TIME-SERIES CAUSAL DISCOVERY VIA DIFFERENTIABLE PERMUTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal discovery with instantaneous effects in multivariate time series is challenging, as the instantaneous structure must be acyclic. Prior methods enforce this by either recovering a causal order through discrete combinatorial search or imposing algebraic acyclicity constraints via complex augmented Lagrangian optimization, both of which incur high computational cost. In this work, we propose a fundamentally different method: we learn a differentiable permutation of variables using the Gumbel–Sinkhorn operator and triangularize the instantaneous coefficient matrix of a Structural Vector Autoregressive (SVAR) model in learned order. This converts acyclicity from a hard constraint into a parameterization and keeps it valid throughout optimization. In doing so, our method enables unified, continuous optimization with gradient-based learning, leading to improved efficiency in time series causal discovery. Across three real-world benchmarks, our method achieves the best overall performance compared with 12 baselines in both discovery accuracy and efficiency. On the large-scale benchmark, it further demonstrates strong scalability, achieving more than a 6× speedup over competing methods.

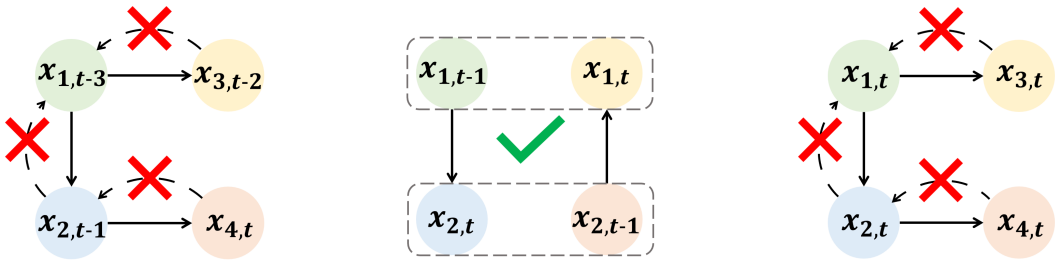
1 INTRODUCTION

Time-series causal discovery helps recover cause–effect relationships in dynamical systems, and is widely applied in diverse fields such as economics (Hoover, 2001), earth science (Runge et al., 2019a), and industrial systems (Mogensen et al., 2024). As noted by Assaad et al. (2022b), true causal discovery methods for time series should account for both instantaneous causal effects, where $x_{i,t}$ affects $x_{j,t}$ within the same step; and lagged causal effects, where a past state $x_{i,t-\tau}$ (with $\tau > 0$) influences a future state $x_{j,t}$. These relationships are typically formalized as directed graphs. The main goal of causal discovery is to build a causal graph from observed data (Assaad et al., 2022b).

To ensure validity, the causal graph needs to satisfy acyclic constraints (as shown in Figure 1). In particular, for lagged effects, one could easily avoid cycles by avoiding causal links from the future to the past. For instantaneous effects, however, cycles may occur if two variables affect each other within the same time step. To avoid this, instantaneous effects are typically constrained to form a Directed Acyclic Graph (DAG) to guarantee identifiability of the causal structure (Kilian, 2006).

The central challenge in time-series causal discovery is therefore enforcing acyclicity on instantaneous effects. Among existing methods that explicitly enforce acyclicity, two main strategies exist: (1) discrete combinatorial search for a causal order, followed by estimation of causal effects based on the obtained order, as in VARLiNGAM (Hyvärinen et al., 2010) and TiMINo (Peters et al., 2013); and (2) algebraic acyclicity constraints enforced by an augmented Lagrangian optimization while estimating causal effects, as in DYNOTEARS (Pamfil et al., 2020).

While effective, these approaches have notable limitations: (1) They rely on hard acyclicity constraints: discrete order search fixes a causal order upfront that cannot adapt even if it poorly fits the data; and augmented Lagrangian methods enforce strict algebraic constraints that require constraint optimization until exact satisfaction. (2) This rigidity forces a multi-stage process, separating acyclicity enforcement from causal-effect estimation and thereby risking error propagation, as discussed in (Pamfil et al., 2020). (3) They incur high computational cost: combinatorial search grows exponentially, and augmented Lagrangian optimization involves nested loops with unpredictable iterations before the constraint is met. These limitations motivate the need for more flexible, unified methods that efficiently enforce instantaneous acyclicity and scale to high-dimensional data.



(a) Lagged acyclicity (b) Valid temporal loop (c) Instantaneous acyclicity

Figure 1: Illustration of acyclicity in time-series causal graphs, where nodes denote variables and arrows indicate directional influences between variables. (a) Lagged acyclicity: cycles cannot occur across lagged dependencies because future states cannot cause the past. (b) Valid temporal loop: feedback loops across time are allowed as long as all arrows follow temporal order. (c) Instantaneous acyclicity: instantaneous cycles within the same time step are forbidden to ensure identifiability.

Contributions We propose a new time-series causal discovery method based on a Structural Vector Autoregressive (SVAR) model (Swanson & Granger, 1997; Demiralp & Hoover, 2003; Moneta & Spirtes, 2006), where causal structure learning is reformulated as the problem of fitting SVAR coefficients, from which the causal graph can be directly constructed. Our main contributions include:

- **Soft acyclicity** (see Section 4.3). We cast acyclicity as a permutation learning problem and design a Gumbel–Sinkhorn operator to relax the permutation into a differentiable form. This enables the causal order to be learned adaptively during optimization rather than fixed upfront, making it soft, data-driven, and dynamically adaptive to observed time series.
- **Unified and scalable optimization** (see Sections 4.1–4.2). Reparameterizing acyclicity with a learnable permutation matrix enables joint optimization where both acyclicity enforcement and causal effect estimation are handled in a single stage. This avoids constrained optimization and the need for multi-stage procedures used in existing methods, and allows gradient-based optimizers to be applied directly to the entire learning problem, leading to improved efficiency and scalability for large-scale settings.
- **Extensive evaluation on real-world data** (see Section 5). We evaluate on three real-world benchmarks covering 11 datasets: IT monitoring (Aıt-Bachir et al., 2023), SWaT (Maiti et al., 2023), and CausalRiver (Gideon et al., 2025). Across all datasets, our method achieves the best overall performance compared with 12 baselines in both discovery accuracy and efficiency. On CausalRiver, the largest benchmark to date, our method achieves substantially higher accuracy with over 6× speedup compared to competing methods, demonstrating strong scalability to high-dimensional data.

2 Related Work

Many methods have been developed for time-series causal discovery (Gong et al., 2024). Some approaches ignore instantaneous dependencies and recover only lagged relationships, most notably Granger causality (Granger, 1969), which relies on VAR model (Sims, 1980). Examples include MVGC (Barrett et al., 2010; Barnett & Seth, 2014), TCDF (Nauta et al., 2019), and neural Granger causality (Tank et al., 2021). Constraint-based methods, in contrast, do not enforce acyclicity on instantaneous effects explicitly but rather handle it implicitly through conditional independence tests, such as tsFCI (Entner & Hoyer, 2010; Gerhardus & Runge, 2020), PCMCI+ (Runge et al., 2019b; Runge, 2020), and PCGCE (Assaad et al., 2022a). Since our focus is time-series causal discovery with instantaneous effects for theoretical completeness (following the argument of Assaad et al. (2022b) that true causal discovery should consider both instantaneous and lagged effects), we do not review above methods in detail, but include them as baselines in our experiments for comparison.

Our main focus is on time-series causal discovery methods that explicitly address the enforcement of acyclicity on instantaneous effects, most of which extend static causal discovery to temporal settings. Broadly, two families can be distinguished depending on how acyclicity is enforced.

The first family relies on discrete combinatorial search for a fixed causal order, followed by causal-effect estimation. This includes two noise-based methods. First, VARLiNGAM (Hyvärinen et al., 2010) extends LiNGAM (Shimizu et al., 2011) by combining a VAR with a non-Gaussian instantaneous model. Identifiability is achieved under the assumption of non-Gaussian errors (Shimizu et al., 2006), with acyclicity enforced through Independent Component Analysis (ICA) (Lee, 1998). Second, TiMINo (Peters et al., 2013) generalizes structural equation models to time series, using nonlinear additive noise models (Mooij et al., 2009) and residual-independence tests to iteratively identify sources and build an acyclic order. These methods are multi-stage, sensitive to order-estimation errors, and can have exponential complexity in discrete combinatorial search.

The second family enforces acyclicity by formulating it as an algebraic constraint within a constrained optimization problem. Proposed as a score-based method, DYNOTEARS (Pamfil et al., 2020) adapts NOTEARS (Zheng et al., 2018) to dynamic settings, jointly estimating causal effects within an SVAR formulation. It imposes acyclicity via the smooth equality constraint $h(W) = \mathrm{tr}(e^{W \circ W}) - d = 0$, where W is the weighted instantaneous effects and d is the number of variables. This constraint is enforced through a complex augmented Lagrangian method (Nemirovsky, 1999). While this allows joint estimation of intra- and inter-slice effects, the augmented Lagrangian iterations are computationally demanding and unpredictable in large-scale settings.

In response to the limitations of existing acyclicity handling methods, we propose a new time-series causal discovery approach built on differentiable permutation learning (Maddison et al., 2017; Jang et al., 2017; Mena et al., 2018), particularly the Gumbel–Sinkhorn relaxation. This relaxation maps hard permutations to differentiable doubly-stochastic matrices for gradient-based learning, with hard permutations recovered at inference. Although related work on differentiable DAG sampling (Charpentier et al., 2022) exists, our method fundamentally differs: (i) their approach targets static DAG learning, while we address time-series structure; and (ii) they rely on Gumbel-Top-k with straight-through sampling inside a probabilistic framework, whereas we employ a continuous, learnable permutation that reparameterizes the causal order and preserves acyclicity throughout optimization.

3 Problem Setup

While the objective of time-series causal discovery is to construct a valid causal graph that captures both instantaneous and lagged dependencies among variables, the SVAR model reframes this task as estimating the model coefficients that best explain the observed data, since each coefficient matrix B_τ corresponds to a subgraph. Formally, the SVAR model is given by:

$$\mathbf{x}_t = \sum_{\tau=0}^{k} B_\tau \mathbf{x}_{t-\tau} + \boldsymbol{\epsilon}_t, \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the vector of observed variables at time t. k is the maximum lag order considered. $B_\tau \in \mathbb{R}^{d \times d}$ is the coefficient matrix at lag τ: B_0 encodes instantaneous effects, while B_τ for $\tau > 0$ encodes lagged effects. $\boldsymbol{\epsilon}_t$ is the vector of error terms at time t.

The linear structure of SVAR offers strong interpretability. Its coefficients not only reveal the presence of causal relationships but also quantify their strength and direction (positive or negative). This provides richer information than methods that merely identify links and further enables direct modeling of system dynamics. For this reason, we adopt the linear SVAR model as our foundation.

Identifiability Identifiability is a central issue in SVAR models (Pamfil et al., 2020). For lagged effects, identification follows from standard VAR assumptions. In contrast, instantaneous effects are harder, since covariance information alone is insufficient to uniquely identify B_0 (Hyvärinen et al., 2010). Two common sufficient conditions under independent $\boldsymbol{\epsilon}_t$ are: (i) non-Gaussian noise, yielding identifiability via ICA/Marcinkiewicz arguments (Hyvärinen et al., 2010; Lanne et al., 2017); and (ii) Gaussian noise with equal error variances (e.g., standard Gaussian), under which the DAG is identifiable, together with acyclicity of the instantaneous graph (Peters & Bühlmann, 2014). Following Pamfil et al. (2020), we assume at least one of these conditions holds.

4 METHODOLOGY

4.1 UNIFIED OPTIMIZATION

As discussed in Section 3, fitting the SVAR model reduces to estimating coefficient matrices that best explain the observed data. We thus define our optimization objective $l(\tilde{B}_0, \{B_\tau\})$ as the mean squared error (MSE) of the SVAR residuals, which is equivalent to maximizing the data likelihood:

$$\min_{\tilde{B}_0, \{B_\tau\}_{\tau=1}^k} l(\tilde{B}_0, \{B_\tau\}) = \frac{1}{N} \sum_{t=k+1}^{T} \left\| (I - \tilde{B}_0)\mathbf{x}_t - \sum_{\tau=1}^{k} B_\tau \mathbf{x}_{t-\tau} \right\|_2^2, \tag{2}$$

where T is the total number of time points, $N = T - k$ is the number of effective samples, and $\|\cdot\|_2^2$ denotes the squared L_2-norm. Crucially, the optimization is performed over $\tilde{B}_0$, the instantaneous effects matrix constrained to be acyclic. This matrix is derived from an unconstrained matrix B_0 with the Gumbel–Sinkhorn technique (detailed in Section 4.3).

Since causal structures are typically sparse, L_1 regularization and its variants are commonly employed as sparsity penalties (Hyvärinen et al., 2010). We adopt standard ℓ_1-regularization, consistent with prior work (Pamfil et al., 2020). This choice isolates the impact of our proposed acyclicity method from the effects of more advanced sparsity mechanisms. The final penalized objective is:

$$\min_{\tilde{B}_0, \{B_\tau\}_{\tau=1}^k} f(\tilde{B}_0, \{B_\tau\}) = l(\tilde{B}_0, \{B_\tau\}) + \lambda_0 \|\tilde{B}_0\|_1 + \lambda_\tau \sum_{\tau=1}^{k} \|B_\tau\|_1, \tag{3}$$

where $\|\cdot\|_1$ is the element-wise L_1-norm and λ_0, λ_τ are penalty hyperparameters.

Direct gradient-based optimization Though this form of optimization objective has been applied to causal discovery before (Pamfil et al., 2020), it has so far been used as constrained optimization solved with nested updates, which makes it impossible to apply a direct gradient-based optimizer to the entire problem. Our method, however, reparameterizes the acyclicity enforcement (see Section 4.3), resolving this issue and relaxing the problem into a unified and unconstrained optimization. This enables direct gradient-based optimization of the whole objective, which is theoretically more efficient than constrained approaches in this setting (Nocedal & Wright, 2006; Jaggi, 2013).

4.2 THEORETICAL JUSTIFICATION

The theoretical justification for equation 3 is supported by Aragam et al. (2015). They showed that penalized least-squares estimators with concave penalties achieve uniform support recovery and deviation bounds when $n \gtrsim d \log p$ (where n is the sample size, d the maximum in-degree, and p the number of variables), and that the global minimizer is statistically consistent with exact edge recovery under a beta-min condition. This provides a strong foundation for our choice of penalized least-squares. While the theoretical support applies mainly to Gaussian settings, empirical studies further show that squared loss also works well in non-Gaussian cases (Pamfil et al., 2020).

We acknowledge that minimizing residual noise does not always guarantee recovery of the true causal structure, which creates a potential mismatch between the optimization objective and the underlying mechanism. The idea here is that, under correct model specification, unexplained variation should ideally exhibit "clean" properties, such as relative smallness. By minimizing residual error subject to causal constraints (Section 4.3), the SVAR model captures the main causal structure, while weak or ambiguous links can be pruned in post-processing. Despite this subtle inconsistency, penalized least-squares remains a valid and statistically consistent estimation strategy.

4.3 ACYCLICITY ENFORCEMENT VIA DIFFERENTIABLE PERMUTATION

The acyclicity of B_0 is satisfied if there exists a permutation matrix P such that $PB_0P^\top$ is (close to) strictly lower triangular (Hyvärinen et al., 2010). Thus, enforcing acyclicity on instantaneous effects can be framed as finding a permutation that triangularizes B_0 while still fitting the data well. However, permutation matrices are discrete and therefore incompatible with the continuous optimization in equation 3. The key challenge is to integrate this combinatorial choice into a differentiable framework. To address this, we employ the Gumbel–Sinkhorn technique (Mena et al., 2018), which learns a continuous parameterization that yields a differentiable approximation of P.

4.3.1 GUMBEL–SINKHORN RELAXATION FOR CAUSAL ORDERING

The Gumbel–Sinkhorn method provides a differentiable relaxation of permutation matrices, enabling gradients to backpropagate through them (Mena et al., 2018). We develop a Gumbel–Sinkhorn method to learn the causal order as follows: we first introduce a learnable matrix of unconstrained real-valued logits $\Lambda \in \mathbb{R}^{d\times d}$, which encodes the model's preference over variable orderings. To encourage exploration during training, we apply the "Gumbel trick" by adding a noise matrix $\mathbf{G}$ with i.i.d. Gumbel entries, forming perturbed scores $\Lambda + \mathbf{G}$. The perturbed logits are then temperature-scaled:

$$X = \frac{\Lambda + \mathbf{G}}{\tau_{\text{temp}}}, \tag{4}$$

where τ_{temp} is a temperature parameter: higher values yield smoother distributions for stable early training, while lower values sharpen the matrix toward a discrete permutation.

We then apply the Sinkhorn operator $\mathcal{S}(\cdot)$, which iteratively normalizes the rows and columns of a positive matrix until convergence to a doubly stochastic matrix (where both rows and columns sum to 1), i.e. a continuous relaxation of a permutation. The procedure is defined as:

$$\mathcal{S}^0(X) = \exp(X), \tag{5}$$

$$\mathcal{S}^l(X) = \mathcal{T}_c(\mathcal{T}_r(\mathcal{S}^{l-1}(X))), \quad l \geq 1, \tag{6}$$

$$\mathcal{S}(X) = \lim_{l\to\infty} \mathcal{S}^l(X), \tag{7}$$

where exponentiation in $\mathcal{S}^0(X)$ ensures strictly positive entries, a requirement for the normalization steps, and $\mathcal{T}_r$, $\mathcal{T}_c$ denote row and column normalization, respectively. In practice, this limit is approximated with a fixed number of iterations L. The resulting soft permutation matrix $\tilde{P}$ is:

$$\tilde{P} = \mathcal{S}\left(\frac{\Lambda + \mathbf{G}}{\tau_{\text{temp}}}\right), \tag{8}$$

which converges to a discrete permutation as $\tau_{\text{temp}} \to 0$, but remains differentiable for $\tau_{\text{temp}} > 0$.

Recovery of Hard Permutation During training, the relaxed permutation $\tilde{P}$ remains differentiable, which allows gradients to flow through the optimization. However, causal discovery requires a hard causal ordering. Therefore, we recover a hard permutation matrix by projecting $\tilde{P}$ onto the nearest discrete permutation. In practice, this can be done by taking the row- and column-wise $\arg\max$ after the final Sinkhorn iteration, yielding a valid P. This ensures that the final learned causal order corresponds to a discrete DAG structure, while the relaxation only serves to enable backpropagation during optimization.

4.3.2 ENFORCING ACYCLICITY

The differentiable permutation matrix $\tilde{P}$ enables acyclicity enforcement on B_0 directly within the optimization loop. Starting from the unconstrained instantaneous effects B_0, we first permute into the learned causal order:

$$B_0' = \tilde{P} B_0 \tilde{P}^\top. \tag{9}$$

Then apply a strictly lower-triangular mask:

$$B_0'' = \text{tril}(B_0', -1), \tag{10}$$

which zeros the diagonal and upper-triangular entries to enforce acyclicity in the permuted space.

Finally, map back to the original variable order,

$$\tilde{B}_0 = \tilde{P}^\top B_0'' \tilde{P}, \tag{11}$$

which yields the acyclicity-constrained instantaneous matrix $\tilde{B}_0$ used in the objective $f(\tilde{B}_0, \{B_\tau\})$ (Eq. 3). By embedding these operations in the forward pass, the DAG constraint becomes fully differentiable, enabling true end-to-end learning of the causal structure.

5 EVALUATION

5.1 EXPERIMENTAL SETTINGS

5.1.1 DATASETS

As noted by Reisach et al. (2021), synthetic data is easy to game and may not reflect a method's true performance. We therefore exclude synthetic benchmarks and focus on three real-world benchmarks: the small-scale IT Monitoring Aıt-Bachir et al. (2023), the medium-scale SWaT Maiti et al. (2023), and the large-scale CausalRivers Gideon et al. (2025). An overview is provided in Table 1.

Table 1: Overview of benchmarks and datasets used in our experiments. SR stands for sampling rate, #D stands for number of time series data, and #V stands for number of variables.

Benchmark	Origin	Dataset	Scenario	SR	#D	#V
IT Monitoring	UAI2023	MOM 1	Message-Oriented Middleware	1 sec	288	7
		MOM 2			364	7
		Storm	Storm ingestion topology	1 min	991	8
		Web 1	Web server activity	5 mins	7500	10
		Web 2			7501	10
		AntiV 1	Antivirus impact on server	5 mins	1320	13
		AntiV 2			1321	13
SWaT	Arxiv2023	SWaT	Water treatment	1 sec	≈97k	51
CausalRiver	ICLR2025	Flood	River discharge (Elbe Flood)	15 min	3010	42
		Bavaria	River discharge (Bavaria)	15 min	≈175k	494
		Germany	River discharge (East Germany)	15 min	≈175k	666

5.1.2 BASELINES

For comparison, we use 12 time-series causal discovery methods as baselines, summarized in Table 2, selected for their relevance and prevalence in prior studies. Although VAR is not a strict causal discovery method, we include it as a classical and widely adopted baseline. Most baselines do not explicitly handle instantaneous effects (see Section 2); notable exceptions are DYNOTEARS (Pamfil et al., 2020), VARLiNGAM (Hyvärinen et al., 2010), and TiMINo (Peters et al., 2013), which are directly comparable to our method and are of particular interest. We also include TCDF (Nauta et al., 2019) and three neural Granger causality variants (Tank et al., 2021) as representatives of recent deep learning–based approaches for nonlinear, high-dimensional causal discovery.

Table 2: Baselines used in our experiments.

Type	Methods
Traditional	VAR (Sims, 1980), tsFCI (Entner & Hoyer, 2010), VARLiNGAM (Hyvärinen et al., 2010), TiMINO (Peters et al., 2013), MVGC (Barnett & Seth, 2014)
State-of-the-art	TCDF (Nauta et al., 2019), DYN0TEARS (Pamfil et al., 2020), PCMCI+ Runge (2020), Neural Granger causality variants (cMLP, cLSTM, cRNN) (Tank et al., 2021), PCGCE Assaad et al. (2022a)

5.1.3 METRICS

We evaluate performance using Precision, Recall, and F1-Score, following prior work Assaad et al. (2022b); Aıt-Bachir et al. (2023); Nauta et al. (2019), and report F1-Score in this section. While some studies use AUC-ROC (Gideon et al., 2025), this metric can remain high under severe class imbalance, which is a common issue in causal discovery. In contrast, F1-Score is stricter and directly balances false positives and false negatives, making it more suitable for evaluation of this task.

5.1.4 PARAMETER SETTING

Experiments were conducted on an Intel Core i7-14700K (20 cores, 28 threads, 3.4 GHz, 33 MB cache) and 128GB of DDR5-5600 memory. The CPU was used for fair comparison, as some baselines only support CPU execution, and all methods employ multithreaded implementations. Runtimes were limited to 3 hours per run; longer runs are reported as TLE (Time Limit Exceeded). Baselines were executed with default parameters. Maximum lags of 3, 5, 10, and 15 were used as in Aıt-Bachir et al. (2023), excluding TCDF which infers lags automatically. VAR-LiNGAM applied its built-in pruning, while DYNOTEARS, VAR, and our method pruned coefficients below 0.01 Assaad et al. (2022b); Aıt-Bachir et al. (2023). Sparsity penalties λ_{B_0} and λ_{B_τ} were fixed at 0.001 Pamfil et al. (2020). Models were trained with Adam (lr = 0.002) for up to 6000 epochs with early stopping, using 20 Sinkhorn iterations Mena et al. (2018).

5.2 PERFORMANCE ANALYSIS

In this section, we report the F1 scores of all methods across three benchmarks. The results on the IT Monitoring benchmark with a maximum lag of 3 are shown in Table 3, while the results on the SWaT and CausalRiver benchmarks with maximum lags of 5 and 10 are presented in Table 4. Additional F1 score results for other lag settings are provided in Appendix A.3, with the corresponding Recall and Precision results given in Appendix A.5 and Appendix A.6, respectively.

Table 3: F1 scores on IT Monitoring with a maximum lag of 3. For each dataset, the highest and second-highest scores are highlighted in dark green and light green, respectively. The last two columns report the frequency with which each method achieves the highest F1 score (wins, W) and the second-highest score (runner-ups, R), with the highest frequency also marked in light green.

Method	**MoM 1**	**MoM 2**	**Ingestion**	**Web 1**	**Web 2**	**AntiV 1**	**AntiV 2**	**W**	**R**
VAR	0.1667	0.1622	0.2449	0.2222	0.2895	0.2047	0.2188	0	1
MVGC	0.0909	0.0000	0.0667	0.2059	0.2400	0.1613	0.1374	0	0
cLSTM	0.3390	0.3390	0.2500	0.2321	0.2500	0.1739	0.1739	0	2
cMLP	0.2857	0.1622	0.1379	0.1905	0.1081	0.2247	0.2857	0	0
cRNN	0.1714	0.1500	0.2069	0.1778	0.1667	0.2273	0.2903	2	0
TCDF	0.0000	0.0000	0.0000	0.0000	0.1053	0.1818	0.2000	0	0
PCMCI+	0.0000	0.0000	0.0000	0.3243	0.1579	0.0357	0.0833	1	0
tsFCI	0.2286	0.0870	0.0000	0.1818	0.1714	0.1852	0.1159	0	0
PCGCE	0.0909	0.1538	0.1935	0.2143	0.1961	0.2250	0.2400	0	2
DYNOTEARS	0.2857	0.2353	0.1538	0.2623	0.2895	0.1905	0.2056	0	2
VARLiNGAM	0.0000	0.0000	0.3846	0.2593	0.2667	0.1923	0.2188	1	0
TiMINO	0.1538	0.1818	0.0000	0.0000	0.0000	0.0000	0.0000	0	0
Our Method	0.4000	0.3415	0.2769	0.3243	0.3243	0.1830	0.1875	4	1

Table 4: F1 scores on SWaT and CausalRiver with maximum lags of 5 and 10. The last two columns report the counts of wins (W) and runner-ups (R). TLE: Time Limit (3 hours) Exceeded.

Method	**Lag = 5**				**Lag = 10**				**W**	**R**
	SWaT	**Flood**	**Bavaria**	**Germany**	**SWaT**	**Flood**	**Bavaria**	**Germany**		
VAR	0.0565	0.0607	0.0059	0.0045	0.0539	0.0574	0.0056	0.0041	0	4
MVGC	0.0767	0.0559	TLE	TLE	0.0787	0.0572	TLE	TLE	0	0
cLSTM	0.0279	0.0465	TLE	TLE	0.0000	0.0466	TLE	TLE	0	0
cMLP	0.0902	0.1203	TLE	TLE	0.0833	0.1158	TLE	TLE	0	2
cRNN	0.0788	0.0952	TLE	TLE	0.0729	0.1009	TLE	TLE	0	0
TCDF	0.0000	0.0000	TLE	TLE	0.0000	0.0000	TLE	TLE	0	0
PCMCI+	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	0	0
tsFCI	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	0	0
PCGCE	0.0648	0.1116	TLE	TLE	0.0827	0.0813	TLE	TLE	0	0
DYNOTEARS	0.0206	0.1342	TLE	TLE	0.0215	0.1333	TLE	TLE	0	2
VARLiNGAM	0.0660	0.0403	TLE	TLE	0.0621	0.0362	TLE	TLE	0	0
TiMINo	0.0580	0.0000	TLE	TLE	0.0435	0.0000	TLE	TLE	0	0
Our Method	0.2202	0.3000	0.1751	0.1351	0.2182	0.3226	0.1860	0.1388	8	0

On the small-scale IT Monitoring datasets (Table 3), our method achieves the highest F1 score on four out of seven datasets and remains competitive on Ingestion, yielding the largest overall count of wins and runner-ups (5/7). This demonstrates consistent performance, whereas other methods show strong dataset-specific biases—for example, VARLiNGAM excels on Ingestion but fails on MoM1/2, DYNOTEARS performs well on Web datasets, and cRNN and PCGCE on AntiVirus. None, however, achieve robust performance across most datasets as our method does.

On AntiV1/AntiV2 our method perform less strongly. This could be due to the sensitivity of pruning thresholds when applied to these antivirus datasets, which contain sparse and irregular dynamics arising from mixed sampling rates, partial sleeping series, and interpolation steps (Aıt-Bachir et al., 2023). Under these conditions, the fixed pruning threshold we used can be fragile, where small but meaningful effects hard to distinguish from noise, while bursty fluctuations can be retained. As shown in Appendix A.4, tuning the threshold improves results, which points to a methodological limitation of sensitivity to pruning and motivates adaptive or stability-based pruning strategies.

On the medium-scale SWaT and Flood datasets and the large-scale Bavaria and Germany datasets, our method shows a clear advantage. In all eight cases reported in Table 4, it achieves the best performance, with F1 scores often more than double those of the second-best method. This dominance also holds across other lag settings, as shown in Appendix A.3. On large-scale datasets, our method consistently finishes within the 3-hour time limit, while most baselines fail to complete, and still delivers the highest F1 scores. Moreover, performance remains stable across lag values, indicating that even as lag doubles, our method captures the key causal links without introducing excessive spurious ones. Interestingly, despite the reputation of deep learning methods for handling large data, approaches such as TCDF, cLSTM, cRNN, and cMLP do not show clear advantages in causal discovery. Finally, we observe that performance tends to decline as the number of variables grows, which indicates the intrinsic difficulty of large-scale causal discovery.

5.3 Computational Efficiency

In this section, we report the runtime (in seconds) of each method on all datasets with maximum lag set to 15, as shown in Table 5, where the fastest and second-fastest methods are highlighted. Results for lags 3, 5, and 10 are provided in Appendix A.7. Although VAR is included as a baseline for accuracy, we exclude it in this section. Unlike other methods, which aim at causal discovery (with or without explicit acyclicity enforcement), VAR doesn't attempt to discover the true causal structure; it only encodes dependencies over time. Its consistently shorter runtimes therefore reflect solving a simpler problem, and including them would give a misleading impression of efficiency.

Table 5: Runtime (in seconds) with maximum lag of 15. TLE: Time Limit (3 hours) Exceeded. VAR is excluded from comparison because it does not consider instantaneous causal effects.

Method	MoM 1	MoM 2	Storm	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	Germany
VAR (Ref. Only)	0.02	0.03	0.03	0.06	0.06	0.06	0.07	0.79	0.60	82.69	151.38
MVGC	0.30	0.36	0.24	2.00	3.48	10.35	10.75	3632.96	4842.76	TLE	TLE
cLSTM	32.09	39.31	61.60	199.44	221.08	131.27	125.94	TLE	727.86	TLE	TLE
cMLP	12.45	12.58	14.55	43.48	54.03	35.85	35.80	836.62	277.24	TLE	TLE
cRNN	22.15	24.26	34.21	216.56	212.54	107.78	117.74	6933.72	1612.64	TLE	TLE
TCDF	8.69	9.02	9.94	58.36	58.55	16.65	17.81	550.15	258.03	TLE	TLE
PCMCI+	0.51	0.89	4.50	145.98	61.36	34.09	38.51	TLE	TLE	TLE	TLE
tsFCI	1252.319	1232.84	2.70	4649.73	TLE	TLE	TLE	TLE	TLE	TLE	TLE
PCGCE	0.33	0.68	1.48	10.10	5.62	16.41	8.78	725.87	86.29	TLE	TLE
DYNOTEARS	8.36	4.92	0.39	64.06	403.92	85.21	490.66	4641.99	795.53	TLE	TLE
VARLINGAM	0.35	0.38	0.69	9.23	9.60	2.22	2.14	5077.79	54.87	TLE	TLE
TiMINo	1.34	1.99	2.51	16.68	12.77	8.23	6.22	770.68	91.03	TLE	TLE
Our Method	7.25	5.62	6.99	9.97	13.99	7.54	8.56	64.29	20.50	1915.20	3240.05

On the first seven small-scale datasets (from MoM 1 to AntiV 2), our method is not the fastest. Traditional methods such as VAR, MVGC, and VARLiNGAM have lower runtimes. However, our method consistently completes its analysis in under 15 seconds in all cases. This performance is highly competitive and often faster than many complex models like cLSTM, cRNN, and tsFCI. In practical applications, the minor difference of a few seconds is negligible.

The true advantages of our method become obvious when applied to medium and large-scale datasets. On the SWaT dataset, for instance, our method is over 8 times faster than the next-best com-

petitor, TCDF. Since a core focus of this work is efficient acyclicity enforcement, a direct comparison with methods like VARLiNGAM, DYNOTEARS, and TiMINo is most telling. On SWaT, our algorithm is approximately 79 times faster than VARLiNGAM, 72 times faster than DYNOTEARS, and 12 times faster than TiMINo. This performance gap is also evident on the Flood dataset, where our method is nearly 39 times faster than DYNOTEARS.

The most compelling evidence of our method's scalability is observed on the large-scale Bavaria and Germany datasets. On these complex, high-dimensional tasks, all competing methods, including all those that explicitly enforce acyclicity, failed to produce a result within the 3-hour time limit. In contrast, our method successfully completed the analysis in approximately 32 minutes for Bavaria and 54 minutes for Germany. This indicates our method is at least 6 times faster on Bavaria and 3 times faster on Germany than the other leading acyclicity-enforcing methods.

6 Discussion

Limitations Our work addresses acyclicity enforcement on instantaneous effects and demonstrates its improved accuracy and efficiency in Section 5, but several limitations remain. We did not analyze the convergence or potential sub-optimality of the Gumbel–Sinkhorn relaxation beyond empirical evidence, which could provide deeper insight. While we only examined lag choices, systematic exploration of optimal lags would be useful. Our analysis of sensitivity to pruning thresholds was limited, and more detailed experiments could strengthen this evidence. We also did not study the impact of sampling rates, which may affect results and reveal temporal structure. In addition, our evaluation focused on runtime efficiency without considering memory or cache usage. Finally, although the method supports gradient-based optimization, our experiments were limited to CPU execution, while hardware acceleration on GPUs or FPGAs could further improve efficiency. We acknowledge these limitations and leave a more detailed investigation to future work.

Causality and Instantaneous Effects As shown in Section 5.2, the VAR model, though not designed for causal discovery, can still perform competitively, highlighting that much of what is termed "causal discovery" remains statistical estimation of associations rather than identification of true causal mechanisms. While causality in principle concerns directional influence and interventions (Pearl & Mackenzie, 2018), current approaches, including ours, rely on observational data and probabilistic models that cannot fully resolve this distinction. Moreover, the notion of "instantaneous effects" often reflects the compression of interactions within the sampling interval rather than true immediate causal influences. Such effects may therefore represent aggregated lagged dependencies, with potential biases tied to sampling frequency. Without accounting for these artifacts, there remains a risk of mistaking statistical regularities for genuine causal mechanisms.

7 Conclusion

A major challenge in time-series causal discovery is enforcing acyclicity on instantaneous effects, which most methods treat as a hard constraint, limiting flexibility and increasing computational cost. We address this with a new SVAR-based method that uses the Gumbel–Sinkhorn technique to impose acyclicity via a soft, differentiable permutation, enabling unified and efficient end-to-end optimization with gradient-based optimizer. Evaluations on IT monitoring, SWaT, and CausalRiver benchmarks show that our approach is both accurate and scalable: it delivers competitive and stable results on seven smaller IT monitoring datasets, achieves up to twice the F1 score of the second-best method on larger datasets, and provides substantial speedups, reaching up to 72× on SWaT and over 6× on CausalRiver compared to methods with explicit acyclicity constraints.

Future Work Several directions remain for future exploration. First, additional experiments could be conducted to address the limitations discussed in Section 6. Second, as discussed in Section 5.2, results can be sensitive to the choice of fixed pruning threshold, which motivates adaptive or stability-based pruning strategies that better distinguish noise from weak but meaningful effects. Another direction is to extend the model to capture non-linear causal relationships by incorporating non-linear functions or kernels, since our acyclicity enforcement mechanism is model-agnostic. Finally, as noted in Section 4.1, we use simple L_1 penalties for sparsity control, but strategies such as progressively increasing penalties across lags can be explored to further regularize causal discovery.

8 Reproducibility Statement

All hyperparameter settings are listed in Section 5.1. Details on preprocessing and postprocessing are provided in Appendix A.1 and Appendix A.2, respectively.

Our code is anonymously available at `https://anonymous.4open.science/r/Time-Series-Causal-Discovery-via-Differentiable-Permutations-6170/`.

The IT Monitoring and CausalRiver benchmarks are publicly accessible. The IT Monitoring benchmark can be obtained from `https://github.com/ckassaad/Case_Studies_of_Causal_Discovery_from_IT_Monitoring_Time_Series`, and the CausalRiver benchmark from `https://github.com/CausalRivers/causalrivers`. The SWaT dataset is private and accessible only under agreement; we obtained permission to use it in this study.

For the baselines used in our experiments, all implementations are in Python:

- VAR: available in the `statsmodels` library.
- MVGC: implemented by us in Python.
- Neural Granger causality variants (cLSTM, cRNN, cMLP): `https://github.com/iancovert/Neural-GC`.
- TCDF: `https://github.com/M-Nauta/TCDF`.
- PCMCI+: `https://github.com/jakobrunge/tigramite`.
- tsFCI: `https://sites.google.com/site/dorisentner/publications/tsfci` (originally in R; we provide a Python reimplementation).
- PCGCE: `https://github.com/ckassaad/PCGCE`.
- DYNOTEARS: available in the `causalnex` library.
- VARLiNGAM: `https://github.com/cdt15/lingam`.
- TiMINo: `https://web.math.ku.dk/~peters/code.html` (originally in R; we provide a Python reimplementation).

All experiments can be reproduced using the above code and datasets.

References

Ali Aıt-Bachir, Charles K Assaad, Christophe de Bignicourt, Emilie Devijver, Simon Ferreira, Eric Gaussier, Hosein Mohanna, and Lei Zan. Case studies of causal discovery from it monitoring time series, 2023. In *The History and Development of Search Methods for Causal Structure Workshop at the 39th Conference on Uncertainty in Artificial Intelligence*, 2023.

Bryon Aragam, Arash A Amini, and Qing Zhou. Learning directed acyclic graphs with penalized neighbourhood regression. *arXiv preprint arXiv:1511.08963*, 2015.

Charles K Assaad, Emilie Devijver, and Eric Gaussier. Discovery of extended summary graphs in time series. In *Uncertainty in Artificial Intelligence*, pp. 96–106. Pmlr, 2022a.

Charles K Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 2022b.

Lionel Barnett and Anil K Seth. The mvgc multivariate granger causality toolbox: a new approach to granger-causal inference. *Journal of neuroscience methods*, 223:50–68, 2014.

Adam B Barrett, Lionel Barnett, and Anil K Seth. Multivariate granger causality and generalized variance. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 81(4):041907, 2010.

Bertrand Charpentier, Simon Kibler, and Stephan Günnemann. Differentiable DAG sampling. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=9wOQOgNe-w`.

Selva Demiralp and Kevin D Hoover. Searching for the causal structure of a vector autoregression. *Oxford Bulletin of Economics and statistics*, 65:745–767, 2003.

Doris Entner and Patrik O Hoyer. On causal discovery from time series data using fci. *Probabilistic graphical models*, 16, 2010.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in neural information processing systems*, 33:12615–12625, 2020.

Stein Gideon, Shadaydeh Maha, Blunk Jan, Penzel Niklas, and Denzler Joachim. Causalrivers - scaling up benchmarking of causal discovery for real-world time-series. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wmV4cIbgl6.

Chang Gong, Chuzhe Zhang, Di Yao, Jingping Bi, Wenbin Li, and Yongjun Xu. Causal discovery from temporal data: An overview and new perspectives. *ACM Computing Surveys*, 57(4):1–38, 2024.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

Kevin D Hoover. *Causality in macroeconomics*. Cambridge University Press, 2001.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pp. 427–435. PMLR, 2013.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Lutz Kilian. New introduction to multiple time series analysis. *Econometric theory*, 22(5):961–967, 2006.

Markku Lanne, Mika Meitz, and Pentti Saikkonen. Identification and estimation of non-gaussian structural vector autoregressions. *Journal of Econometrics*, 196(2):288–304, 2017.

Te-Won Lee. Independent component analysis. In *Independent component analysis: Theory and applications*, pp. 27–66. Springer, 1998.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Rajib Ranjan Maiti, Sridhar Adepu, and Emil Lupu. Iccps: Impact discovery using causal inference for cyber attacks in cpss. *arXiv preprint arXiv:2307.14161*, 2023.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Byt3oJ-0W.

Søren Wengel Mogensen, Karin Rathsman, and Per Nilsson. Causal discovery in a complex industrial system: A time series benchmark. In *Causal Learning and Reasoning*, pp. 1218–1236. PMLR, 2024.

Alessio Moneta and Peter Spirtes. Graphical models for the identification of causal structures in multivariate time series models. In *9th Joint International Conference on Information Sciences (JCIS-06)*, pp. 613–616. Atlantis press, 2006.

Joris Mooij, Dominik Janzing, Jonas Peters, and Bernhard Schölkopf. Regression by dependence minimization and its application to causal inference in additive noise models. In *Proceedings of the 26th annual international conference on machine learning*, pp. 745–752, 2009.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

AS Nemirovsky. Optimization ii. numerical methods for nonlinear continuous optimization. *Technion–Israel Institute of Technology*, 1999.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1595–1605. Pmlr, 2020.

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

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

Jonas Peters, Dominik Janzing, and Bernhard Scholkopf. Causal inference on time series using restricted structural equation models. *Advances in neural information processing systems*, 26, 2013.

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

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on uncertainty in artificial intelligence*, pp. 1388–1397. Pmlr, 2020.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Munoz-Mari, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019a.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5 (11):eaau4996, 2019b.

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.

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.

Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pp. 1–48, 1980.

Norman R Swanson and Clive WJ Granger. Impulse response functions based on a causal approach to residual orthogonalization in vector autoregressions. *Journal of the American Statistical Association*, 92(437):357–367, 1997.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

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

USE OF LARGE LANGUAGE MODELS (LLMS)

Large Language Models (LLMs) were used in this work solely as writing assistant. Specifically, they were employed to refine grammar, improve clarity, shorten text when requested, and polish the presentation of results (e.g., experimental analysis, future work discussion, and table overviews). In addition, LLMs were used to help organize code and improve comments, but not for developing or directly writing code implementation for the main method.

LLMs were not used for research ideation, methodological design, data analysis, or result interpretation. All conceptual contributions, experimental design choices, data processing, and analysis were conducted entirely by the authors. The authors take full responsibility for the contents of this paper.

A APPENDIX

A.1 PREPROCESSING

For all datasets, we preprocess the raw data by removing timestamps to ensure fair comparison across methods. Since the ground-truth causal graphs are provided in different formats across benchmarks, we convert them into a standardized summary matrix in `.npy` format for evaluation.

For the SWaT dataset (Maiti et al., 2023), we resample the data at 5-second intervals instead of the original 1-second sampling rate. This is because system responses in SWaT typically exhibit delays of more than 10 seconds. Without resampling, the valid maximum lag would be unrealistically large, making evaluation infeasible and potentially producing invalid results. In addition, we remove variables that remain constant throughout the dataset, since they provide no statistical information and can cause certain methods (e.g., VARLiNGAM) to crash.

A.2 POSTPROCESSING

All ground-truth graphs are represented as *summary graphs* (see definition in Assaad et al. (2022b)). Accordingly, we process the outputs of all methods (including VAR, VARLiNGAM, and our method), which generate separate results for different lags, into a single summary graph. Specifically, for each edge position we take the maximum value across lags, including instantaneous effects. Following Assaad et al. (2022b); Aıt-Bachir et al. (2023); Gideon et al. (2025), we remove diagonal elements when evaluating, as they represent self-autocorrelation, which—although important—does not provide additional information about causality between variables. Finally, for methods requiring a pruning threshold, we apply a fixed threshold of 0.01, consistent with prior work (Assaad et al., 2022b; Aıt-Bachir et al., 2023).

A.3 ADDITIONAL F1 RESULTS

Tables 6–9 present the F1 scores across all datasets under maximum lags of 3, 5, 10, and 15.

Table 6: F1 scores on all datasets with a maximum lag of 3

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.1667	0.1622	0.2449	0.2222	0.2895	0.2047	0.2188	0.0590	0.0638	0.0064	0.0049
MVGC	0.0909	0.0000	0.0667	0.2059	0.2400	0.1613	0.1374	0.0758	0.0508	TLE	TLE
cLSTM	0.3390	0.3390	0.2500	0.2321	0.2500	0.1739	0.1739	0.0465	0.0465	TLE	TLE
cMLP	0.2857	0.1622	0.1379	0.1905	0.1081	0.2247	0.2857	0.0830	0.1139	TLE	TLE
cRNN	0.1714	0.1500	0.2069	0.1778	0.1667	0.2273	0.2903	0.0788	0.0991	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.1053	0.1818	0.2000	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.0000	0.0000	0.3243	0.1579	0.0357	0.0833	TLE	TLE	TLE	TLE
tsFCI	0.2286	0.0870	0.0000	0.1818	0.1714	0.1852	0.1159	TLE	TLE	TLE	TLE
PCGCE	0.0909	0.1538	0.1935	0.2143	0.1961	0.2250	0.2400	0.0820	0.0901	TLE	TLE
DYNOTEARS	0.2857	0.2353	0.1538	0.2623	0.2895	0.1905	0.2056	0.0440	0.1311	TLE	TLE
VARLiNGAM	0.0000	0.0000	0.3846	0.2593	0.2667	0.1923	0.2188	0.0677	0.0326	TLE	TLE
TiMINo	0.1538	0.1818	0.0000	0.0000	0.0000	0.0000	0.0000	0.0339	0.0000	TLE	TLE
Our Method	0.4000	0.3415	0.2769	0.3243	0.3243	0.1830	0.1875	0.2162	0.3125	0.1745	0.1212

First, our method consistently achieves top or near-top performance across datasets and lag settings. For example, it outperforms all baselines on MoM1/2, Ingestion, and Web datasets under most lags,

Table 7: F1 scores on all datasets with a maximum lag of 5

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.2632	0.2051	0.2400	0.2540	0.2857	0.2047	0.2188	0.0565	0.0607	0.0059	0.0045
MVGC	0.0833	0.1333	0.0667	0.2188	0.2368	0.1185	0.1212	0.0767	0.0559	TLE	TLE
cLSTM	0.3390	0.3390	0.2500	0.2321	0.2500	0.1739	0.1739	0.0279	0.0465	TLE	TLE
cMLP	0.2857	0.1622	0.1538	0.1860	0.1000	0.2247	0.2571	0.0902	0.1203	TLE	TLE
cRNN	0.1714	0.1905	0.2143	0.1818	0.1143	0.2353	0.3030	0.0788	0.0952	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.1053	0.1818	0.2000	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.1000	0.0000	0.2703	0.2051	0.0323	0.0800	TLE	TLE	TLE	TLE
tsFCI	0.2286	0.0870	0.0000	0.1875	0.1212	0.1250	0.1270	TLE	TLE	TLE	TLE
PCGCE	0.0833	0.0769	0.2286	0.1961	0.1132	0.2105	0.2632	0.0648	0.1116	TLE	TLE
DYNOTEARS	0.1250	0.3000	0.1538	0.2593	0.3462	0.1765	0.2131	0.0206	0.1342	TLE	TLE
VARLiNGAM	0.0000	0.0909	0.2500	0.2373	0.2535	0.1584	0.1655	0.0660	0.0403	TLE	TLE
TiMINo	0.0000	0.1667	0.0000	0.0000	0.0000	0.0000	0.0000	0.0580	0.0000	TLE	TLE
Our Method	0.4615	0.3784	0.2769	0.2697	0.2899	0.1916	0.2014	0.2202	0.3000	0.1751	0.1351

Table 8: F1 scores on all datasets with a maximum lag of 10

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.2500	0.2927	0.2264	0.2647	0.2785	0.2016	0.2154	0.0539	0.0574	0.0056	0.0041
MVGC	0.0833	0.0000	0.0667	0.2258	0.2368	0.1194	0.1408	0.0787	0.0572	TLE	TLE
cLSTM	0.3390	0.3390	0.2500	0.2321	0.2500	0.1739	0.1739	0.0000	0.0466	TLE	TLE
cMLP	0.1622	0.1463	0.1481	0.1778	0.1053	0.2198	0.2353	0.0833	0.1158	TLE	TLE
cRNN	0.1667	0.1500	0.2069	0.1818	0.1111	0.2299	0.2985	0.0729	0.1009	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.1053	0.1818	0.2000	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.0000	0.0000	0.2927	0.2051	0.0597	0.1071	TLE	TLE	TLE	TLE
tsFCI	0.2353	0.0870	0.0000	0.2000	0.1176	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.0714	0.2222	0.2917	0.1538	0.1772	0.2933	0.0827	0.0813	TLE	TLE
DYNOTEARS	0.2353	0.2222	0.1538	0.2759	0.3673	0.1757	0.2299	0.0215	0.1333	TLE	TLE
VARLiNGAM	0.2667	0.0000	0.2727	0.2545	0.2308	0.1856	0.1642	0.0621	0.0362	TLE	TLE
TiMINo	0.1429	0.2667	0.0000	0.0000	0.0000	0.0000	0.0000	0.0435	0.0000	TLE	TLE
Our Method	0.4286	0.4000	0.2812	0.2697	0.3014	0.1905	0.2162	0.2182	0.3226	0.1860	0.1388

Table 9: F1 scores on all datasets with a maximum lag of 15.

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.2927	0.2500	0.2308	0.2535	0.2683	0.2000	0.2137	0.0512	0.0561	0.0054	0.0039
MVGC	0.0833	0.0000	0.0667	0.2258	0.2192	0.1231	0.1630	0.0877	0.0556	TLE	TLE
cLSTM	0.3390	0.3390	0.2500	0.2321	0.2500	0.1739	0.1739	0.0000	0.0466	TLE	TLE
cMLP	0.2162	0.1579	0.1481	0.1905	0.1053	0.2247	0.2647	0.0844	0.1172	TLE	TLE
cRNN	0.2222	0.1463	0.2069	0.1905	0.1111	0.2247	0.2941	0.0753	0.1026	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.1053	0.1818	0.2000	0.0000	0.0000	TLE	TLE
PCMCI+	0.2727	0.0000	0.0000	0.3000	0.2051	0.0870	0.0984	TLE	TLE	TLE	TLE
tsFCI	0.2353	0.0870	0.0000	0.2000	0.1176	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.0714	0.1579	0.2353	0.1852	0.1707	0.2338	0.0722	0.0735	TLE	TLE
DYNOTEARS	0.3636	0.2500	0.1538	0.2963	0.3600	0.1892	0.2273	0.0230	0.1560	TLE	TLE
VARLiNGAM	0.0000	0.1333	0.2857	0.2642	0.2368	0.1818	0.1395	0.0618	0.0407	TLE	TLE
TiMINo	0.1538	0.1667	0.1667	0.0000	0.0000	0.0000	0.0000	0.0364	0.0000	TLE	TLE
Our Method	0.3415	0.4091	0.2769	0.2697	0.3143	0.1893	0.2222	0.1818	0.3438	0.1997	0.1279

and maintains competitive accuracy on the AntiVirus datasets, where most methods struggle due to their sparse and irregular event-driven nature.

Second, baseline methods often show dataset-specific strengths but lack robustness. VARLiNGAM performs well on Ingestion but poorly on MoM datasets, DYNOTEARS excels on Web1/Web2 but degrades on MoM and AntiVirus, while cRNN and PCGCE perform relatively better on AntiVirus datasets. This indicates that these methods may overfit to particular data characteristics rather than generalize across domains.

Third, increasing the maximum lag generally benefits our method, especially on larger datasets such as Flood, Bavaria, and East Germany, where it achieves the strongest performance once longer temporal dependencies are captured. In contrast, many baselines either fail to scale to these datasets (TLE) or show little improvement when lag increases.

Overall, these results confirm that our method provides both stable and scalable causal discovery performance across heterogeneous datasets, whereas existing baselines are less consistent and often limited to narrow data regimes.

Table 10: F1 scores on AntiV1 and AntiV2 datasets with different pruning thresholds across lags. A fixed pruning threshold of 0.01 is used in our main experiments.

Threshold	AntiV1				AntiV2			
	0.01	0.05	0.10	0.15	0.01	0.05	0.10	0.15
Lag 3	0.1830	0.2128	0.2034	0.2222	0.1875	0.2273	0.2273	0.3030
Lag 5	0.1916	0.2500	0.2500	0.2667	0.2014	0.2857	0.2857	0.2581
Lag 10	0.1905	0.2059	0.2400	0.2400	0.2162	0.2581	0.2581	0.2581
Lag 15	0.1893	0.1778	0.1778	0.1778	0.2222	0.2500	0.2500	0.2500

A.4 Additional Results with Different Pruning Thresholds

Table 10 reports F1 scores on the AntiV1 and AntiV2 datasets with different pruning thresholds. In our main experiments, we fixed the threshold at 0.01, under which the performance of our method is less impressive compared to other datasets. However, when the pruning threshold is tuned, the results improve noticeably on both datasets (e.g., AntiV1 increases from 0.1916 at threshold 0.01 to 0.2667 at threshold 0.15 for lag 5; AntiV2 increases from 0.2014 at threshold 0.01 to 0.3030 at threshold 0.15 for lag 3).

These findings indicate that our method is sensitive to the choice of pruning threshold: a fixed universal value may not be optimal across different datasets, particularly for the AntiVirus datasets with sparse and irregular event-driven signals. While tuning the threshold can improve performance to some extent, this sensitivity also highlights a limitation of our current approach and motivates future work on developing adaptive pruning strategies that can automatically distinguish spurious from meaningful causal effects.

A.5 Additional Recall Results

Tables 11–14 summarize recall across all datasets and lag settings. Several consistent patterns emerge.

Table 11: Recall on all datasets with a maximum lag of 3. *TLE*: time limit exceeded (3 hours).

Method	**MoM 1**	**MoM 2**	**Ingestion**	**Web 1**	**Web 2**	**AntiV 1**	**AntiV 2**	**SWaT**	**Flood**	**Bavaria**	**East Germany**
VAR	0.3000	0.3000	0.6667	0.5000	0.7857	0.8125	0.8750	0.7297	1.0000	0.9816	0.9339
MVGC	0.1000	0.0000	0.1111	0.5000	0.6429	0.6250	0.5625	0.7297	0.5238	TLE	TLE
cLSTM	1.0000	1.0000	1.0000	0.9286	1.0000	1.0000	1.0000	1.0000	0.9762	TLE	TLE
cMLP	0.5000	0.3000	0.2222	0.2857	0.1429	0.6250	0.6250	0.2703	0.3810	TLE	TLE
cRNN	0.3000	0.3000	0.3333	0.2857	0.2143	0.6250	0.5625	0.3636	0.4048	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.0714	0.1250	0.1250	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.0000	0.0000	0.4286	0.2143	0.0625	0.1250	TLE	TLE	TLE	TLE
tsFCI	0.4000	0.1000	0.0000	0.2143	0.2143	0.3125	0.2500	TLE	TLE	TLE	TLE
PCGCE	0.1000	0.2000	0.3333	0.4286	0.3571	0.5625	0.5625	0.2703	0.2381	TLE	TLE
DYNOTEARS	0.3000	0.2000	0.1111	0.5714	0.6429	0.8750	0.6875	0.0541	0.2857	TLE	TLE
VARLiNGAM	0.0000	0.0000	0.5556	0.5000	0.5714	0.6250	0.8750	0.7297	0.3333	TLE	TLE
TiMINo	0.1000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0270	0.0000	TLE	TLE
Our method	0.8000	0.7000	1.0000	0.8571	0.8571	0.8750	0.7500	0.3243	0.2381	0.1061	0.0676

Table 12: Recall on all datasets with a maximum lag of 5. *TLE*: time limit exceeded (3 hours).

Method	**MoM 1**	**MoM 2**	**Ingestion**	**Web 1**	**Web 2**	**AntiV 1**	**AntiV 2**	**SWaT**	**Flood**	**Bavaria**	**East Germany**
VAR	0.5000	0.4000	0.6667	0.5714	0.7857	0.8125	0.8750	0.7297	1.0000	0.9837	0.9386
MVGC	0.1000	0.2000	0.1111	0.5000	0.6429	0.5000	0.5000	0.7027	0.6429	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.0714	0.1250	0.1250	1.0000	0.9762	TLE	TLE
PCMCI+	0.0000	0.1000	0.0000	0.3571	0.2857	0.0625	0.1250	0.2973	0.3810	TLE	TLE
tsFCI	0.4000	0.1000	0.0000	0.2143	0.1429	0.1875	0.2500	0.3636	0.3810	TLE	TLE
PCGCE	0.1000	0.1000	0.4444	0.3571	0.2143	0.5000	0.6250	0.0000	0.0000	TLE	TLE
cLSTM	1.0000	1.0000	1.0000	0.9286	1.0000	1.0000	1.0000	TLE	TLE	TLE	TLE
cMLP	0.5000	0.3000	0.2222	0.2857	0.1429	0.6250	0.5625	TLE	TLE	TLE	TLE
cRNN	0.3000	0.4000	0.3333	0.2857	0.1429	0.6250	0.6250	0.2162	0.3095	TLE	TLE
DYNOTEARS	0.1000	0.3000	0.1111	0.5000	0.6429	0.7500	0.8125	0.0270	0.2381	TLE	TLE
VARLiNGAM	0.0000	0.1000	0.3333	0.5000	0.6429	0.5000	0.7500	0.7838	0.3333	TLE	TLE
TiMINo	0.0000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0541	0.0000	TLE	TLE
Our method	0.9000	0.7000	1.0000	0.8571	0.7143	1.0000	0.8750	0.3243	0.2143	0.1061	0.0768

Table 13: Recall on all datasets with a maximum lag of 10. *TLE*: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.5000	0.6000	0.6667	0.6429	0.7857	0.8125	0.8750	0.7568	1.0000	0.9898	0.9493
MVGC	0.1000	0.0000	0.1111	0.5000	0.6429	0.5000	0.6250	0.6757	0.7381	TLE	TLE
cLSTM	1.0000	1.0000	1.0000	0.9286	1.0000	1.0000	1.0000	0.0000	0.9762	TLE	TLE
cMLP	0.3000	0.3000	0.2222	0.2857	0.1429	0.6250	0.5000	0.2703	0.3571	TLE	TLE
cRNN	0.3000	0.3000	0.3333	0.2857	0.1429	0.6250	0.6250	0.3182	0.3810	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.0714	0.1250	0.1250	0.0000	0.0000	TLE	TLE
PCMICI+	0.0000	0.0000	0.0000	0.4286	0.2857	0.1250	0.1875	TLE	TLE	TLE	TLE
tsFCI	0.4000	0.1000	0.0000	0.2143	0.1429	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.1000	0.4444	0.5000	0.2857	0.4375	0.6875	0.2973	0.2381	TLE	TLE
DYNOTEARS	0.2000	0.2000	0.1111	0.5714	0.6429	0.8125	0.6250	0.0270	0.2143	TLE	TLE
VARLiNGAM	0.2000	0.0000	0.3333	0.5000	0.6429	0.5625	0.6875	0.7838	0.2381	TLE	TLE
TiMINo	0.1000	0.2000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0270	0.0000	TLE	TLE
Our method	0.9000	0.8000	1.0000	0.8571	0.7857	1.0000	1.0000	0.3243	0.2381	0.1163	0.0783

Table 14: Recall on all datasets with a maximum lag of 15. *TLE*: time limit exceeded (3 hours) limit.

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.6000	0.5000	0.6667	0.6429	0.7857	0.8125	0.8750	0.7297	1.0000	0.9898	0.9508
MVGC	0.1000	0.0000	0.1111	0.5000	0.5714	0.5000	0.6875	0.7027	0.7619	TLE	TLE
cLSTM	1.0000	1.0000	1.0000	0.9286	1.0000	1.0000	1.0000	0.0000	0.9762	TLE	TLE
cMLP	0.4000	0.3000	0.2222	0.2857	0.1429	0.6250	0.5625	0.2703	0.3571	TLE	TLE
cRNN	0.4000	0.3000	0.3333	0.2857	0.1429	0.6250	0.6250	0.3182	0.3810	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.0714	0.1250	0.1250	0.0000	0.0000	TLE	TLE
PCMCI+	0.3000	0.0000	0.0000	0.4286	0.2857	0.1875	0.1875	TLE	TLE	TLE	TLE
tsFCI	0.4000	0.1000	0.0000	0.2143	0.1429	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.1000	0.3333	0.4286	0.3571	0.4375	0.5625	0.2703	0.2143	TLE	TLE
DYNOTEARS	0.4000	0.2000	0.1111	0.5714	0.6429	0.8750	0.6250	0.0270	0.2619	TLE	TLE
VARLiNGAM	0.0000	0.1000	0.3333	0.5000	0.6429	0.5000	0.5625	0.7838	0.2857	TLE	TLE
TiMINo	0.1000	0.1000	0.1111	0.0000	0.0000	0.0000	0.0000	0.0270	0.0000	TLE	TLE
Our method	0.7000	0.9000	1.0000	0.8571	0.7857	1.0000	1.0000	0.2703	0.2619	0.1286	0.0722

First, our method achieves high recall on small- and medium-scale datasets such as MoM1/2, Ingestion, and Web, where it often reaches values close to or equal to 1.0. This indicates that our approach is effective at recovering true causal edges in settings with limited dimensionality, and performs comparably or better than strong baselines such as cLSTM.

Second, on the AntiVirus datasets, our method maintains strong recall (0.75–1.0), highlighting its ability to capture the majority of ground-truth causal relations even in event-driven time series. By contrast, many baselines either underperform or show large variance across different lags.

Third, recall drops substantially on large-scale datasets such as Flood, Bavaria, and East Germany, where our method records much lower values (e.g., below 0.3 in several cases). In contrast, VAR consistently achieves nearly perfect recall on these datasets, though at the cost of very low precision (as shown in Section A.6). This suggests that our pruning strategy is conservative, favoring precision over recall in high-dimensional settings.

Overall, these results show that our method provides balanced recall in small- and medium-scale settings while remaining competitive on more challenging datasets. However, the decline in recall on large-scale datasets underscores the limitation of using a fixed pruning threshold, which may discard weak but meaningful causal effects. Developing adaptive pruning strategies could help alleviate this trade-off and improve recall without sacrificing precision.

A.6 Additional Precision Results

Tables 15–18 report precision across all datasets and lag settings. Several patterns can be observed.

First, our method consistently achieves the highest precision on large-scale datasets such as Flood, Bavaria, and East Germany, where it maintains values above 0.44 and up to 0.61. This demonstrates that our approach is effective at avoiding false positives when scaling to high-dimensional settings, in contrast to most baselines which either fail to scale (TLE) or degrade sharply in precision.

Second, on small- and medium-scale datasets, our method shows more moderate precision compared to some baselines. For example, DYNOTEARS and VARLiNGAM occasionally achieve higher precision on Web or Ingestion datasets, while TiMINo can reach very high precision in isolated

Table 15: Precision on all datasets with a maximum lag of 3. *TLE*: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.1154	0.1111	0.1500	0.1429	0.1774	0.1171	0.1250	0.0308	0.0329	0.0032	0.0025
MVGC	0.0833	0.0000	0.0476	0.1296	0.1475	0.0926	0.0783	0.0400	0.0267	TLE	TLE
cLSTM	0.2041	0.2041	0.1429	0.1327	0.1429	0.0952	0.0952	0.0238	0.0238	TLE	TLE
cMLP	0.2000	0.1111	0.1000	0.1429	0.0870	0.1370	0.1852	0.0490	0.0669	TLE	TLE
cRNN	0.1200	0.1000	0.1500	0.1290	0.1364	0.1389	0.1957	0.0442	0.0565	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.2000	0.3333	0.5000	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.0000	0.0000	0.2609	0.1250	0.0250	0.0625	TLE	TLE	TLE	TLE
tsFCI	0.1600	0.0769	0.0000	0.1579	0.1429	0.1316	0.0755	TLE	TLE	TLE	TLE
PCGCE	0.0833	0.1250	0.1364	0.1429	0.1351	0.1406	0.1525	0.0483	0.0556	TLE	TLE
DYNOTEARS	0.2727	0.2857	0.2500	0.1702	0.2500	0.1069	0.1209	0.0370	0.0851	TLE	TLE
VARLiNGAM	0.0000	0.0000	0.2941	0.1750	0.1739	0.1136	0.1250	0.0355	0.0171	TLE	TLE
TiMINo	0.3333	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0455	0.0000	TLE	TLE
Our method	0.2667	0.2258	0.1607	0.2000	0.2000	0.1022	0.1071	0.1622	0.4545	0.4906	0.5867

Table 16: Precision on all datasets with a maximum lag of 5. *TLE*: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.1786	0.1379	0.1463	0.1633	0.1746	0.1171	0.1250	0.0294	0.0313	0.0030	0.0022
MVGC	0.0714	0.1000	0.0476	0.1400	0.1452	0.0672	0.0690	0.0406	0.0292	TLE	TLE
cLSTM	0.2041	0.2041	0.1429	0.1327	0.1429	0.0952	0.0952	0.0141	0.0238	TLE	TLE
cMLP	0.2000	0.1111	0.1176	0.1379	0.0769	0.1370	0.1667	0.0531	0.0714	TLE	TLE
cRNN	0.1200	0.1250	0.1579	0.1333	0.0952	0.1449	0.2000	0.0442	0.0544	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.2000	0.3333	0.5000	0.0000	0.0000	TLE	TLE
PCMCI+	0.0000	0.1000	0.0000	0.2174	0.1600	0.0217	0.0588	TLE	TLE	TLE	TLE
tsFCI	0.1600	0.0769	0.0000	0.1667	0.1053	0.0938	0.0851	TLE	TLE	TLE	TLE
PCGCE	0.0714	0.0625	0.1538	0.1351	0.0769	0.1333	0.1667	0.0381	0.0681	TLE	TLE
DYNOTEARS	0.1667	0.3000	0.2500	0.1750	0.2368	0.1000	0.1226	0.0167	0.0935	TLE	TLE
VARLiNGAM	0.0000	0.0833	0.2000	0.1556	0.1579	0.0941	0.0930	0.0344	0.0215	TLE	TLE
TiMINo	0.0000	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0625	0.0000	TLE	TLE
Our method	0.3103	0.2593	0.1607	0.1600	0.1818	0.1060	0.1138	0.1667	0.5000	0.5000	0.5618

Table 17: Precision on all datasets with a maximum lag of 10. *TLE*: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.1667	0.1935	0.1364	0.1667	0.1692	0.1150	0.1228	0.0280	0.0296	0.0028	0.0021
MVGC	0.0714	0.0000	0.0476	0.1458	0.1452	0.0678	0.0794	0.0418	0.0298	TLE	TLE
cLSTM	0.2041	0.2041	0.1429	0.1327	0.1429	0.0952	0.0952	0.0000	0.0239	TLE	TLE
cMLP	0.1111	0.0968	0.1111	0.1290	0.0833	0.1333	0.1538	0.0493	0.0691	TLE	TLE
cRNN	0.1154	0.1000	0.1500	0.1333	0.0909	0.1408	0.1961	0.0412	0.0582	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.2000	0.3333	0.5000	0.0000	0.0000	TLE	TLE
PCMICI+	0.0000	0.0000	0.0000	0.2222	0.1600	0.0392	0.0750	TLE	TLE	TLE	TLE
tsFCI	0.1667	0.0769	0.0000	0.1875	0.1000	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.0556	0.1481	0.2059	0.1053	0.1111	0.1864	0.0480	0.0490	TLE	TLE
DYNOTEARS	0.2857	0.2500	0.2500	0.1818	0.2571	0.0985	0.1408	0.0179	0.0968	TLE	TLE
VARLiNGAM	0.4000	0.0000	0.2308	0.1707	0.1406	0.1111	0.0932	0.0323	0.0196	TLE	TLE
TiMINo	0.2500	0.4000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1111	0.0000	TLE	TLE
Our method	0.2812	0.2667	0.1636	0.1600	0.1864	0.1053	0.1212	0.1644	0.5000	0.4634	0.6071

Table 18: Precision on all datasets with a maximum lag of 15. *TLE*: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Ingestion	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	East Germany
VAR	0.1935	0.1667	0.1395	0.1579	0.1618	0.1140	0.1217	0.0265	0.0289	0.0027	0.0020
MVGC	0.0714	0.0000	0.0476	0.1458	0.1356	0.0702	0.0924	0.0468	0.0288	TLE	TLE
cLSTM	0.2041	0.2041	0.1429	0.1327	0.1429	0.0952	0.0952	0.0000	0.0239	TLE	TLE
cMLP	0.1481	0.1071	0.1111	0.1429	0.0833	0.1370	0.1731	0.0500	0.0701	TLE	TLE
cRNN	0.1538	0.0968	0.1500	0.1429	0.0909	0.1370	0.1923	0.0427	0.0593	TLE	TLE
TCDF	0.0000	0.0000	0.0000	0.0000	0.2000	0.3333	0.5000	0.0000	0.0000	TLE	TLE
PCMCI+	0.2500	0.0000	0.0000	0.2308	0.1600	0.0566	0.0667	TLE	TLE	TLE	TLE
tsFCI	0.1667	0.0769	0.0000	0.1875	0.1000	0.0000	0.0000	TLE	TLE	TLE	TLE
PCGCE	0.0000	0.0556	0.1034	0.1622	0.1250	0.1061	0.1475	0.0417	0.0443	TLE	TLE
DYNOTEARS	0.3333	0.3333	0.2500	0.2000	0.2500	0.1061	0.1389	0.0200	0.1111	TLE	TLE
VARLiNGAM	0.0000	0.2000	0.2500	0.1795	0.1452	0.1111	0.0796	0.0322	0.0219	TLE	TLE
TiMINO	0.3333	0.5000	0.3333	0.0000	0.0000	0.0000	0.0000	0.0556	0.0000	TLE	TLE
Our method	0.2258	0.2647	0.1607	0.1600	0.1964	0.1046	0.1250	0.1370	0.5000	0.4468	0.5595

cases (e.g., MoM2 with lag 3). However, these methods tend to be unstable across datasets, often collapsing to zero or near-zero precision in other settings, whereas our method remains consistently competitive.

Third, precision on the AntiVirus datasets remains relatively low for all methods, including ours, reflecting the difficulty of handling sparse and irregular event-driven signals. This again highlights the sensitivity of pruning strategies in such settings, as discussed in Section A.4.

Overall, these results confirm that while some baselines may excel on particular datasets, our method provides the best trade-off between stability and scalability. In particular, its strong precision on large-scale datasets shows its practical value for high-dimensional causal discovery tasks.

A.7 Additional Runtime Results

Tables 19–21 report the runtime across all datasets and lag settings. On small-scale datasets (the first seven columns), our method is not the fastest—linear approaches such as VAR and PCGCE consistently achieve lower runtime. However, our method still completes within seconds to tens of seconds, which remains practical for these scales.

Table 19: Runtime (in seconds) for All Datasets with Lag = 3. TLE: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Storm	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	Germany
VAR	0.000	0.000	0.000	0.015	0.016	0.006	0.002	0.09	0.05	5.32	9.49
MVGC	0.090	0.100	0.079	0.367	0.312	0.592	1.090	135.58	129.37	TLE	TLE
cLSTM	22.89	25.684	34.635	67.289	65.577	52.093	44.384	1262.41	227.32	TLE	TLE
cMLP	5.586	5.641	7.257	29.618	30.507	13.395	13.082	209.29	135.37	TLE	TLE
cRNN	17.165	17.413	21.860	62.714	62.004	47.574	47.899	1144.85	160.22	TLE	TLE
TCDF	8.69	9.02	9.94	58.36	58.55	16.65	17.81	550.15	258.03	TLE	TLE
PCMCI+	0.250	0.293	3.233	40.194	31.030	23.617	38.671	TLE	TLE	TLE	TLE
tsFCI	26.378	14.421	1.048	100.715	332.605	245.800	620.111	TLE	TLE	TLE	TLE
PCGCE	0.139	0.433	0.858	20.652	4.460	18.210	7.867	777.46	75.00	TLE	TLE
DYNOTEARS	2.194	1.701	0.031	15.649	37.256	10.595	161.464	740.44	259.33	TLE	TLE
VARLINGAM	0.088	0.122	0.128	2.189	2.201	0.437	0.750	409.37	15.02	TLE	TLE
TiMINo	1.760	1.809	2.759	17.630	21.774	6.888	6.539	1654.83	79.98	TLE	TLE
Our Method	9.679	8.174	8.180	11.881	14.797	8.146	10.065	23.52	23.29	716.42	1213.15

Table 20: Runtime (in seconds) for All Datasets with Lag = 5. TLE: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Storm	Web 1	Web 2	AntiV 1	AntiV 2	SWaT	Flood	Bavaria	Germany
VAR	0	0.00	0.00	0.02	0.02	0.02	0.02	0.17	0.10	12.08	19.69
MVGC	0.110	0.17	0.11	1.41	1.52	4.07	4.10	1064.92	816.72	TLE	TLE
cLSTM	20.117	24.24	41.86	94.96	92.77	72.17	73.23	4346.78	367.78	TLE	TLE
cMLP	6.736	6.91	9.08	30.46	36.89	18.81	21.35	261.13	138.31	TLE	TLE
cRNN	16.55	19.89	27.82	83.52	73.93	49.95	50.87	1845.22	274.41	TLE	TLE
TCDF	8.69	9.02	9.94	58.36	58.55	16.65	17.81	550.15	258.03	TLE	TLE
PCMCI+	0.190	0.37	6.32	187.47	78.93	87.51	65.56	TLE	TLE	TLE	TLE
tsFCI	29.77	19.36	0.94	131.07	203.35	368.48	786.60	TLE	TLE	TLE	TLE
PCGCE	0.180	0.54	0.91	13.65	4.65	22.43	9.72	871.41	75.22	TLE	TLE
DYNOTEARS	10.77	2.90	0.06	56.42	57.53	15.33	364.57	1897.80	313.75	TLE	TLE
VARLINGAM	0.093	0.13	0.21	4.75	4.58	1.06	1.07	1208.18	29.36	TLE	TLE
TiMINo	1.312	1.76	2.12	13.83	17.30	7.07	7.09	1051.04	79.85	TLE	TLE
Our Method	8.317	7.47	8.34	12.75	15.87	6.55	8.07	25.06	26.89	929.51	1461.60

Table 21: Runtime (in seconds) for All Datasets with Lag = 10. TLE: time limit exceeded (3 hours).

Method	MoM 1	MoM 2	Storm	Web 1	Web 2	Antivirus 1	Antivirus 2	SWaT	Flood	Bavaria	East Germany
VAR	0	0.00	0.00	0.03	0.03	0.03	0.05	0.43	0.43	37.35	67.78
MVGC	0.209	0.35	0.19	4.52	4.42	4.83	5.24	2064.97	1265.48	TLE	TLE
cLSTM	34.63	38.22	51.91	125.74	165.95	106.02	94.09	TLE	656.06	TLE	TLE
cMLP	12.724	14.41	14.86	47.48	47.87	29.79	21.95	359.23	204.77	TLE	TLE
cRNN	19.313	21.27	29.25	167.09	129.71	66.18	66.44	6281.79	797.46	TLE	TLE
TCDF	8.69	9.02	9.94	58.36	58.55	16.65	17.81	550.15	258.03	TLE	TLE
PCMCI+	0.619	1.33	7.67	231.03	81.34	39.42	39.85	TLE	TLE	TLE	TLE
tsFCI	93.79	53.46	1.55	549.18	3582.58	3179.41	TLE	TLE	TLE	TLE	TLE
PCGCE	0.299	0.61	1.46	9.09	5.62	38.82	10.85	856.84	81.11	TLE	TLE
DYNOTEARS	11.554	12.46	0.30	50.85	365.70	54.52	306.68	3086.13	516.16	TLE	TLE
VARLINGAM	0.320	0.34	0.59	8.94	9.44	1.30	1.24	2945.26	55.11	TLE	TLE
TiMINo	1.468	1.48	2.34	14.98	13.57	7.00	7.09	636.87	78.71	TLE	TLE
Our Method	7.168	8.60	7.41	12.43	16.38	7.61	7.04	33.78	18.01	1498.22	2521.38

The difference becomes more pronounced on medium- to large-scale datasets. For SWaT and Flood, our method finishes in tens of seconds, whereas baselines such as cLSTM, cRNN, DYNOTEARS,

and TiMINo often require hundreds to thousands of seconds, and methods like MVGC, PCMCI+, and tsFCI regularly exceed the 3-hour timeout. On the largest datasets (Bavaria and East Germany), nearly all baseline methods fail to return results within the time limit, while our method is able to complete in under one hour to about forty minutes, depending on the lag setting.

An additional observation is that while deep learning-based methods (cLSTM, cRNN, cMLP) often scale poorly and time out on larger datasets, our approach remains robust across different lags. Moreover, causal discovery methods that explicitly enforce acyclicity (e.g., DYNOTEARS, VAR-LiNGAM, TiMINo) typically exhibit severe runtime growth, while our method avoids this by embedding acyclicity into a differentiable reparameterization, allowing for end-to-end optimization without expensive combinatorial operations.