such that

$$\sup_{a \in \bar{A}} \|\mathcal{G}^\dagger(a) - \mathcal{G}(a)\|_{W_2^k(\Omega)} \leq \varepsilon. \quad (36)$$

Thus, (34) is proved. Also, (35) holds automatically because $\mathcal{U} = W_2^k(\Omega)$ is a Hilbert space with the inner product (30). The theorem is proved. \square

Remark A.3. Theorem A.1 implies that there exists a set of L -layer neural operators which can approximate the operator \mathcal{G}^\dagger which acts on a set of smooth coefficients $a = (\mathbf{B}_t, \mathbf{G}_t)_{t \in [0, T]}$. Since $(\mathbf{B}_t, \mathbf{G}_t)_{t \in [0, T]}$ determines marginal densities $(\mu_t : t \in [0, T])$ of stochastic equations (27), the inequality (34) in Theorem A.1 enhances the approximation to weak derivatives of the target score function \mathbf{S}_* in the Sobolev space. To obtain the desired approximation result in Hölder space, we utilize the Sobolev embedding theorem (see Remark A.2), upon Theorem A.1.

Now we prove the main result. Let us restate Theorem 3.1 formally.

Theorem A.2. Let $k \in \mathbb{N}$ such that $k > 2 + \frac{D}{2}$. For any compact subset $K \subset \Omega \subset \mathcal{X}$ and $\varepsilon > 0$, there exists a neural operator $\hat{\mathbf{S}}^\theta$ such that

$$\sup_{\mathbf{x} \in K} \sum_{|\alpha| \leq m} \left| \partial_{\mathbf{x}}^\alpha \mathbf{S}(\mathbf{x}) - \partial_{\mathbf{x}}^\alpha \hat{\mathbf{S}}^\theta(\mathbf{x}) \right| \leq \varepsilon, \quad m = \lfloor k - \frac{D}{2} \rfloor. \quad (37)$$

Proof. Let $\varepsilon > 0$ be given and fix a compact set $K \subset \Omega$. Due to Theorem A.1, there exists a neural operator \mathcal{G} which maps a smooth coefficients $a = (\mathbf{B}_t, \mathbf{G}_t)_{t \in [0, T]} \in \mathcal{A}$ to $W_2^k(\Omega)$ -valued function, $\mathcal{G}(a) = \hat{\mathbf{S}}^\theta : \Omega \rightarrow \mathbb{R}^D$, with the approximation power (34). Therefore, by (34), it holds that

$$\|\mathbf{S}_* - \hat{\mathbf{S}}^\theta\|_{W_2^k(\Omega)} \leq \varepsilon/C, \quad (38)$$

where the constant C is determined by (31). Having $K \subset \Omega$ in mind, by (31) and (32),

$$\sup_{\mathbf{x} \in K} \sum_{|\alpha| \leq m} \left| \partial_{\mathbf{x}}^\alpha \mathbf{S}(\mathbf{x}) - \partial_{\mathbf{x}}^\alpha \hat{\mathbf{S}}^\theta(\mathbf{x}) \right| \leq \|\mathbf{S}_* - \hat{\mathbf{S}}^\theta\|_{C^{k-\frac{D}{2}}(\Omega)} \leq C \|\mathbf{S}_* - \hat{\mathbf{S}}^\theta\|_{W_2^k(\Omega)} \leq \varepsilon, \quad (39)$$

hence we obtain (37). The theorem is proved. \square

Remark A.4. Since we assume $k > 2 + \frac{D}{2}$, we have $m \geq 2$ so that $\hat{\mathbf{S}}^\theta$ is at least twice continuously differentiable which can approximate the target score function \mathbf{S}_* with respect to the Hölder norm $\|\cdot\|_{C^{k-\frac{D}{2}}}$. Theorems A.1 and A.2 are crucial for the approximation (6) to the Hessian diagonal $\mathcal{D}(\mathbf{x})$ since (35) supports the numerical stability and (37) guarantees the approximation power for computing the second-order derivatives of score models $\hat{\mathbf{S}}^\theta$.

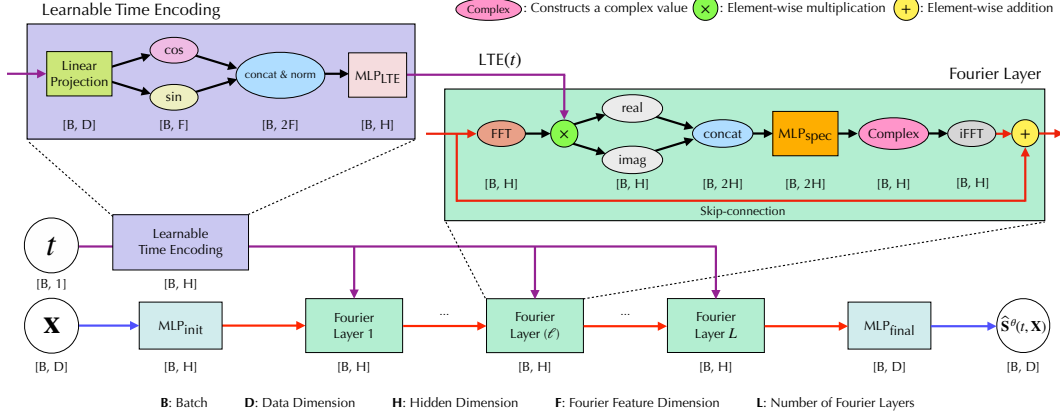


Figure A.1: Architecture of Score-informed Neural Operator

A.3 Architectural Details of SciNO

A.3.1 Overview of the SciNO Architecture

A D -dimensional input data $\mathbf{X} \in \mathbb{R}^D$ is passed through the initial $\text{MLP}_{\text{init}} : \mathbb{R}^D \rightarrow \mathbb{R}^H$ layer:

$$\mathbf{X}^{(1)} = \text{MLP}_{\text{init}}(\mathbf{X}). \quad (40)$$

To explain the intermediate layers in SciNO, let $\ell \in \{1, \dots, L\}$ denote the index of Fourier layers. Then we transform $\mathbf{X}^{(\ell)}$ into the spectral domain to get the following \mathbb{C}^H -dimensional tensor:

$$\xi^{(\ell)} = \text{FFT}(\mathbf{X}^{(\ell)}) \odot \text{LTE}(t), \quad t \in [0, 1] \quad (41)$$

where FFT means the Fast Fourier Transform, \odot denotes the element-wise multiplication, and $\text{LTE} : [0, 1] \rightarrow \mathbb{R}^H$ is the *Learnable Time Encoding* module. The real $\Re(\xi^{(\ell)})$ and imaginary $\Im(\xi^{(\ell)})$ parts are then split to construct an \mathbb{R}^{2H} -valued tensor $\chi^{(\ell)} = [\Re(\xi^{(\ell)}), \Im(\xi^{(\ell)})]$. Then we plug the tensor into an $\text{MLP}_{\text{spec}}^{(\ell)} : \mathbb{R}^{2H} \rightarrow \mathbb{R}^{2H}$ layer in the spectral domain:

$$\mathbf{Z}^{(\ell)} = \text{MLP}_{\text{spec}}^{(\ell)}(\chi^{(\ell)}). \quad (42)$$

Then we combine $\mathbf{Z}^{(\ell)} = [\mathbf{Z}_{\text{real}}^{(\ell)}, \mathbf{Z}_{\text{imag}}^{(\ell)}]$ to get a \mathbb{C}^H -valued tensor $\zeta^{(\ell)} = \mathbf{Z}_{\text{real}}^{(\ell)} + i\mathbf{Z}_{\text{imag}}^{(\ell)}$. Next, we transform the signal $\zeta^{(\ell)}$ from the spectral domain to the spatial domain with skip connection, which helps alleviate gradient vanishing during training:

$$\mathbf{X}^{(\ell+1)} = \mathbf{X}^{(\ell)} + \text{iFFT}(\zeta^{(\ell)}), \quad (43)$$

where $\text{iFFT}(\cdot)$ denotes the inverse Fourier transformation. By repeating the above procedure for $\ell \in \{1, \dots, L\}$, we get the output of the Fourier layer $\mathbf{X}^{(L+1)}$. Then $\mathbf{X}^{(L+1)}$ is passed through the final $\text{MLP}_{\text{final}} : \mathbb{R}^H \rightarrow \mathbb{R}^D$ layer for final processing:

$$\hat{\mathbf{S}}^\theta(t, \mathbf{X}) = \text{MLP}_{\text{final}}(\mathbf{X}^{(L+1)}). \quad (44)$$

A.3.2 MLP Modules in SciNO

The MLP modules used in SciNO are organized as follows.

Initial MLP Layer. The input data $\mathbf{X} \in \mathbb{R}^D$ is first projected into \mathbb{R}^H through MLP_{init} :

$$\text{MLP}_{\text{init}}(\mathbf{X}) = \text{Dropout}_{0.2}(\text{LayerNorm}(\text{LeakyReLU}(W_{\text{init}}\mathbf{X}))) \in \mathbb{R}^H, \quad (45)$$

where $W_{\text{init}} \in \mathbb{R}^{H \times D}$.

887 **Spectral MLP Layer.** MLP_{spec} is designed to effectively capture differential information in the
 888 frequency domain. The input $\chi^{(\ell)} \in \mathbb{R}^{2H}$ is processed as follows:

$$\text{MLP}_{\text{spec}}^{(\ell)}(\chi^{(\ell)}) = \text{BatchNorm}\left(\text{LeakyReLU}\left(W_{\text{spec}}^{(\ell)}\chi^{(\ell)} + b_{\text{spec}}^{(\ell)}\right)\right) \in \mathbb{R}^{2H}. \quad (46)$$

889 The layer is parameterized by $W_{\text{spec}}^{(\ell)}$ and $b_{\text{spec}}^{(\ell)}$, with shapes $\mathbb{R}^{2H \times 2H}$ and \mathbb{R}^{2H} .

890 **Final MLP Layer.** The output $\mathbf{X}^{(L+1)} \in \mathbb{R}^H$ from the final Fourier layer is mapped to \mathbb{R}^D via
 891 $\text{MLP}_{\text{final}}$:

$$\mathbf{h}_1 = \text{LeakyReLU}(W_{\text{final}}^{(1)}\mathbf{X}^{(L+1)} + b_{\text{final}}^{(1)}), \quad (47)$$

$$\mathbf{h}_2 = \text{LeakyReLU}(W_{\text{final}}^{(2)}\mathbf{h}_1 + b_{\text{final}}^{(2)}), \quad (48)$$

$$\text{MLP}_{\text{final}}(\mathbf{X}^{(L+1)}) = W_{\text{final}}^{(3)}\mathbf{h}_2 + b_{\text{final}}^{(3)}. \quad (49)$$

892 Here, $W_{\text{final}}^{(1)} \in \mathbb{R}^{H \times H}$, $b_{\text{final}}^{(1)} \in \mathbb{R}^H$, $W_{\text{final}}^{(2)} \in \mathbb{R}^{S \times H}$, $b_{\text{final}}^{(2)} \in \mathbb{R}^S$, $W_{\text{final}}^{(3)} \in \mathbb{R}^{D \times S}$, and $b_{\text{final}}^{(3)} \in \mathbb{R}^D$
 893 denote the weights and biases of the $\text{MLP}_{\text{final}}$. The dimension S denotes an intermediate hidden
 894 dimension between D and H .

895 **Time Encoding MLP Layer.** Given the Fourier feature $\Phi(t) \in \mathbb{R}^{2F}$, the time encoding $\text{LTE}(t)$ is
 896 computed via MLP_{LTE} :

$$\text{MLP}_{\text{LTE}}(\Phi(t)) = W_{\text{LTE}}^{(2)}\text{GeLU}\left(W_{\text{LTE}}^{(1)}\Phi(t) + b_{\text{LTE}}^{(1)}\right) + b_{\text{LTE}}^{(2)} \in \mathbb{R}^H, \quad (50)$$

897 where $W_{\text{LTE}}^{(1)} \in \mathbb{R}^{M \times 2F}$, $b_{\text{LTE}}^{(1)} \in \mathbb{R}^M$, $W_{\text{LTE}}^{(2)} \in \mathbb{R}^{H \times M}$, and $b_{\text{LTE}}^{(2)} \in \mathbb{R}^H$. Here, M denotes the hidden
 898 dimension of the intermediate layer in the Time Encoding MLP.

899 A.4 Experimental Setting in Sections 4.1 and 4.2

900 A.4.1 Datasets

901 **Physics** We generate a Physics commonsense-based synthetic dataset using the physics-based
 902 causal DAG introduced in [18], which models causal relations among 7 variables related to water
 903 evaporation. Each edge weight is sampled from the following uniform distributions:

$$e \sim \mathcal{U}(-1, -0.1) \cup \mathcal{U}(0.1, 1). \quad (51)$$

904 Using the sampled adjacency matrix A , we generate 5,000 samples via the following nonlinear
 905 Structural Equation Model (SEM):

$$x = 2 \sin(A^\top(x + 0.5 \cdot \mathbf{1})) + A^\top(x + 0.5 \cdot \mathbf{1}) + z, \quad z \sim \mathcal{N}(0, 1). \quad (52)$$

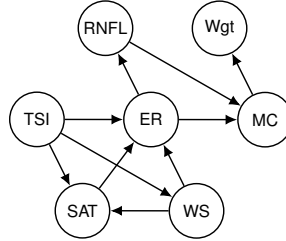


Figure A.2: Physics commonsense-based synthetic graph with 7 nodes: Rainfall (RNFL), Total Solar Irradiance (TSI), Surface Air Temperature (SAT), Wind Speed (WS), Evaporation Rate (ER), Moisture Content (MC), and Object Weight (Wgt).

906 **Sachs** The Sachs dataset [32] contains 7,466 single-cell measurements of 11 proteins and phospho-
 907 proteins involved in human immune signaling, collected under various stimulation conditions using
 908 flow cytometry. To ensure a DAG structure for evaluating causal order, we exclude terminal nodes
 909 (praf, plcγ, and PIP2) involved in cycles and use the remaining 8-node graph for experiments.

BNLearn We use four real-world DAGs from the BNLearn repository—MAGIC-NIAB [36] (44 nodes/ 66 edges), ECOLI70 [34] (46 nodes/ 70 edges), MAGIC-IRRI [35] (64 nodes/ 102 edges), and ARTH150 [26] (107 nodes/ 150 edges)—and generate data based on their causal structures. For the linear case, samples are generated using the graph structures and linear functional relationships provided by BNLearn. For the nonlinear case, we generate data using nonlinear SEMs defined over the same graphs, where each functional relationship is parameterized by Multilayer Perceptron(MLP). We generate 10,000 samples per graph in both linear and nonlinear settings for evaluation.

917 A.4.2 Metrics

Order Divergence. Given a causal order π and a true adjacency matrix \mathcal{G} , the OD(Order Divergence, [31]) is defined as:

$$D_{\text{top}}(\pi, \mathcal{G}) = \sum_{i=1}^D \sum_{j: \pi_i > \pi_j} \mathcal{G}_{ij}. \quad (53)$$

Order divergence counts the number of edges in \mathcal{G} that are inconsistent with the order π . A smaller value indicates a closer alignment between π and \mathcal{G} .

SHD. SHD(Structural Hamming Distance) [38] measures the structural discrepancy between a predicted graph $\hat{\mathcal{G}}$ and the ground-truth graph \mathcal{G} . It is defined as the total number of edge insertions, deletions, and reversals required to convert $\hat{\mathcal{G}}$ into \mathcal{G} :

$$\text{SHD}(\mathcal{G}, \hat{\mathcal{G}}) = \# \left\{ (i, j) \in \mathcal{V}^2 \mid \mathcal{G} \text{ and } \hat{\mathcal{G}} \text{ do not have the same type of edge between } i \text{ and } j \right\}. \quad (54)$$

A lower SHD indicates that the predicted graph $\hat{\mathcal{G}}$ is closer to the true causal graph \mathcal{G} in terms of structural similarity.

SID. SID(Structural Intervention Distance) [27] quantifies how similarly a predicted graph $\hat{\mathcal{G}}$ and the ground-truth graph \mathcal{G} infer causal relationships under intervention:

$$\text{SID}(\mathcal{G}, \hat{\mathcal{G}}) = \# \left\{ (i, j), i \neq j \mid \begin{array}{l} \text{Pa}^{\hat{\mathcal{G}}}(X_i) \text{ not a valid adjustment set for} \\ \text{the intervention } X_j | \text{do}(X_i) \text{ in graph } \mathcal{G} \end{array} \right\}. \quad (55)$$

A lower SID indicates that the predicted graph $\hat{\mathcal{G}}$ is more causally consistent with the true causal graph \mathcal{G} under interventions, making SID a suitable metric for evaluating the reliability of causal inference.

932 A.4.3 Implementation Details

For synthetic datasets, we set the number of Fourier layers to $L = 10$, and for real-world datasets, we use $L = 1$. The Fourier feature dimension in the LTE module (50) is fixed at $F = 32$. We set the hidden dimensions of the model as $H = \max(1024, 5D)$, and $S = \max(128, 3D)$ in (45), (46), and (47), where D denotes the number of nodes $|\mathcal{V}|$. All experiments are conducted on a single NVIDIA A6000 GPU. Data, checkpoints, and all training configurations are publicly available¹.

¹<https://anonymous.4open.science/r/scino-llm-neurips2025-43E2>

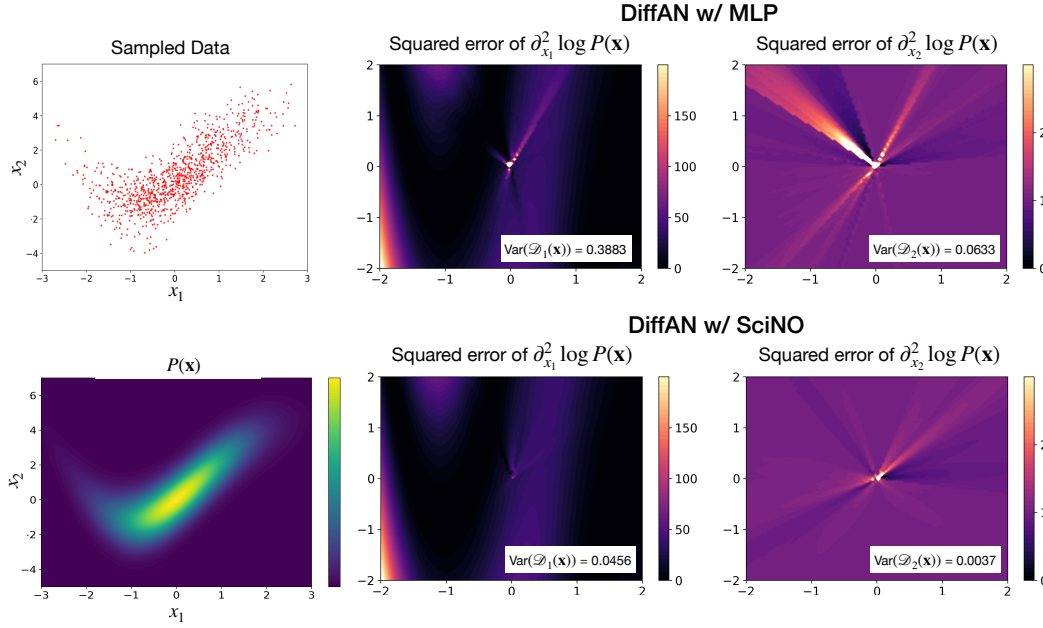


Figure A.3: Comparison of Hessian diagonal approximation between DiffAN with MLP (top row) and DiffAN with SciNO (bottom row). The left column shows sampled data and its true density. The center and right columns illustrate the squared approximation error of the second derivatives of $\log P(\mathbf{x})$ with respect to x_1 and x_2 , computed over a meshgrid.

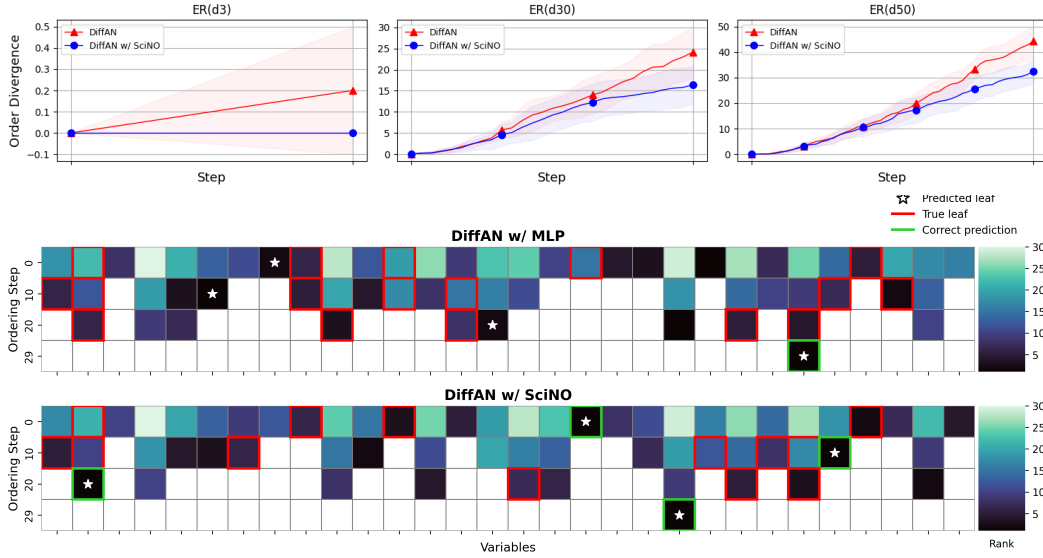


Figure A.4: **(Top)** Order divergence in ER datasets per each step. The solid lines represent the average order divergence, and the shaded regions indicate the 95% confidence intervals across 10 random graphs with $D \in \{3, 30, 50\}$. **(Bottom)** Heatmaps of variance of the Hessian diagonal estimated by DiffAN with MLP and SciNO, respectively. Darker colors indicate lower estimated variance. Red boxes \square denote ground-truth causal leaves, and white stars \star indicate the variables selected by models. Green boxes \square denote cases where the model predicts leaf node correctly.

A.5 Controlling LLM for Causal Ordering

A.5.1 Experimental Details in Section 4.3

Implementation Details in LLM Control The LLM prompt is designed to select a leaf node, a variable among the unordered variables that does not causally influence any other variable. Each prompt consists of an **Unordered Variables** list, a **Data Description** summarizing the overall data context, and **Variable Descriptions** providing natural language descriptions of each variable (as illustrated in Figure A.6). Simple input-output examples are provided within the prompt to clarify the task objective and promote consistent output behavior. To avoid formatting inconsistencies such as stray whitespace or mismatched capitalization, each variable name is prefixed with **node_** and presented in the format **node_variable_name**.

For each variable name, the LLM generates token-level conditional probabilities for the sequence of tokens t_1, t_2, \dots, t_n that constitute v , where $v = (t_1, \dots, t_n)$. The probability of generating the full variable name v is computed using the chain rule:

$$P_{\text{LLM}}(v \mid \text{context}) = \prod_{i=1}^n P_{\text{LLM}}(t_i \mid t_1, \dots, t_{i-1}, \text{context}). \quad (56)$$

Here, **context** refers to contextual information, including variable descriptions or domain knowledge. In practice, we construct the full input sequence by appending each variable name to the prompt in the form of **node_variable_name**, and then feed the entire sequence to the model in a single forward pass. To compute the probability of each variable name, we consider only the tokens following the common prefix **node**, since tokenization often splits strings like **node_v** into **node** and **_v**. This allows us to isolate the informative portion of the name and compute joint probabilities efficiently, without repeatedly modifying the prompt.

Length Normalization Since variable names typically consist of multiple tokens with varying length, using raw probabilities introduces a bias against longer names. To address this, we apply length normalization by exponentiating the average of the log-likelihoods of the tokens composing the variable name [41]. To further mitigate over-penalization of longer names, we normalize the value by dividing it by length^α ($0 < \alpha \leq 1$). In this paper, we use $\alpha \in \{0.5, 1.0\}$.

$$P_{\alpha\text{-norm}}(v \mid \text{context}) = \exp\left(\frac{1}{n^\alpha} \sum_{i=1}^n \log P_{\text{LLM}}(t_i \mid t_1, \dots, t_{i-1}, \text{context})\right). \quad (57)$$

Leaf node selection through LLM control. Llama-3 provides token-level probabilities directly, enabling us to compute the leaf node probabilities using (57). The leaf node is selected as $v^* = \arg \max_{v \in \mathcal{V} \setminus \pi_k} P_{\alpha\text{-norm}}(v)$, i.e., the variable with the highest length-normalized probability, where π_k denotes the set of selected leaf nodes up to step k . For GPT-4o, we treat the variables mentioned in the final response as leaf nodes, since the model’s probability distribution cannot be externally controlled during generation. We set the temperature to its default value of 1 and perform three inference runs per dataset, reporting the result with the highest performance.

Results Figure A.5 presents the results under the setting where all variables are well-described, i.e., $\mathcal{V}_{\text{context}} = \mathcal{V}$. The top plot compares the LLM prior probabilities with the posterior probabilities—updated via control—of the ground-truth leaf nodes at each step of the causal reasoning process. A substantial increase in the posterior probability is consistently observed across high-dimensional datasets such as ECOLI70, MAGIC-NIAB, MAGIC-IRRI, and ARTH150. The effect is especially pronounced in the early stages of the causal ordering process, where the true causal structure must be inferred from a large pool of candidate nodes. Furthermore, the bottom plot illustrates several cases in which the posterior successfully identifies the true leaf node, even when SciNO alone fails to do so. These results indicate that the LLM prior can be complemented through integration with SciNO’s data-informed evidence, while the semantic knowledge embedded in the LLM can also compensate for incomplete or uncertain statistical inference.

A.5.2 Additional Experiments on LLM Control

In many domains such as biomedicine, healthcare, and finance, variable names are deliberately abstracted and metadata is often scarce and uncertain. To account for this practical challenge, we

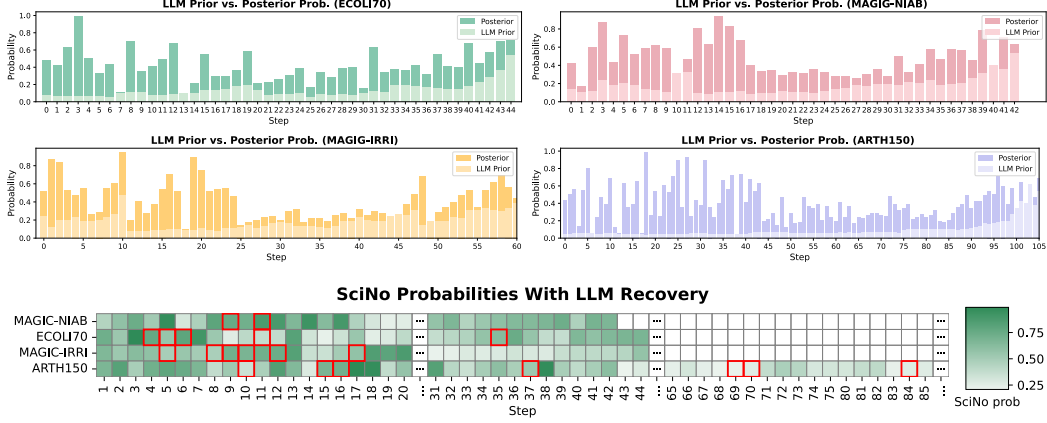


Figure A.5: **(Top)** Comparison of LLM prior and posterior probabilities for the true leaf nodes across four datasets (ECOLI70, MAGIC-NIAB, MAGIC-IRRI, and ARTH150). Posterior probabilities are computed by multiplying the LLM prior with SciNO computed probabilities. **(Bottom)** Heatmap of SciNO probabilities for the SciNO-selected nodes at each ordering step across datasets. Red boxes \square denote steps where SciNO alone fails to select the correct leaf node, but the correct node is successfully recovered via integration with LLM priors.

984 conduct an additional experiment in which all variable names are masked and the proportion of
 985 variables accompanied by descriptions is systematically varied.

986 **Experimental Setting** We consider an experiment to simulate practical conditions involving a
 987 mixture of $\mathcal{V}_{\text{context}}$ and $\mathcal{V}_{\neg\text{context}} \neq \emptyset$, where $\mathcal{V}_{\text{context}}$ refers to variables that contain contextual or
 988 domain-specific information, whereas $\mathcal{V}_{\neg\text{context}}$ includes variables without such information. Variable
 989 names are replaced with randomly generated four-letter alphabetic strings (e.g., ‘zntr’, ‘lgvp’,
 990 ‘aqsm’), and descriptions are provided for 10%, 40%, or 70% of the variables. Each experiment is
 991 repeated three times, with randomization applied to both the selection of context-given variables
 992 and their input order. We use real datasets from BNLearn: MAGIC-NIAB (7/44 variables have
 993 descriptions), ECOLI70 (41/46), MAGIC-IRRI (10/64), and ARTH150 (105/107). Due to the limited
 994 availability of variable descriptions in MAGIC-NIAB and MAGIC-IRRI, we restrict experiments on
 995 these two datasets to the 10% description setting.

996 **Control with *Hard* and *Soft* Supervision** Under the above experimental conditions, we follow the
 997 causal ordering procedure described in Algorithm 1. The control mechanism is formalized as:

$$\mathbb{P}(x_{t+1}|x_{1:t}, \text{stat}, \text{context}) \propto \mathbb{P}_{\text{AR}}(x_{t+1}|x_{1:t}, \text{context}) \times \mathbb{P}_{\text{SciNO}}(\text{stat}|x_{1:t+1}). \quad (58)$$

998 Let π_t denote the partial causal order determined by $x_{1:t}$, the sequence of nodes selected in the
 999 previous steps. The evidence term $\mathbb{P}_{\text{SciNO}}(\text{stat}|x_{1:t+1})$ is estimated using two approaches introduced
 1000 in Section 3.3. The first estimator relies on the average rank $\bar{r}(i)$ for each node $i \in \mathcal{V} \setminus \pi_t$ across
 1001 models:

$$\hat{P}_{m \sim M}^{(\text{rank})}(\sigma_i^{(m)} < \sigma_j^{(m)} | x_{1:t}, x_{t+1} = i) = \frac{\exp(-\bar{r}(i))}{\sum_{j \in \mathcal{V} \setminus \pi_t} \exp(-\bar{r}(j))}, \quad \bar{r}(i) = \frac{1}{M} \sum_{m=1}^M r_i^{(m)}. \quad (59)$$

1002 Here, $r_i^{(m)}$ is the rank of node i in model m , based on the variance of its Hessian diagonal. The
 1003 second estimator is based on confidence intervals:

$$\hat{P}_{m \sim M}^{(\text{CI})}(\text{CI}_{\text{lower}}(i) \leq \text{CI}_{\text{upper}}(j_{\min}^{(m)}) | x_{1:t}, x_{t+1} = i) = \frac{1}{M} \sum_{m=1}^M \mathbb{1} [\text{CI}_{\text{lower}}(i) \leq \text{CI}_{\text{upper}}(j_{\min}^{(m)})]. \quad (60)$$

1004 where $j_{\min}^{(m)}$ is the node with the smallest variance of the Hessian diagonal in model m .

1005 Depending on whether variable descriptions are available, we apply two supervision strategies. For
 1006 variables without descriptions $\mathcal{V}_{\neg\text{context}}$, we use *hard supervision*, relying exclusively on the evidence

term estimated by (10) or (12):

$$P(v) \propto \frac{1}{|\mathcal{V} \setminus \pi_t|} \hat{P}_{\text{SciNO}}(\text{stat}|\pi_t, x_{t+1} = v). \quad (61)$$

For variables with descriptions $\mathcal{V}_{\text{context}}$, we apply *soft supervision* by multiplying the LLM-derived prior (56) to the evidence term (10) or (12):

$$P(v) \propto P_{\alpha\text{-norm}}(x_{t+1} = v|\pi_t, \text{context}) \cdot \hat{P}_{\text{SciNO}}(\text{stat}|\pi_t, x_{t+1} = v). \quad (62)$$

To adapt the soft supervision signal, one can use a temperature parameter $\tau \geq 0$:

$$\hat{P}_\tau(\text{stat}|\pi_t, x_{t+1} = v) = \left(\hat{P}_{\text{SciNO}}(\text{stat}|\pi_t, x_{t+1} = v) \right)^\tau, \quad (63)$$

$$\hat{P}_{\tau\text{-soft}}(\text{stat}|\pi_t, v) = \text{softmax}(\hat{P}_\tau(\text{stat}|\pi_t, x_{t+1} = v)). \quad (64)$$

We choose $\tau \in \{0, 1, 2, 3, 4, 5\}$ for experiments. When $\tau > 5$, the evidence term grows exponentially and overwhelms the prior, leading to predictions that are nearly indistinguishable from using the data statistic alone. Applying the above τ -temperature to the soft supervision, the posterior predictive term is computed by:

$$P(v) \propto P_{\alpha\text{-norm}}(v|\mathbf{x}_\pi, \text{context}) \cdot \hat{P}_{\tau\text{-soft}}(\text{stat}|\pi_t, x_{t+1} = v), \quad v \in \mathcal{V} \setminus \pi_t. \quad (65)$$

Then we select the variable with the highest posterior probability as the next leaf node:

$$v^* = \underset{v \in \mathcal{V} \setminus \pi_t}{\text{argmax}} P(v). \quad (66)$$

Results Table A.1 reports the order divergence under the setting where $\mathcal{V}_{\text{context}} \neq \mathcal{V}$, across varying levels of description proportions and τ values. Compared to the results in Table 4, performance without control reveals that the LLM’s ability to infer causality deteriorates as descriptive contents decrease and variable names are masked. Across all datasets, increasing τ from 0 to 1 consistently improves performance, even under low description conditions (e.g., 10%), indicating robustness in the combination of LLM prior and SciNO evidence. When the description is limited, performance tends to plateau after $\tau = 1$, suggesting that the contribution from LLM priors has been fully exploited, leaving SciNO as the dominant signal. Conversely, with a higher description ratio, the effect of soft supervision becomes more pronounced as τ varies, allowing the model to better leverage both semantic and structural signals. Interestingly, a higher τ does not always lead to better results. This highlights the importance of finding an optimal balance between the LLM’s semantic knowledge and the evidence from SciNO. We compare various combinations of description proportions and τ values. Among them, the configuration using 70% descriptions with $\tau = 5$ achieves the highest performance on the ARTH150 dataset. It outperforms all other tested combinations, including those with lower description ratios, different τ values, and even the use of SciNO alone. This suggests that when richer semantic information is available, the integration with SciNO becomes especially effective. In conclusion, these results demonstrates that even under limited description conditions, combining LLM semantic knowledge with data-driven causal inference can lead to improved performance.

| Dataset | Proportion | Method | w/o control | $\tau = 0$ | $\tau = 1$ | $\tau = 2$ | $\tau = 3$ | $\tau = 4$ | $\tau = 5$ |
|------------|------------|---|-----------------|-----------------|----------------------------------|---------------------------------|---------------------------------|----------------------------------|----------------------------------|
| MAGIC-NIAB | 10% | Llama-3.1-8b(1-norm) w/ control(rank) | 24.7 \pm 9.3 | 5.3 \pm 0.6 | 3.3 \pm 0.6 | 3.3 \pm 0.6 | 3.3 \pm 0.6 | 3.7 \pm 0.6 | 3.7 \pm 0.6 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 22.3 \pm 7.6 | 4.7 \pm 0.6 | 3.3 \pm 0.6 | 3.3 \pm 0.6 | 3.3 \pm 0.6 | 3.7 \pm 0.6 | 3.7 \pm 0.6 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 24.7 \pm 9.3 | 5.0 \pm 1.0 | 4.0 \pm 0.0 | 4.0 \pm 0.0 | 4.0 \pm 0.0 | 4.0 \pm 0.6 | 4.0 \pm 0.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 22.3 \pm 7.6 | 5.0 \pm 1.0 | 4.0 \pm 0.0 | 4.0 \pm 0.0 | 4.0 \pm 0.0 | 4.3 \pm 0.6 | 4.3 \pm 0.6 |
| ECOLI70 | 70% | Llama-3.1-8b(1-norm) w/ control(rank) | 30.3 \pm 2.1 | 35.7 \pm 2.9 | 8.7 \pm 0.6 | 8.3 \pm 0.6 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.3 \pm 0.6 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 28.7 \pm 2.9 | 39.0 \pm 4.0 | 9.0 \pm 0.0 | 8.3 \pm 0.6 | 8.3 \pm 0.6 | 8.3 \pm 0.6 | 8.7 \pm 0.6 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 30.3 \pm 2.1 | 24.0 \pm 2.0 | 11.3 \pm 1.5 | 11.7 \pm 2.3 | 12.7 \pm 1.5 | 12.0 \pm 1.0 | 11.7 \pm 1.2 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 28.7 \pm 2.9 | 27.3 \pm 4.2 | 11.0 \pm 2.7 | 12.7 \pm 1.5 | 12.0 \pm 1.0 | 12.3 \pm 3.8 | 10.0 \pm 2.0 |
| | 40% | Llama-3.1-8b(1-norm) w/ control(rank) | 33.0 \pm 1.7 | 21.7 \pm 8.5 | 8.3 \pm 0.6 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 36.7 \pm 9.7 | 20.7 \pm 7.5 | 9.3 \pm 0.6 | 8.3 \pm 0.6 | 8.3 \pm 0.6 | 8.3 \pm 0.6 | 8.3 \pm 0.6 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 33.0 \pm 1.7 | 17.0 \pm 4.4 | 12.3 \pm 4.0 | 11.7 \pm 0.6 | 11.7 \pm 0.6 | 11.7 \pm 0.6 | 11.7 \pm 0.6 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 36.7 \pm 9.7 | 16.3 \pm 3.2 | 12.7 \pm 4.0 | 11.0 \pm 1.7 | 11.0 \pm 1.7 | 11.0 \pm 1.7 | 11.0 \pm 1.7 |
| | 10% | Llama-3.1-8b(1-norm) w/ control(rank) | 34.3 \pm 3.8 | 13.0 \pm 3.6 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 33.3 \pm 2.5 | 14.0 \pm 2.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 | 8.0 \pm 0.0 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 34.3 \pm 3.8 | 13.0 \pm 3.6 | 11.3 \pm 0.6 | 11.0 \pm 1.0 | 10.7 \pm 0.6 | 10.7 \pm 1.5 | 10.7 \pm 1.5 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 33.3 \pm 2.5 | 12.7 \pm 3.8 | 11.0 \pm 0.0 | 10.7 \pm 0.6 | 10.7 \pm 1.5 | 10.7 \pm 1.5 | 11.0 \pm 1.0 |
| MAGIC-IRRI | 10% | Llama-3.1-8b(1-norm) w/ control(rank) | 29.7 \pm 1.5 | 11.3 \pm 1.2 | 10.3 \pm 0.6 | 9.7 \pm 0.6 | 9.3 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 25.3 \pm 2.9 | 9.7 \pm 2.1 | 9.7 \pm 1.2 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.3 \pm 0.6 | 9.7 \pm 0.6 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 29.7 \pm 1.5 | 10.0 \pm 1.0 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 25.3 \pm 2.9 | 10.3 \pm 1.5 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 | 9.7 \pm 0.6 |
| ARTH150 | 70% | Llama-3.1-8b(1-norm) w/ control(rank) | 73.0 \pm 3.5 | 80.7 \pm 1.2 | 20.3 \pm 1.2 | 20.7 \pm 0.6 | 21.3 \pm 1.2 | 21.0 \pm 1.0 | 21.0 \pm 1.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 64.3 \pm 3.5 | 73.3 \pm 8.6 | 19.7 \pm 0.6 | 19.7 \pm 0.6 | 20.0 \pm 0.0 | 20.3 \pm 0.6 | 20.3 \pm 0.6 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 73.0 \pm 3.5 | 77.3 \pm 5.5 | 20.0 \pm 1.7 | 19.7 \pm 1.2 | 19.0 \pm 1.7 | 20.0 \pm 2.0 | 19.3 \pm 2.1 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 64.3 \pm 3.5 | 71.3 \pm 8.1 | 17.7 \pm 0.6 | 17.0 \pm 1.0 | 16.7 \pm 2.1 | 16.3 \pm 1.2 | 16.3 \pm 1.2 |
| | 40% | Llama-3.1-8b(1-norm) w/ control(rank) | 76.0 \pm 10.4 | 81.3 \pm 12.9 | 19.7 \pm 1.2 | 20.0 \pm 1.0 | 20.0 \pm 0.0 | 20.0 \pm 0.0 | 20.0 \pm 0.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 87.7 \pm 1.2 | 77.7 \pm 9.0 | 19.0 \pm 0.0 | 19.3 \pm 0.6 | 20.0 \pm 0.0 | 20.0 \pm 0.0 | 20.0 \pm 0.0 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 76.0 \pm 10.4 | 57.0 \pm 25.5 | 22.3 \pm 1.2 | 20.7 \pm 1.2 | 20.7 \pm 1.2 | 20.0 \pm 1.7 | 20.0 \pm 2.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 87.7 \pm 1.2 | 59.3 \pm 18.5 | 19.3 \pm 1.5 | 21.7 \pm 0.6 | 20.7 \pm 1.2 | 20.7 \pm 1.2 | 19.7 \pm 1.5 |
| | 10% | Llama-3.1-8b(1-norm) w/ control(rank) | 76.0 \pm 7.6 | 40.7 \pm 17.4 | 19.7 \pm 0.6 | 20.0 \pm 1.0 | 20.0 \pm 1.0 | 20.7 \pm 1.2 | 20.7 \pm 1.2 |
| | | Llama-3.1-8b(0.5-norm) w/ control(rank) | 88.7 \pm 6.4 | 37.0 \pm 13.8 | 19.7 \pm 0.6 | 19.7 \pm 0.6 | 20.3 \pm 1.5 | 20.7 \pm 1.2 | 20.7 \pm 1.2 |
| | | Llama-3.1-8b(1-norm) w/ control(CI) | 76.0 \pm 7.6 | 24.7 \pm 8.1 | 19.3 \pm 1.5 | 19.3 \pm 1.5 | 19.7 \pm 2.1 | 19.0 \pm 1.0 | 19.0 \pm 1.0 |
| | | Llama-3.1-8b(0.5-norm) w/ control(CI) | 88.7 \pm 6.4 | 25.0 \pm 7.0 | 19.3 \pm 1.5 | 19.3 \pm 1.5 | 19.7 \pm 2.1 | 19.0 \pm 1.0 | 19.0 \pm 1.0 |

Table A.1: Comparison of OD across real datasets under the control with hard and soft supervision setting with varying description proportions (70%, 40%, 10%) and temperature parameters $\tau \in \{0, 1, 2, 3, 4, 5\}$. The lowest OD for each dataset and description proportion is highlighted in bold. Each score is recorded over 3 independent runs.

You are an AI assistant tasked with identifying the most likely leaf node in a causal structure.
 A leaf node is a variable that does not cause any other variables in the unordered variables set.
 Your goal is to determine the best leaf node among **Unordered Variables** using the given information.

Selection Criteria:

- A leaf node does not act as a cause for any other variable in **Unordered Variables**.
- If multiple candidates exist, select the one that is influenced by others but does not influence any other variable in **Unordered Variables**.

Important Formatting Rules:

- Respond **only** with the variable name of the selected leaf node.
- Do **not** include any punctuation, reasoning, quotes, or formatting.
- **Leaf Node** must be **exactly one** variable name as plain text, matching one from the **Unordered Variables** list.
- Do **not** include any additional text before or after the variable name.

Example 1:

Input :

Unordered Variables: ["node_CloudCover", "node_Humidity", "node_Pressure", "node_Temperature"]

Data Description: The dataset contains weather data recorded hourly with multiple atmospheric variables.

Variable Descriptions:

```
[
  "node_CloudCover": "The fraction of the sky covered by clouds.",
  "node_Humidity": "The amount of water vapor in the air.",
  "node_Pressure": "The atmospheric pressure at a given location.",
  "node_Temperature": "The measure of how hot or cold the air is."
]
```

Output : node_Temperature

Input :

Unordered Variables: {unordered_variables}

Data Description: {data_description}

Variable Descriptions: {variable_description}

Output :

Figure A.6: Prompting for Leaf Node Selection

Physics (α (=alpha)=0.5, Control: rank-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'Weight of object': 0.303, 'Moisture Content of object': 0.205, 'Rainfall': 0.144, ..., 'Total Solar Irradiance': 0.071, 'Surface Air Temperature': 0.068}

SciNO Likelihood: {'Weight of object': 0.635, 'Rainfall': 0.233, 'Moisture Content of object': 0.086, ..., 'Wind Speed': 0.003, 'Total Solar Irradiance': 0.002}

Updated Posterior: {'**Weight of object**': 0.776, 'Rainfall': 0.136, 'Moisture Content of object': 0.071, ..., 'Wind Speed': 0.001, 'Total Solar Irradiance': 0.001}

Current Causal Order π_1 : [Weight of object]

=== Causal Ordering Step t=2 ===

LLM Prior: {'Moisture Content of object': 0.288, 'Rainfall': 0.220, 'Rate of Evaporation': 0.219, ..., 'Total Solar Irradiance': 0.086, 'Surface Air Temperature': 0.086}

SciNO Likelihood: {'Rainfall': 0.644, 'Moisture Content of object': 0.214, 'Surface Air Temperature': 0.084, ..., 'Wind Speed': 0.010, 'Total Solar Irradiance': 0.005}

Updated Posterior: {'**Rainfall**': 0.641, 'Moisture Content of object': 0.278, 'Rate of Evaporation': 0.040, ..., 'Wind Speed': 0.005, 'Total Solar Irradiance': 0.002}

Current Causal Order π_2 : [Weight of object, Rainfall]

(....)

=== Causal Ordering Step t=5 ===

LLM Prior: {'Wind Speed': 0.464, 'Total Solar Irradiance': 0.300, 'Surface Air Temperature': 0.235}

SciNO Likelihood: {'Surface Air Temperature': 0.677, 'Wind Speed': 0.218, 'Total Solar Irradiance': 0.104}

Updated Posterior: {'**Surface Air Temperature**': 0.545, 'Wind Speed': 0.346, 'Total Solar Irradiance': 0.107}

Current Causal Order π_5 : [Weight of object, Rainfall, ..., Surface Air Temperature]

=== Causal Ordering Step t=6 ===

LLM Prior: {'Wind Speed': 0.584, 'Total Solar Irradiance': 0.415}

SciNO Likelihood: {'Wind Speed': 0.704, 'Total Solar Irradiance': 0.295}

Updated Posterior: {'**Wind Speed**': 0.769, 'Total Solar Irradiance': 0.230}

Current Causal Order π_6 : [Weight of object, Rainfall, ..., Wind Speed]

=== Final Results ===

Complete Causal Order: [Total Solar Irradiance, Wind Speed, ..., Rainfall, Weight of object]

Order Divergence: 1

Figure A.7: LLM control log on the Physics dataset.

Sachs (α (=alpha)=1.0, Control: CI-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'pakts473': 0.195, 'p44/42': 0.166, 'pjnk': 0.163, ..., 'PIP3': 0.077, 'P38': 0.071}

SciNO Likelihood: {'pmek': 0.125, 'PIP3': 0.125, 'p44/42': 0.125, ..., 'P38': 0.125, 'pjnk': 0.12}

Updated Posterior: {'pakts473': 0.195, 'p44/42': 0.166, 'pjnk': 0.163, ..., 'PIP3': 0.077, 'P38': 0.071}

Current Causal Order π_1 : [pakts473]

=== Causal Ordering Step t=2 ===

LLM Prior: {'p44/42': 0.213, 'PKC': 0.210, 'pjnk': 0.204, ..., 'PIP3': 0.089, 'P38': 0.074}

SciNO Likelihood: {'pmek': 0.142, 'PIP3': 0.142, 'p44/42': 0.143, ..., 'P38': 0.143, 'pjnk': 0.142}

Updated Posterior: {'p44/42': 0.213, 'PKC': 0.210, 'pjnk': 0.204, ..., 'PIP3': 0.089, 'P38': 0.074}

Current Causal Order π_2 : [pakts473, p44/42]

(.....)

=== Causal Ordering Step t=6 ===

LLM Prior: {'P38': 0.496, 'PKA': 0.286, 'PIP3': 0.217}

SciNO Likelihood: {'PIP3': 0.416, 'P38': 0.416, 'PKA': 0.166}

Updated Posterior: {'P38': 0.599, 'PIP3': 0.262, 'PKA': 0.138}

Current Causal Order π_6 : [pakts473, p44/42, ..., P38]

=== Causal Ordering Step t=7 ===

LLM Prior: {'PIP3': 0.605, 'PKA': 0.394}

SciNO Likelihood: {'PIP3': 0.500, 'PKA': 0.500}

Updated Posterior: {'PIP3': 0.605, 'PKA': 0.394}

Current Causal Order π_7 : [pakts473, p44/42, ..., P38, PIP3]

=== Final Results ===

Complete Causal Order: [PKA, PIP3, ..., p44/42, pakts473]

Order Divergence: 3

Figure A.8: LLM control log on the Sachs dataset.

MAGIC-NIAB (α (=alpha)=0.5, Control: rank-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'YR.GLASS': 0.176, 'YLD': 0.142, 'HT': 0.118, ..., 'G1750': 0.019, 'G1276': 0.016}
SciNO Likelihood: {'YR.FIELD': 0.328, 'FT': 0.296, 'YLD': 0.297, ..., 'G1294': 0.000, 'G1276': 0.000}
Updated Posterior: {'YLD': 0.424, 'FT': 0.311, 'YR.FIELD': 0.216, ..., 'G1276': 0.000, 'G1294': 0.000}

Current Causal Order π_1 : [YLD]

=== Causal Ordering Step t=2 ===

LLM Prior: {'FT': 0.161, 'HT': 0.149, 'G43': 0.108, ..., 'G1750': 0.022, 'G800': 0.015}
SciNO Likelihood: {'YR.FIELD': 0.443, 'FT': 0.414, 'FUS': 0.102, ..., 'G1276': 0.000, 'G1263': 0.000}
Updated Posterior: {'FT': 0.712, 'YR.FIELD': 0.171, 'FUS': 0.111, ..., 'G1294': 0.000, 'G1263': 0.000}

Current Causal Order π_2 : [YLD, FT]

(.....)

=== Causal Ordering Step t=42 ===

LLM Prior: {'G418': 0.359, 'G1750': 0.347, 'G1217': 0.292}
SciNO Likelihood: {'G418': 0.685, 'G1750': 0.193, 'G1217': 0.121}
Updated Posterior: {'G418': 0.706, 'G1750': 0.192, 'G1217': 0.101}

Current Causal Order π_{42} : [YLD, FT, ..., G418]

=== Causal Ordering Step t=43 ===

LLM Prior: {'G1750': 0.527, 'G1217': 0.472}
SciNO Likelihood: {'G1217': 0.660, 'G1750': 0.339}
Updated Posterior: {'G1217': 0.635, 'G1750': 0.364}

Current Causal Order π_{43} : [YLD, FT, ..., G418, G1217]

=== Final Results ===

Complete Causal Order: [G1750, G1217, ..., G38, FT, YLD]
Order Divergence: 2

Figure A.9: LLM control log on the MAGIC-NIAB dataset.

ECOLI70 (α (=alpha)=0.5, Control: rank-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'atpG': 0.072, 'asnA': 0.054, 'atpD': 0.054, ..., 'b1191': 0.030, 'yedE': 0.027}
SciNO Likelihood: {'ygbD': 0.742, 'tnaA': 0.193, 'nmpC': 0.030, ..., 'sucD': 0.000, 'yecO': 0.000}
Updated Posterior: {'ygbD': 0.476, 'tnaA': 0.440, 'atpG': 0.058, ..., 'yecO': 0.000, 'sucD': 0.000}

Current Causal Order π_1 : [ygbD]

=== Causal Ordering Step t=2 ===

LLM Prior: {'atpG': 0.060, 'yheI': 0.048, 'asnA': 0.046, ..., 'b1583': 0.033, 'b1191': 0.032}
SciNO Likelihood: {'cspA': 0.720, 'tnaA': 0.274, 'pspA': 0.003, ..., 'dnaG': 0.000, 'yecO': 0.000}
Updated Posterior: {'cspA': 0.579, 'tnaA': 0.418, 'pspA': 0.001, ..., 'dnaG': 0.000, 'icdA': 0.000}

Current Causal Order π_2 : [ygbD, cspA]

(.....)

=== Causal Ordering Step t=44 ===

LLM Prior: {'b1191': 0.359, 'cspG': 0.352, 'eutG': 0.287}
SciNO Likelihood: {'b1191': 0.669, 'eutG': 0.214, 'cspG': 0.115}
Updated Posterior: {'b1191': 0.701, 'eutG': 0.179, 'cspG': 0.118}

Current Causal Order π_{44} : [ygbD, cspA, ..., b1191]

=== Causal Ordering Step t=45 ===

LLM Prior: {'eutG': 0.540, 'cspG': 0.459}
SciNO Likelihood: {'eutG': 0.673, 'cspG': 0.326}
Updated Posterior: {'eutG': 0.708, 'cspG': 0.291}

Current Causal Order π_{45} : [ygbD, cspA, ..., b1191, eutG]

=== Final Results ===

Complete Causal Order: [cspG, eutG, ..., cspA, ygbD]
Order Divergence: 10

Figure A.10: LLM control log on the ECOLI70 dataset.

ARTH150 (α (=alpha)=0.5, Control: CI-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'519': 0.066, '78': 0.044, '93': 0.037, ..., '4': 0.025, '96': 0.024}
SciNO Likelihood: {'496': 0.500, '677': 0.500, '4': 0.000, ..., '47': 0.000, '61': 0.000}
Updated Posterior: {'496': 0.662, '677': 0.337, '4': 0.000, ..., '47': 0.000, '61': 0.000}

Current Causal Order π_1 : [496]

=== Causal Ordering Step t=2 ===

LLM Prior: {'519': 0.073, '78': 0.047, '93': 0.040, ..., '783': 0.026, '539': 0.025}
SciNO Likelihood: {'677': 1.000, '4': 0.000, '8': 0.000, ..., '61': 0.000, '63': 0.000}
Updated Posterior: {'677': 1.000, '4': 0.000, '8': 0.000, ..., '61': 0.000, '63': 0.000}

Current Causal Order π_2 : [496, 677]

(.....)

=== Causal Ordering Step t=105 ===

LLM Prior: {'738': 0.378, '539': 0.311, '783': 0.309}
SciNO Likelihood: {'539': 0.333, '738': 0.333, '783': 0.333}
Updated Posterior: {'738': 0.378, '539': 0.311, '783': 0.309}

Current Causal Order π_{105} : [496, 677, ..., 738]

=== Causal Ordering Step t=106 ===

LLM Prior: {'783': 0.533, '539': 0.466}
SciNO Likelihood: {'539': 0.967, '783': 0.032}
Updated Posterior: {'539': 0.963, '783': 0.036}

Current Causal Order π_{106} : [496, 677, ..., 738, 539]

=== Final Results ===

Complete Causal Order: [783, 539, ..., 677, 496]
Order Divergence: 18

Figure A.12: LLM control log on the ARTH150 dataset.

MAGIC-IRRI (α (=alpha)=0.5, Control: rank-based)

=== Causal Ordering Step t=1 ===

LLM Prior: {'YLD': 0.238, 'HT': 0.070, 'BROWN': 0.069, ..., 'G3092': 0.023, 'AMY': 0.021}
SciNO Likelihood: {'CHALK': 0.664, 'GW': 0.158, 'YLD': 0.134, ..., 'FT': 0.000, 'AMY': 0.000}
Updated Posterior: {'YLD': 0.518, 'CHALK': 0.445, 'BROWN': 0.026, ..., 'AMY': 0.000, 'FT': 0.000}

Current Causal Order π_1 : [YLD]

=== Causal Ordering Step t=2 ===

LLM Prior: {'HT': 0.119, 'CHALK': 0.099, 'BROWN': 0.093, ..., 'G3927': 0.030, 'G3098': 0.024}
SciNO Likelihood: {'CHALK': 0.652, 'GW': 0.232, 'BROWN': 0.068, ..., 'AMY': 0.000, 'GTEMP': 0.000}
Updated Posterior: {'CHALK': 0.866, 'BROWN': 0.084, 'GW': 0.045, ..., 'FT': 0.000, 'HT': 0.000}

Current Causal Order π_2 : [YLD, CHALK]

(.....)

=== Causal Ordering Step t=62 ===

LLM Prior: {'G3925': 0.379, 'G3823': 0.360, 'G4156': 0.260}
SciNO Likelihood: {'G4156': 0.439, 'G3925': 0.284, 'G3823': 0.275}
Updated Posterior: {'G4156': 0.355, 'G3925': 0.335, 'G3823': 0.308}

Current Causal Order π_{62} : [YLD, CHALK, ..., G4156]

=== Causal Ordering Step t=63 ===

LLM Prior: {'G3823': 0.690, 'G3925': 0.309}
SciNO Likelihood: {'G3925': 0.689, 'G3823': 0.310}
Updated Posterior: {'G3823': 0.500, 'G3925': 0.499}

Current Causal Order π_{63} : [YLD, CHALK, ..., G4156, G3823]

=== Final Results ===

Complete Causal Order: [G3925, G3823, ..., CHALK, YLD]
Order Divergence: 9

Figure A.11: LLM control log on the MAGIC-IRRI dataset.

References

- [1] Alessandro Antonucci, Gregorio Piqué, and Marco Zaffalon. Zero-shot causal graph extrapolation from text via LLMs. *arXiv preprint arXiv:2312.14670*, 2023.
- [2] Taiyu Ban, Lyuzhou Chen, Xiangyu Wang, and Huanhuan Chen. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data. *CoRR*, 2023.
- [3] Daniel Bernstein, Basil Saeed, Chandler Squires, and Caroline Uhler. Ordering-based causal structure learning in the presence of latent variables. In *International conference on artificial intelligence and statistics*, pages 4098–4108. PMLR, 2020.
- [4] Peter Bühlmann, Jonas Peters, and Jan Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, pages 2526–2556, 2014.
- [5] Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? *Advances in Neural Information Processing Systems*, 37:96640–96670, 2024.
- [6] David Maxwell Chickering. Learning Bayesian networks is NP-complete. *Learning from data: Artificial intelligence and statistics V*, pages 121–130, 1996.
- [7] Victor-Alexandru Darvari, Stephen Hailes, and Mirco Musolesi. Large Language Models are Effective Priors for Causal Graph Discovery. *arXiv e-prints*, 2024.
- [8] Paul Erdos and Alfred Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5(1):17–60, 1960.
- [9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [10] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.
- [11] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*, 2024.
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [14] Emre Kiciman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal Reasoning and Large Language Models: Opening a New Frontier for Causality. *Transactions on Machine Learning Research*, 2024. Featured Certification.
- [15] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [16] Nicolai Vladimirovich Krylov. *Lectures on Elliptic and Parabolic Equations in Sobolev Spaces*, volume 96. American Mathematical Soc., 2008.
- [17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

- [18] Chanhui Lee, Juhyeon Kim, YongJun Jeong, Yoonseok Yeom, Juhyun Lyu, Jung-Hee Kim, Sangmin Lee, Sangjun Han, Hyeokjun Choe, Soyeon Park, Woohyung Lim, Kyunghoon Bae, Sungbin Lim, and Sanghack Lee. On Incorporating Prior Knowledge Extracted from Pre-trained Language Models into Causal Discovery. In *Causality and Large Models @NeurIPS 2024*, 2024.
- [19] Peiwen Li, Xin Wang, Zeyang Zhang, Yuan Meng, Fang Shen, Yue Li, Jialong Wang, Yang Li, and Wenwu Zhu. RealTCD: temporal causal discovery from interventional data with large language model. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4669–4677, 2024.
- [20] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable Fourier Features for Multi-Dimensional Spatial Positional Encoding. *Advances in Neural Information Processing Systems*, 34:15816–15829, 2021.
- [21] Yingzhen Li and Richard E Turner. Gradient Estimators for Implicit Models. In *International Conference on Learning Representations*, 2018.
- [22] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021.
- [23] Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023.
- [24] Sungbin Lim, Eunbi Yoon, Taehyun Byun, Taewon Kang, Seungwoo Kim, Kyungjae Lee, and Sungjoon Choi. Score-based Generative Modeling through Stochastic Evolution Equations in Hilbert Spaces. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [25] Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal Discovery with Language Models as Imperfect Experts. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.
- [26] Rainer Opge-Rhein and Korbinian Strimmer. From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC systems biology*, 1:1–10, 2007.
- [27] Jonas Peters and Peter Bühlmann. Structural Intervention Distance for Evaluating Causal Graphs. *Neural Computation*, 27(3):771–799, 03 2015.
- [28] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.
- [29] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1):2009–2053, 2014.
- [30] Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.
- [31] 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.
- [32] 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.
- [33] Pedro Sanchez, Xiao Liu, Alison Q O’Neil, and Sotirios A Tsaftaris. Diffusion Models for Causal Discovery via Topological Ordering. In *The Eleventh International Conference on Learning Representations*, 2023.

- 383 [34] Juliane Schäfer and Korbinian Strimmer. A shrinkage approach to large-scale covariance matrix
384 estimation and implications for functional genomics. *Statistical applications in genetics and*
385 *molecular biology*, 4(1), 2005.
- 386 [35] M Scutari. Bayesian networks, magic populations and multiple trait prediction. In *Invited Talk*
387 *at the 5th International Conference on Quantitative Genetics (ICQG 2016)*, 2016.
- 388 [36] Marco Scutari, Phil Howell, David J Balding, and Ian Mackay. Multiple quantitative trait
389 analysis using Bayesian networks. *Genetics*, 198(1):129–137, 2014.
- 390 [37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and
391 Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In
392 *International Conference on Learning Representations*, 2021.
- 393 [38] Ioannis Tsamardinos, Laura Brown, and Constantin Aliferis. The Max-Min Hill-Climbing
394 Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65:31–78, 10 2006.
- 395 [39] Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N.
396 Balasubramanian, and Amit Sharma. Causal Order: The Key to Leveraging Imperfect Experts
397 in Causal Inference. In *The Thirteenth International Conference on Learning Representations*,
398 2025.
- 399 [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
400 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information*
401 *processing systems*, 30, 2017.
- 402 [41] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang
403 Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine
404 translation system: Bridging the gap between human and machine translation. *arXiv preprint*
405 *arXiv:1609.08144*, 2016.
- 406 [42] Zhuopeng Xu, Yujie Li, Cheng Liu, and Ning Gui. Ordering-Based Causal Discovery for
407 Linear and Nonlinear Relations. In *The Thirty-eighth Annual Conference on Neural Information*
408 *Processing Systems*, 2024.