LEARNING TIME-VARYING DIRECTED GRAPHS VIA CONVEX TV-REGULARIZED OPTIMIZATION AND DAG PROJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous optimization methods for learning directed acyclic graphs (DAGs) have made progress in structure learning. However, they face two key limitations: (1) they assume a single, stationary graph structure across all observations, and (2) their nested nonconvex optimization with matrix exponential evaluations scales inefficiently as the number of variables grows—a bottleneck in high-dimensional applications. To address these limitations, we propose DACORD: **DA**G via **C**onvex **O**ptimization with **R**egime **D**etection, a two-stage method for learning time-varying DAGs. We utilize the insight that for models with a fixed feature map, acyclicity is the core source of nonconvexity in score-based DAG learning. Stage 1 jointly estimates window-specific coefficients by solving a convex regression problem with sparsity and total variation regularization. A single-loop primal–dual algorithm solves this problem, and changes in the estimated coefficients provide candidate regime boundaries. Stage 2 constructs a DAG for each estimated regime using greedy ordering and back-edge removal. Extensive experiments show that, at $d = 200$ with $R = 10$ regimes, DACORD completes in under 34 seconds with less than 1GB of GPU memory, whereas several existing methods either time out or fail numerically, and the completed baselines achieve lower precision and F1. Across completed comparable settings including the DREAM4 dataset, DACORD achieves the highest precision and F1 at high dimensional situation.

1 INTRODUCTION

Score-based causal discovery via continuous optimization has achieved remarkable progress. Zheng et al. (2018) reformulated the Directed Acyclic Graph (DAG) learning problem as continuous constrained optimization using a differentiable acyclicity loss. These methods have been applied to diverse domains such as gene regulatory network inference, incorporating Large Language Models, and economic policy evaluation (Glymour et al., 2019; Lee et al., 2025; Huber, 2024).

However, these methods assume stationarity, meaning that all observations belong to a single regime: a contiguous interval over which the causal structure remains constant (Scheffer et al., 2012). In practice, causal mechanisms often change over time so that multiple regimes can arise, leading to so–called non-stationarity as illustrated in Figure 1. Non-stationarity becomes important in long-horizon scenarios arising in domains such as economics (e.g. structural breaks in the Phillips curve) and epidemics (e.g. COVID-19) (Jung & Shah, 2015; Ahmed & Sarkodie, 2021).

Furthermore, existing continuous optimization approaches face scalability challenges that become severe in the non-stationary setting. The augmented Lagrangian scheme demands nested optimization loops: an outer loop that progressively increases the penalty parameter ρ, an inner loop that runs L-BFGS until convergence and a line search within each L-BFGS step that re-evaluates the matrix exponential multiple times. As ρ grows, the conditioning of the subproblem deteriorates, requiring more inner iterations (Ng et al., 2022). When extended to non-stationary settings with T time steps and R regimes, both the memory and the per-iteration cost can lead to out-of-memory errors.

These observations motivate two research questions that guide our work: **RQ1 (Non-stationarity).** *How can we discover time-varying causal structures?* **RQ2 (Scalability).** *How can we maintain computational feasibility as the number of variables d grows and the number of regimes R increases?*

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

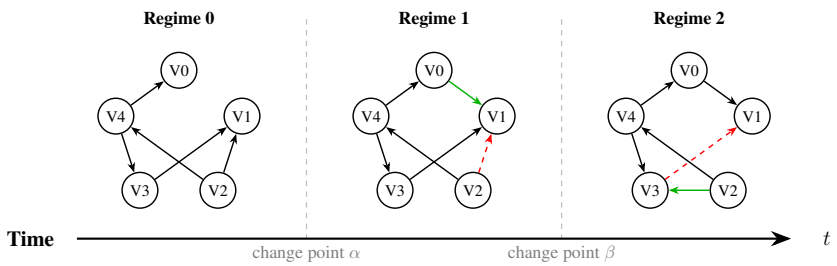

Figure 1: Non-stationary example: Ground-truth causal DAGs across three regimes in our dataset.

Non-stationarity inherently multiplies the problem size: R regimes across T time windows require estimating T adjacency matrices simultaneously. For instance, causal discovery in material fabrication involves hundreds of process variables; coupling this with regime-varying structure renders the problem effectively intractable (Keil, 2017; Barakati et al., 2025).

To address both research questions, we propose DACORD: **DA**G via **C**onvex **O**ptimization with **R**egime **D**etection, a two-stage framework. For RQ1, we formulate time-varying DAG learning as a convex program with a total variation (TV) penalty applied through a linear time-difference operator. For RQ2, we solve this convex formulation using the primal-dual algorithm, which operates in a single optimization loop with fixed step sizes. Our contributions are as follows:

- **TV-regularized non-stationary causal discovery with automatic change-point detection (RQ1).** We incorporate TV regularization through a linear time-difference operator K, enabling joint estimation of regime-specific DAGs and automatic change-point detection from the piecewise constancy of the primal solution. Empirically, DACORD achieves strong regime accuracy across d and attains the highest graph precision and F1.
- **Convex single-loop optimization for scalable causal discovery (RQ2).**
  We decompose non-stationary DAG learning into a convex TV-regularized structure estimation and a regime-consistent DAG projection. This eliminates the nested augmented Lagrangian scheme, line search, and penalty scheduling, yielding 20–40$\times$ wall-clock speedup at the same asymptotic complexity $O(d^3)$. At d=200, R=10, DACORD completes in 33 s with 824 MB GPU memory, while several baselines either time out or fail numerically, and the remaining completed baselines achieve substantially lower accuracy.

2 Related Work

2.1 Causal discovery

Constraint-based methods such as PCMCI (Runge et al., 2019) perform causal discovery via conditional independence tests and the PC algorithm (Spirtes et al., 2000). CD-NOD (Huang et al., 2020) detects mechanism changes via a surrogate-variable formulation and constraint-based testing. Score-based methods reformulate structure learning as continuous optimization. NOTEARS introduced a differentiable constraint (Zheng et al., 2018), followed by extensions to temporal and nonlinear models (Pamfil et al., 2020; Zheng et al., 2020). We build on fixed-basis regression while deferring acyclicity enforcement, allowing the coefficient-estimation stage to remain convex. DiffAN (Sanchez et al., 2023), NTS-NOTEARS (Sun et al., 2023), and kernel-based SVARM (Shen et al., 2019; Marques et al., 2020) capture nonlinear dependencies. DyCAST (Cheng et al., 2025) learns time-varying causal structures through neural networks. However, these continuous optimization approaches face scalability limitations that become problematic in non-stationary settings. The augmented Lagrangian formulation requires nested optimization loops, and extending this framework to track multiple time-varying causal graphs incurs prohibitive computational and memory costs (Lee et al., 2019).

When the ground-truth DAG $\mathcal{G}^*$ changes over time, the observation period is partitioned by *change points* $\{\tau_1, \ldots, \tau_{R-1}\}$ into R *regimes*. A regime is a maximal contiguous block of observations that share the same DAG (Scheffer et al., 2012). RPCMCI (Saggioro et al., 2020) extends PCMCI by alternating an optimization-based regime-learning. SpaceTime (Mameche et al., 2025) extends score-

based learning to spatiotemporal settings with regime-dependent structure. FANTOM (Rahmani & Frossard, 2025) tracks evolving graphs through continuous optimization.

Despite these advances, non-stationary causal discovery remains challenging because segmentation and structure learning are coupled. With d variables, T windows, and R regimes the learner must estimate a set of regime-specific DAGs $\{\mathcal{G}_r\}_{r=1}^{R}$, while also identifying the change points. Increasing R expands the number of graph parameters while reducing the effective sample size available within each regime, creating a trade-off between capturing local mechanism changes and obtaining reliable structure estimates. This trade-off becomes more pronounced in high-dimensional settings or when many candidate change points and regimes must be considered as shown in Table 2.

2.2 Primal–dual problem

Primal–dual methods jointly update the primal variable and the dual variable to approximately satisfy the optimality conditions of a nonsmooth convex optimization (Boyd & Vandenberghe, 2004; Hajinezhad & Hong, 2019). One representative example is the Chambolle–Pock algorithm, which solves problems of the form $\min_x\ G(x) + F(Kx)$, where F and G are convex functions and K is a linear operator (Chambolle & Pock, 2011). By introducing a dual variable for the linearly composed term $F(Kx)$, the problem is cast as a saddle-point problem and solved via alternating primal and dual proximal updates with an extrapolation step. This makes the method particularly suitable for nonsmooth regularizers such as total variation.

The Chambolle–Pock algorithm (Chambolle & Pock, 2011) solves problems of the form $\min_w\ g(w)+\tilde{f}(\tilde{K}w)$ by alternating proximal updates on the primal variable w and dual variable y:

$$y^{k+1} = \mathrm{prox}_{\sigma\tilde{f}^*}(y^k+\sigma\tilde{K}\bar{w}^k),\ w^{k+1} = \mathrm{prox}_{\tau g}(w^k-\tau\tilde{K}^*y^{k+1}),\ \bar{w}^{k+1} = w^{k+1}+\theta(w^{k+1}-w^k).$$

The algorithm is efficient when prox_g and $\mathrm{prox}_{\tilde{f}^*}$ both admit closed-form solutions. Such methods have been used in image processing, inverse problems, and causal discovery (Yang et al., 2020).

3 DACORD: DAG via Convex Optimization with Regime Detection

3.1 Problem setting

Consider data consisting of d variables with N observations observed over time, partitioned into $T = \lfloor N/n_w \rfloor$ consecutive windows of size n_w. The causal relationships among the variables are encoded by a DAG, where an edge $i \to j$ indicates that variable i is a direct cause of variable j. Notation is summarized in Section A. In many real-world systems, the causal structure does not remain fixed throughout the entire observation period (Alvarez-Ramirez et al., 2012). The observations are partitioned into R regimes $\mathcal{R}_1, \ldots, \mathcal{R}_R$, each governed by a distinct DAG $\mathcal{G}_r$, where adjacent regimes have different structures: $\mathcal{G}_r \neq \mathcal{G}_{r+1}$ (Rahmani & Frossard, 2025). Neither the number of regimes R, nor the regime boundaries, nor the DAGs $\{\mathcal{G}_r\}_{r=1}^{R}$ are known.

Goal. Given the ordered observations $\{x^{(i)}\}_{i=1}^{N}$, simultaneously recover: (i) the number of regimes R and their temporal boundaries, and (ii) the DAG $\mathcal{G}_r$ for each regime r.

3.2 DACORD

Figure 2 illustrates the overall framework of DACORD. We first show the computational bottleneck of existing methods, and how our formulation resolves it while enabling time-varying structure learning.

Computational bottleneck of existing methods. Standard continuous DAG learning methods solve a penalized regression subject to an acyclicity constraint $h(W) = 0$ (Pamfil et al., 2020):

$$\min_{W}\ \frac{1}{2n_w}\|X - \Phi W\|_F^2 + \lambda_1\|W\|_1 \quad \text{s.t.} \quad h(W) = 0. \tag{1}$$

This is solved via the augmented Lagrangian method, which requires nested optimization loops: an outer loop that increases the penalty parameter ρ, an inner loop running a quasi-Newton solver, and line search within each inner step that re-evaluates the acyclicity function, typically a matrix

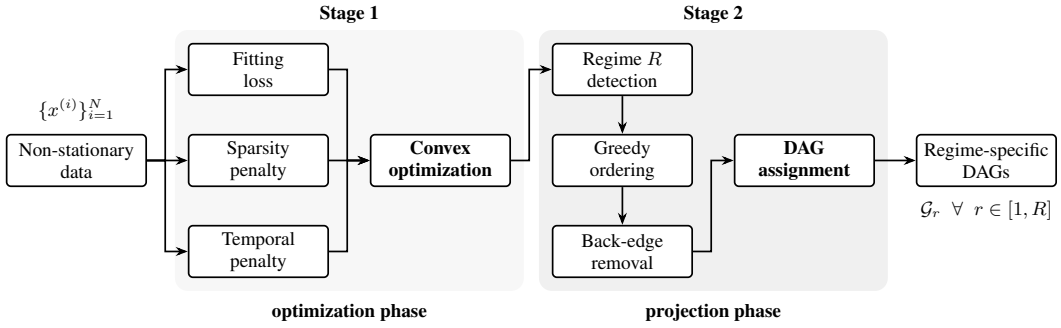

Figure 2: Two-stage optimization framework for causal discovery.

exponential involving matrix multiplications (Zheng et al., 2018). As ρ increases, it requires more inner iterations. When extended to settings with T time windows, both memory and per-iteration cost scale proportionally, quickly becoming prohibitive (Ng et al., 2020).

In Eq. 1, the loss and sparsity penalty are convex for a fixed feature map, whereas the acyclicity constraint is nonconvex (Bello et al., 2023). We defer acyclicity enforcement and jointly estimate the window-specific coefficients using a convex objective. The augmented operator ($\widetilde{K} = [I; K]$) combines sparsity and temporal regularization in a primal–dual splitting with closed-form proximal updates. Regime-level DAG construction follows this estimation stage.

Structural equation model and basis expansion. Within each window t, we model the observations via an additive noise structural equation model (SEM) with independent noise terms, $\varepsilon_{t,j}$:

$$X_{t,j} = \sum_{i \neq j} g_{ij}^{(t)}(X_{t,i}) + \varepsilon_{t,j}, \qquad g_{ij}^{(t)}(x) = \varphi(x)^\top \beta_{t,ij}, \quad j = 1, \ldots, d, \tag{2}$$

where $\varphi : \mathbb{R} \to \mathbb{R}^p$ is a fixed univariate basis function and $\beta_{t,ij} \in \mathbb{R}^p$ are learnable coefficients (Pamfil et al., 2020). Following the additive basis parameterization (Zheng et al., 2020), we model the window-specific additive contribution from variable i to variable j as $f_{t,ij}(x) = \varphi(x)^\top \beta_{t,ij}$. An edge $i \to j$ exists at window t and $\beta_{t,ij} \neq \mathbf{0}_p$ (Zheng et al., 2018); thus, Stage 1 reduces edge-score estimation to identifying the block-sparsity pattern of $W_t \in \mathbb{R}^{dp \times d}$, whose (i, j)-th block is $\beta_{t,ij}$. We set $\beta_{t,jj} = \mathbf{0}_p$ to exclude self-loops. With $\Phi_t = [\varphi(X_{t,1}), \ldots, \varphi(X_{t,d})] \in \mathbb{R}^{n_w \times dp}$, the SEM becomes $X_t \approx \Phi_t W_t + E_t$, and we minimize the loss, quadratic and convex in W_t for fixed Φ_t,

$$\ell(W_t; X_t) = \frac{1}{2n_w} \|X_t - \Phi_t W_t\|_F^2 \tag{3}$$

The choice of the fixed feature map φ controls the expressiveness of the model: $\varphi(x) = x$ recovers a linear SEM, whereas polynomial or RBF basis expansions allow the model to approximate nonlinear relationships (Bühlmann et al., 2014; Hastie et al., 2001). Because φ is constructed in preprocessing and is not optimized jointly with the model parameters, it does not alter convexity with respect to the coefficient variables. In practice, we adapt p to the number of variables d: a richer basis is used when d is small enough to keep the expanded parameter space computationally tractable.

Handling non-stationarity via total variation regularization. To discover time-varying causal structures, we partition the N observations into T consecutive windows of size n_w and assign a separate weight matrix W_t to each window $t \in [T]$ (Harchaoui & Lévy-Leduc, 2010). Regime boundaries can then be detected as transitions between adjacent windows where $W_t \neq W_{t+1}$.

To encode the prior that causal structures change infrequently, we introduce a total variation (TV) penalty through a linear time-difference operator $K = D \otimes I_m \in \mathbb{R}^{(T-1)m \times Tm}$, where D is the first-order finite-difference matrix and $m = dp \cdot d$:

$$\lambda_2 \|Kw\|_1 = \lambda_2 \sum_{t=1}^{T-1} \sum_{i,j} |W_{t+1,ij} - W_{t,ij}|, \qquad w = [\mathrm{vec}(W_1); \ldots; \mathrm{vec}(W_T)]. \tag{4}$$

The elementwise TV penalty promotes sparse coefficient differences across adjacent windows. We detect candidate regime boundaries using ($|\widehat{W}_{t+1} - \widehat{W}_t|_F$): the penalty encourages temporal fusion, while the statistic measures the estimated change (Tibshirani et al., 2005). Appendix B.6 discusses these distinct roles and Appendix E compares alternative temporal penalties.

Unified convex formulation. Combining these, we obtain the Stage 1 problem:

$$\min_{W_1,\dots,W_T} \underbrace{\sum_{t=1}^{T} \ell(W_t; X_t)}_{g(w)} + \underbrace{\lambda_1 \sum_{t=1}^{T} \Omega(W_t) + \lambda_2 \|Kw\|_1}_{\tilde{f}(\tilde{K}w)} \tag{5}$$

where $\tilde{K}$ couples both penalties into a single composite term $\tilde{f}$, and the sparsity penalty is defined as

$$\Omega(W_t) = \begin{cases} \|W_t\|_1 & (p = 1), \\ \sum_{i,j} \|\beta_{t,ij}\|_2 & (p > 1, \text{ group lasso}), \end{cases} \tag{6}$$

and $\|\tilde{K}\|^2 = 1 + \|K\|^2 < 5$ is bounded independently of all dimensions of the problem (Yuan & Lin, 2006; Simon et al., 2013). This formulation addresses **RQ1** via the TV penalty (Tibshirani et al., 2005; Bleakley & Vert, 2011) and **RQ2** via the resulting single-loop structure (Chambolle & Pock, 2011). Thus, non-stationarity handling and scalable optimization are coupled through the same operator-splitting framework, rather than being treated as separate procedures.

The splitting between g and $\tilde{f}$ is chosen for computational reasons. We place the data-fidelity term in g and the sparsity and TV penalties in $\tilde{f}$, so that the primal proximal step remains a closed-form quadratic update, while the dual proximal steps reduce to element-wise clipping or block-wise projection. This splitting does not change the minimizer of the composite objective; it only determines how the proximal updates are computed. If the sparsity penalty were included in g, each primal update would require solving a nonsmooth penalized quadratic subproblem.

The augmented operator $\tilde{K} = [I; K]$ yields two dual blocks: y_1 associated with sparsity and y_2 associated with temporal differences. These dual variables are useful optimization diagnostics, while regime boundaries are primarily inferred from the support of the primal differences $K\hat{w}$.

Thus, Stage 1 satisfies the Chambolle–Pock requirements (Chambolle & Pock, 2011; 2016): $\text{prox}_{\tau g}$ is closed form via the precomputed $(I + \tau H_t)^{-1}$, $\text{prox}_{\sigma \tilde{f}^*}$ is clipping or block projection for $p > 1$, and $\|\tilde{K}\|^2 < 5$ yields dimension-independent step sizes. See Section B.2 for details.

Stage 2: Regime-level DAG construction. Stage 1 outputs coefficient matrices $\hat{W}_t$, which define directed edge scores. After identifying candidate regime boundaries, we average the coefficients within each estimated regime and construct a DAG using greedy ordering and back-edge removal.

Step 1: Regime detection. Using the primal solution from Stage 1, we partition the T windows into $\hat{R}$ regimes $\mathcal{T}_1, \dots, \mathcal{T}_{\hat{R}}$, where regime boundaries are identified by $\{t : \|\hat{W}_{t+1} - \hat{W}_t\|_F > \epsilon_{\text{cp}}\}$.

Step 2: Regime averaging. For each regime r, we compute the average weight matrix $\bar{W}_r = \frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} \hat{W}_t$. This averaging defines a single coefficient matrix for DAG construction within each estimated regime, aggregating the window-specific estimates.

Step 3: Greedy ordering and back-edge removal. For each regime r, we compute the edge score $s(i \to j) = \|\bar{\beta}_{r,ij}\|_2$, or $s(i \to j) = |\bar{W}_{r,ij}|$ when $p = 1$. We compute the net flow $\text{flow}(v) = \sum_j s(v \to j) - \sum_j s(j \to v)$, construct a vertex ordering greedily, and remove all edges violating the ordering to obtain the DAG $\hat{\mathcal{G}}_r$ (Demetrescu & Finocchi, 2003; Baharev et al., 2021).

Step 4: Regime-consistent assignment. The resulting DAG $\hat{G}_r$ is assigned to all windows within the estimated regime T_r. This regime-level assignment is consistent with the fused-lasso/TV formulation, which promotes piecewise-constant trajectories by sparsifying temporal differences; regime boundaries are therefore inferred from the support of $K\hat{w}$ after thresholding (Tibshirani et al., 2005; Harchaoui & Lévy-Leduc, 2010). Thus, after segmentation, a single DAG per regime provides the natural regime-consistent output. By applying the greedy FAS projection only to the R regime

averages rather than to all T window-level estimates, the DAG-projection and storage costs are reduced from $O(Td^2)$ to $O(\hat{R}d^2)$ for dense edge-score matrices.

Empirical convergence. Figure 3 shows the relative objective residual across iterations for representative configurations. The residual stabilizes within 400 iterations, providing an empirical convergence diagnostic consistent with the Stage 1 convergence result. The extension to time-lagged effects ($L > 0$) is presented in Appendix C. The full algorithm is given in Algorithm 1 in Section F. Each DACORD iteration requires a single matrix multiplication per window, applying the precomputed $(I + \tau H_t)^{-1}$ to the right-hand side at cost $O(d^3)$ per window and $O(Td^3)$ in total.

3.3 Theoretical analysis

We establish convergence of the convex estimation stage. The statistical interpretation of its output and the conditions for DAG construction are given in Section B. We assume causal sufficiency (Assumption 1), an additive noise SEM with i.i.d. noise (Assumption 2), faithfulness (Assumption 3), faithful basis representation (Assumption 4), and signal separation between regimes (Assumption 5).

Theorem 3.1 (Convergence)**.** *The Stage 1 problem Eq. 5 is a convex composite optimization problem with an equivalent saddle-point formulation. Under the Chambolle–Pock step-size condition, the ergodic iterates of Algorithm 1 satisfy the standard saddle-point gap bound with rate $O(1/N)$ and converge to a minimizer of the convex Stage 1 objective.*

If, in addition, g is μ-strongly convex on the reduced parameter space, the accelerated Chambolle–Pock variant yields the standard accelerated ergodic rate (Chambolle & Pock, 2016). Existing methods solve nonconvex programs and guarantee convergence to stationary points only (Zheng et al., 2018). The step-size admissibility does not deteriorate with T, d, or p because $\|\tilde{K}\|^2 < 5$; faster rates additionally depend on curvature constants, which may depend on the design.

Proposition 3.2 (Population DAG target under standard SEM classes)**.** *Under additive noise SEMs, a unique DAG is determined by the joint population distribution in each of the following settings: linear non-Gaussian (Shimizu et al., 2006), linear Gaussian with equal error variance (Peters & Bühlmann, 2013), and restricted nonlinear additive noise under causal minimality (Peters et al., 2014). For $L \geq 1$, the contemporaneous and lagged structures are jointly determined in linear non-Gaussian SVAR models (Hyvärinen et al., 2010).*

This proposition establishes that a unique population DAG target exists under the stated SEM classes. Stage 1 estimates directed dependencies that may include non-parent edges, and Stage 2 constructs a DAG using greedy ordering and back-edge removal. Within a correctly identified regime, Lemma B.3 gives sufficient net-flow margin and estimation-error conditions for Stage 2 to preserve true edges present in its input.

Proposition 3.3 (TV-induced regime separation)**.** *Let $\{W_t^\circ\}_{t=1}^T$ be a piecewise-constant reference coefficient sequence with nonempty boundary set $\mathcal{B}^\circ = \{t \in \{1, \dots, T-1\} : W_{t+1}^\circ \neq W_t^\circ\}$, and minimum jump size $\Delta_{\mathrm{reg}} = \min_{t \in \mathcal{B}^\circ} \|W_{t+1}^\circ - W_t^\circ\|_F$. Suppose the Stage-1 estimates satisfy $\max_t \|\widehat{W}_t - W_t^\circ\|_F \leq e$ for some $e \geq 0$. If $\Delta_{\mathrm{reg}} > 4e$, then any threshold satisfying $2e < \epsilon_{\mathrm{cp}} < \Delta_{\mathrm{reg}} - 2e$ recovers the reference boundary set exactly: $\widehat{\mathcal{B}} = \{t : \|\widehat{W}_{t+1} - \widehat{W}_t\|_F > \epsilon_{\mathrm{cp}}\} = \mathcal{B}^\circ$.*

Regime boundaries are detected by thresholding the primal differences $\|\hat{W}_{t+1} - \hat{W}_t\|_F$. The proposition characterizes the separation required for boundary recovery by thresholding coefficient differences. Section B.4 separates the population separation intuition from finite-sample fluctuation.

4 Experiments

4.1 Experimental setup

Dataset. Due to the absence of reliable ground-truth causal graphs in real-world non-stationary settings, we primarily evaluate on synthetic data under both linear and nonlinear SEMs. We additionally evaluate on a DREAM4-based benchmark (Greenfield et al., 2010) with five successive regimes (Table 3). Appendix E provides the construction and case-study details and S&P sector analysis.

The causal graph is sampled from an Erdős–Rényi model with $d \in \{5, 10, 20, 50, 100, 200\}$ nodes (Lopez et al., 2022). For each generated edge, the edge weight is uniformly sampled from a specified range (e.g., $[-2, -0.5] \cup [0.5, 2]$) with equal probability (0.5). Each regime consists of 250 samples (500 for $d = 200$); at regime transitions, 20% of the edges are randomly modified (10% added, 10% removed). The data are generated by a SEM whose causal structure may change over time:

$$x_j^{(i)} = h_j^{(i)}\left(x^{(i)}_{\mathrm{pa}^{(0)}(i,j)}\right) + \sum_{\ell=1}^{L} f_j^{(i,\ell)}\left(x^{(i-\ell)}_{\mathrm{pa}^{(\ell)}(i,j)}\right) + \varepsilon_j^{(i)}, \quad j = 1, \ldots, d, \tag{7}$$

where $\mathrm{pa}^{(0)}(i,j)$ denotes the contemporaneous parents of variable j at observation i, $\mathrm{pa}^{(\ell)}(i,j)$ for $\ell \geq 1$ denotes the lag-ℓ parents, and $\varepsilon_j^{(i)}$ is independent additive Gaussian noise ($\sigma = 1.0$).

Metrics. We evaluate performance along two axes: regime segmentation and causal structure learning. For segmentation, we report regime accuracy: predicted regime labels are matched to ground-truth regimes, and accuracy is the fraction of time steps correctly classified under this optimal assignment. For causal structure learning, we report Precision, Recall, and F1-score. Additional metrics and details are described in Section D.

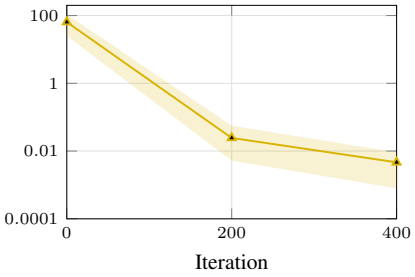

Figure 3: Convergence of the relative objective residual over iterations. Solid lines report the median and shaded regions indicate the interquartile range.

Hyperparameters. All methods are evaluated under a common protocol with a fixed edge-selection threshold of 0.1 and up to 20 independent random seeds. We aligned hyperparameter configurations across methods where possible, but several baselines encountered out of memory errors or excessive runtimes in high-dimensional settings as shown in Table 2; memory-sensitive hyperparameters were adaptively adjusted by d while keeping each method close to its recommended configuration. DACORD is fully deterministic. It uses no mini-batch sampling, dropout, or random initialization. We evaluate scalability by varying d and the number of regimes R, and further analyze the sensitivity of DACORD and other baselines to regularization parameters, window size, basis selection, and regime identification accuracy. Detailed hyperparameters for all baselines are in Table 5.

4.2 Experimental results

Table 1: Precision (P), Recall (R), and F1 across d and R. Values are means over nonlinear datasets. "× too slow" = method did not complete within budget (over 10 hours per run).

Method	d	R=3			R=5			R=10		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
DACORD	50	**0.490**	0.237	**0.318**	**0.504**	0.244	**0.327**	**0.509**	0.245	**0.328**
	100	**0.542**	0.289	**0.374**	**0.538**	0.289	**0.373**	**0.552**	0.291	**0.379**
	200	**0.661**	**0.364**	**0.469**	**0.687**	**0.382**	**0.490**	**0.694**	**0.374**	**0.486**
FANTOM	50	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)
	100	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)
	200	× NaN (numerical failure)								
RPCMCI	50	0.203 (0.01)	0.496 (0.12)	0.256 (0.05)	0.198 (0.02)	0.429 (0.11)	0.230 (0.04)	0.135 (0.11)	0.319 (0.25)	0.171 (0.13)
	100	0.147 (0.03)	0.315 (0.08)	0.155 (0.04)	0.141 (0.02)	0.450 (0.04)	0.196 (0.03)	0.092 (0.08)	0.262 (0.23)	0.126 (0.11)
	200	× too slow								
SpaceTime	50–200	× too slow (Gaussian process)								

Regime accuracy. DACORD achieves strong change-point detection accuracy, outperforming existing non-stationary causal discovery methods and even dedicated regime detection baselines (Figure 4 and Table 16 in Section E), despite requiring no prior knowledge of the number of regimes, unlike RPCMCI. A notable finding is that regime detection accuracy tends to improve or remain stable as the number of variables d increases, a pattern also observed across other methods. One might expect detection to degrade with growing d due to the expanding parameter space; the opposite trend calls for explanation.

This reflects a trade-off between signal aggregation and estimation difficulty (Soh & Chandrasekaran, 2017). In our setting, 20% of edges are modified at each regime transition, so the number of changed entries scales as $O(d)$ and the aggregate signal $\|W_{r+1} - W_r\|_F$ grows as $O(\sqrt{d})$. However, the total number of parameters scales as $O(d^2)$, increasing per-entry estimation noise. The best detection accuracy occurs where signal growth outpaces the estimation difficulty; beyond this point, the latter effect gradually dominates.

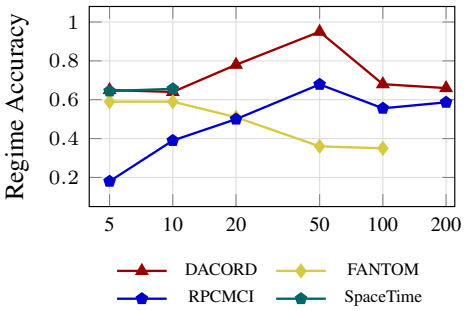

Figure 4: Regime Accuracy across d. Methods with truncated lines failed at higher d due to numerical failure or timeout.

Causal discovery. DACORD achieves the highest precision across all settings, while RPCMCI and FANTOM show higher recall at the cost of substantially lower precision. In particular, FANTOM achieves Recall = 1.0 in all completed settings, but its precision drops to .061 at $d = 100$, indicating that most predicted edges are false positives under the common threshold. False-positive edges correspond to spurious causal claims and can be problematic when the learned graph is used for interpretation or downstream intervention (Li & Wang, 2009). We emphasize precision while still reporting recall and F1 to capture the full precision–recall trade-off.

At $d = 200$, several baselines could not be stably evaluated despite extensive tuning. For FANTOM, we observe persistent NaN values during training, which is consistent with the numerical sensitivity of augmented-Lagrangian and flow-based likelihood optimization in high-dimensional settings. SpaceTime becomes prohibitively slow in our experiments, consistent with the cost of Gaussian-process-based conditional modeling.

Time and memory. Across all tested configurations, DACORD consistently requires less wall-clock time than the baselines, with comparable or lower peak memory usage. For baseline algorithms that could not be stably evaluated in the high-dimensional setting, we made extensive efforts to improve numerical stability, including reducing the batch size from 32 to 8, lowering the learning rate from 0.005 to 0.001, and decreasing the encoder/decoder hidden dimensions from $[100, 100]$ to $[64, 64]$. Nevertheless, NaN values persisted during training, and we therefore report these runs as unavailable due to numerical instability. Full tables and further empirical evaluations, including ablation studies on the acyclicity term and sensitivity analyses for hyperparameters (e.g., regularization, basis, window size, and time lag), as well as results on stationary and linear data, are provided in Section E.

Table 2: Wall-clock time and memory usage across d and R, mean (std) over linear/nonlinear × density. "—" = not measured. "× too slow" = method did not complete within budget.

		R=3			R=5			R=10		
Method	d	Time	GPU	CPU	Time	GPU	CPU	Time	GPU	CPU
DACORD	50	1.1s (0.02)	42MB(0.00)	798MB (0.7)	1.4s (0.05)	61MB(0.00)	804MB (1.1)	2.4s (0.11)	110MB(0.00)	820MB (1.0)
	100	2.1s (0.04)	78MB(0.00)	809MB (0.7)	3.0s (0.06)	116MB(0.00)	821MB (0.9)	5.6s (0.15)	216MB(0.00)	851MB (1.4)
	200	10.4s (0.16)	266MB(0.00)	866MB (0.6)	16.6s (0.15)	428MB(0.00)	903MB (1.2)	32.4s (0.27)	824MB(0.00)	1004MB (0.9)
RPCMCI	50	4.5m (0.46)	—	777MB (3.20)	16.7m (0.90)	—	782MB (0.17)	1.4h (0.24)	—	785MB (0.93)
	100	22.3m (4.39)	—	788MB (3.28)	56.8m (12.43)	—	797MB (7.95)	3.6h (0.61)	—	813MB (6.09)
	200	× too slow								
FANTOM	50	1.3m (0.09)	150MB (0.00)	1.7G (0.00)	2.1m (0.19)	240MB (0.00)	1.7G (0.00)	3.9m (0.27)	463MB (0.00)	1.7G (0.00)
	100	1.3m (0.14)	411MB (0.00)	1.7G (0.00)	2.0m (0.30)	672MB (0.00)	1.7G (0.00)	4.0m (0.25)	1.3G (0.00)	1.7G (0.00)
	200	× NaN (numerical failure)								
SpaceTime	50–200	× too slow (Gaussian process)								

DREAM4 evaluation. Because real non-stationary datasets lack a reference graph against which methods can be compared, we additionally evaluate on a non-stationary benchmark constructed by concatenating DREAM4, a stationary semi-synthetic dataset, so that successive segments follow different gene-regulatory topologies. DACORD attains RegAcc of 1.00 at both dimensions and the highest reported graph F1 of 0.65 at $d = 100$, where oracle DYNOTEARS does not complete within the runtime budget (Table 3). Oracle DYNOTEARS achieves higher F1 at $d = 10$, but

Table 3: DREAM4-based evaluation with five successive regimes. Oracle methods receive ground-truth boundaries. TO denotes timeout. Bold indicates the highest reported F1 at each dimension.

	$d = 10$				$d = 100$			
Method	RegAcc	P	R	F1	RegAcc	P	R	F1
DACORD	1.00	0.49	0.37	0.41	1.00	0.67	0.63	**0.65**
CD-NOD	0.76	0.20	0.27	0.23	0.89	0.05	0.10	0.07
FANTOM	0.52	0.16	1.00	0.27	0.45	0.02	1.00	0.04
RPCMCI	0.28	0.19	0.83	0.31	0.58	0.04	0.48	0.07
SpaceTime		TO				TO		
DYNOTEARS (oracle)	–	0.68	0.68	**0.67**		TO		
NTS-NOTEARS (oracle)	–	0.16	0.30	0.21		TO		

DACORD's advantage in the 100-variable benchmark supports its use in higher-dimensional settings with unknown regime boundaries. Appendix E details the benchmark construction and evaluation.

5 Discussion

Performance. Our ablation with stationary causal discovery methods should be interpreted with care. Such methods can perform strongly in small-scale or single-regime settings, especially when they are applied with oracle regime partitions. However, this protocol removes the change-point detection component and provides a favorable comparison point rather than an end-to-end solution to non-stationary causal discovery. Our target setting is different. Regime boundaries are unknown, and both segmentation and regime-specific DAG estimation must be performed jointly in high-dimensional time series. In this setting, repeatedly applying stationary solvers becomes increasingly costly as the number of variables and regimes grows. The results in Section E show that, compared with non-stationary baselines, DACORD achieves a favorable accuracy–runtime–memory trade-off.

Time and memory complexity. Each iteration is dominated by the primal step: a single matrix multiplication with the precomputed $(I + \tau H_t)^{-1}$, costing $O(d^3)$ per window and $O(Td^3)$ in total. While this asymptotic cost matches existing methods, three structural advantages yield a 20–40$\times$ practical speedup: (1) a single matrix multiplication versus $\sim$26 for the matrix exponential and its gradient; (2) no line search, since step sizes are fixed a priori by $\|\tilde{K}\|$; (3) a single optimization loop without nested penalty scheduling.

In contrast, existing methods cost $O(n_w d^2 + d^3)$ per inner L-BFGS iteration, where the $n_w d^2$ term arises from the loss gradient (absorbed into our one-time precomputation) and the d^3 term arises from the matrix exponential evaluated via Padé approximation with scaling and squaring, requiring on the order of ten matrix multiplications for both forward and backward passes, compared with a single multiplication in our method. Moreover, existing methods require nested loops (augmented Lagrangian $\times$ L-BFGS $\times$ line search), where each line search trial re-evaluates the full objective at cost $O(n_w d^2 + d^3)$. Our method eliminates all three sources of overhead: the loss gradient is precomputed, the acyclicity function is never evaluated, and the step sizes are fixed a priori.

6 Conclusion

Continuous optimization methods for DAG learning face two key limitations: stationarity and scalability in high-dimensional settings. We presented DACORD, a two-stage framework that addresses both limitations. By deferring acyclicity to a post-processing step, we reformulated non-stationary causal discovery as a convex program with ℓ_1 sparsity and total variation temporal regularization. We solve the resulting convex program via a primal-dual algorithm that operates in a single loop, eliminating the nested optimization, line search, and matrix exponential evaluations of existing methods. The TV penalty promotes piecewise constancy of the causal structure and detects regime boundaries through the primal solution. Experiments on synthetic data demonstrate that DACORD achieves competitive or superior accuracy while scaling to settings ($d = 200$, $R = 10$) where several existing methods fail numerically or exceed practical time limits.

ETHICS STATEMENT

This study uses synthetic SEM data, a DREAM4-based benchmark, and aggregated financial market data, S&P dataset. It involves no human-subject experiments or personally identifiable information. Also the real dataset is the publicly released the DREAM4 dataset and S&P which contain no personally identifiable information. We foresee no negative societal impacts specific to this work.

REPRODUCIBILITY STATEMENT

To support the reproducibility of this work, we will publicly release the complete source code used to conduct the experiments upon publication. The released materials will include the implementation of the proposed method, scripts for data preprocessing and training, evaluation code, configuration files, and instructions for reproducing the main experimental results reported in this paper. We also provide details of the experimental environment, including software dependencies, library versions, hardware specifications, and random seeds in Appendix D. These materials are intended to enable other researchers to reproduce our experimental setup and independently verify the reported results.

AI USE STATEMENT

AI tools were used to improve the writing, organization, and readability of this manuscript. They were also used to identify potential logical issues in the proposed ideas and proofs. During the experimental process, AI tools were also used for tasks, such as assisting with code implementation, organizing code structure, and refining comments.

REFERENCES

Maruf Yakubu Ahmed and Samuel Asumadu Sarkodie. Covid-19 pandemic and economic policy uncertainty regimes affect commodity market volatility. *Resources policy*, 74:102303, 2021.

Jose Alvarez-Ramirez, Eduardo Rodriguez, Esteban Martina, and Carlos Ibarra-Valdez. Cyclical behavior of crude oil markets and economic recessions in the period 1986–2010. *Technological Forecasting and Social Change*, 79(1):47–58, 2012.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Ali Baharev, Hermann Schichl, Arnold Neumaier, and Tobias Achterberg. An exact method for the minimum feedback arc set problem. *ACM J. Exp. Algorithmics*, 26, April 2021. ISSN 1084-6654. doi: 10.1145/3446429. URL https://doi.org/10.1145/3446429.

Kamyar Barakati, Aleksander Molak, Chris Nelson, Xiaohang Zhang, Ichiro Takeuchi, and Sergei V Kalinin. Causal discovery from data assisted by large language models. *Applied Physics Letters*, 127(12), 2025.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Dagma: Learning dags via m-matrices and a log-determinant acyclicity characterization, 2023. URL https://arxiv.org/abs/2209.08037.

Monica Billio, Mila Getmansky, Andrew W. Lo, and Loriana Pelizzon. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, 104(3):535–559, 2012. ISSN 0304-405X. doi: https://doi.org/10.1016/j.jfineco.2011.12.010. URL https://www.sciencedirect.com/science/article/pii/S0304405X11002868. Market Institutions, Financial Market Risks and Financial Crisis.

Kevin Bleakley and Jean-Philippe Vert. The group fused lasso for multiple change-point detection. *Arxiv preprint arXiv:1106.4199*, 06 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014. ISSN 00905364. URL http://www.jstor.org/stable/43556502.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, May 2011. doi: 10.1007/s10851-010-0251-1. URL https://doi.org/10.1007/s10851-010-0251-1.

Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1):253–287, Sep 2016. doi: 10.1007/s10107-015-0957-3. URL https://doi.org/10.1007/s10107-015-0957-3.

Yue Cheng, Bochen Lyu, Weiwei Xing, and Zhanxing Zhu. Dycast: Learning dynamic causal structure from time series. In *The Thirteenth International Conference on Learning Representations*, 2025.

Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications*, 158(2): 460–479, 2013.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

Camil Demetrescu and Irene Finocchi. Combinatorial algorithms for feedback problems in directed graphs. *Inf. Process. Lett.*, 86(3):129–136, May 2003. ISSN 0020-0190. doi: 10.1016/S0020-0190(02)00491-X. URL https://doi.org/10.1016/S0020-0190(02)00491-X.

Archibald Fraikin, Adrien Bennetot, and Stéphanie Allassonnière. T-rep: Representation learning for time series using time-embeddings. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

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

Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10): e13397, 2010.

Davood Hajinezhad and Mingyi Hong. Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization. *Mathematical Programming*, 176(1):207–245, Jul 2019. doi: 10.1007/s10107-019-01365-4. URL https://doi.org/10.1007/s10107-019-01365-4.

Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010. ISSN 01621459. URL http://www.jstor.org/stable/27920180.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

Martin Huber. An introduction to causal discovery. *Swiss Journal of Economics and Statistics*, 160 (1):14, 2024.

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.

Kenneth Jung and Nigam H. Shah. Implications of non-stationarity on predictive modeling using ehrs. *Journal of Biomedical Informatics*, 58:168–174, 2015. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2015.10.006. URL https://www.sciencedirect.com/science/article/pii/S1532046415002282.

Sophia Keil. Design of a cyber-physical production system for semiconductor manufacturing. In *Digitalization in Supply Chain Management and Logistics: Smart and Digital Solutions for an Industry 4.0 Environment*, volume 23 of *Proceedings of the Hamburg International Conference of Logistics (HICL)*, pp. 319–340. epubli GmbH, Berlin, Germany, 2017.

Ricardo Laborda and Jose Olmo. Volatility spillover between economic sectors in financial crisis prediction: Evidence spanning the great financial crisis and covid-19 pandemic. *Research in International Business and Finance*, 57:101402, 2021. ISSN 0275-5319. doi: https://doi.org/10.1016/j.ribaf.2021.101402. URL https://www.sciencedirect.com/science/article/pii/S0275531921000234.

Chanhui Lee, Juhyeon Kim, Yongjun Jeong, Yoonseok Yeom, Juhyun Lyu, Junghee Kim, Sangmin Lee, Sangjun Han, Hyeokjun Choe, Soyeon Park, Woohyung Lim, Sungbin Lim, and Sanghack Lee. On incorporating prior knowledge extracted from large language models into causal discovery. *IEEE Access*, 13:194691–194713, 2025. doi: 10.1109/ACCESS.2025.3626040.

Hao-Chih Lee, Matteo Danieletto, Riccardo Miotto, Sarah T. Cherng, and Joel T. Dudley. Scaling structural learning with no-bears to infer causal transcriptome networks, 2019. URL https://arxiv.org/abs/1911.00081.

Junning Li and Z. Jane Wang. Controlling the false discovery rate of the association/causality structure learned with the pc algorithm. *Journal of Machine Learning Research*, 10(17):475–514, 2009. URL http://jmlr.org/papers/v10/li09a.html.

Romain Lopez, Jan-Christian Huetter, Jonathan Pritchard, and Aviv Regev. Large-scale differentiable causal discovery of factor graphs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 19290–19303. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7a8fa1382ea068f3f402b72081df16be-Paper-Conference.pdf.

Sarah Mameche, Lénaïg Cornanguer, Urmi Ninad, and Jilles Vreeken. Spacetime: causal discovery from non-stationary time series. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i18.34136. URL https://doi.org/10.1609/aaai.v39i18.34136.

Antonio G. Marques, Santiago Segarra, and Gonzalo Mateos. Signal processing on directed graphs: The role of edge directionality when processing and learning from network data. *IEEE Signal Processing Magazine*, 37(6):99–116, 2020. doi: 10.1109/MSP.2020.3014597.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. In *Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 849–856, Cambridge, MA, USA, 2001. MIT Press.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Ignavier Ng, Sebastien Lachapelle, Nan Rosemary Ke, Simon Lacoste-Julien, and Kun Zhang. On the convergence of continuous constrained optimization for structure learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8176–8198. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/ng22b.html.

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.

J. Peters and P. Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, November 2013. ISSN 0006-3444. doi: 10.1093/biomet/ast043. URL http://dx.doi.org/10.1093/biomet/ast043.

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(58):2009–2053, 2014. URL http://jmlr.org/papers/v15/peters14a.html.

Abdellah Rahmani and Pascal Frossard. Flow based approach for dynamic temporal causal models with non-gaussian or heteroscedastic noises, 2025. URL https://arxiv.org/abs/2506.17065.

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, 2019. doi: 10.1126/sciadv.aau4996. URL https://www.science.org/doi/abs/10.1126/sciadv.aau4996.

Elena Saggioro, Jana de Wiljes, Marlene Kretschmer, and Jakob Runge. Reconstructing regime-dependent causal relationships from observational time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(11), November 2020. ISSN 1089-7682. doi: 10.1063/5.0020538. URL http://dx.doi.org/10.1063/5.0020538.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10, 2015. URL https://api.semanticscholar.org/CorpusID:14081058.

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 (ICLR)*, 2023.

Marten Scheffer, Stephen R. Carpenter, Timothy M. Lenton, Jordi Bascompte, William Brock, Vasilis Dakos, Johan van de Koppel, Ingrid A. van de Leemput, Simon A. Levin, Egbert H. van Nes, Mercedes Pascual, and John Vandermeer. Anticipating critical transitions. *Science*, 338(6105):344–348, 2012. doi: 10.1126/science.1225244. URL https://www.science.org/doi/abs/10.1126/science.1225244.

Yanning Shen, Georgios B. Giannakis, and Brian Baingana. Nonlinear structural vector autoregressive models with application to directed brain networks. *IEEE Transactions on Signal Processing*, 67(20):5325–5339, 2019. doi: 10.1109/TSP.2019.2940122.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.*, 7:2003–2030, December 2006. ISSN 1532-4435.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013. doi: 10.1080/10618600.2012.681250. URL https://doi.org/10.1080/10618600.2012.681250.

Yong Sheng Soh and Venkat Chandrasekaran. High-dimensional change-point estimation: Combining filtering with convex optimization. *Applied and Computational Harmonic Analysis*, 43(1):122–147, 2017. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2015.11.003. URL https://www.sciencedirect.com/science/article/pii/S1063520315001542.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. Nts-notears: Learning nonparametric dbns with prior knowledge. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 1942–1964. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/sun23c.html.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. doi: https://doi.org/10.1111/j.1467-9868.2005.00490.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00490.x.

Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Admm-csnet: A deep learning approach for image compressive sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(3): 521–538, 2020. doi: 10.1109/TPAMI.2018.2883941.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 02 2006. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2005.00532.x. URL https://doi.org/10.1111/j.1467-9868.2005.00532.x.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987, Jun. 2022. doi: 10.1609/aaai.v36i8.20881. URL https://ojs.aaai.org/index.php/AAAI/article/view/20881.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3414–3425. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/zheng20a.html.

A Notations

Table 4: Summary of notation.

Symbol	Description
Data dimensions	
d	Number of observed variables
N	Total number of observations across all regimes
n_w	Window size: number of samples per window (hyperparameter)
T	Number of windows ($T = \lfloor N/n_w \rfloor$)
$t \in [T]$	Window index (not individual time points)
R	Number of latent regimes (unknown, to be discovered)
$\hat{R}$	Number of estimated regimes
Model dimensions	
p	Basis dimension per variable ($p = 1$ for linear)
L	Maximum time lag ($L = 0$ for contemporaneous only)
m	Vectorized weight dimension per window: $m = dp \cdot d$ for $L = 0$
Data	
$X_t \in \mathbb{R}^{n_w \times d}$	Observed data in window t
$\Phi_t \in \mathbb{R}^{n_w \times (L+1)dp}$	Design matrix (basis-expanded, with lags)

B Theoretical Analysis

B.1 Assumptions

Assumption 1 (Causal sufficiency)**.** *There are no unobserved common causes of any two or more observed variables.*

Assumption 2 (Additive noise SEM). *For each regime r and variable j: $X_{t,j} = \sum_{i \in \mathrm{pa}_r(j)} g_{ij}^{(r)}(X_{t,i}) + \varepsilon_{t,j}$, where the noise terms satisfy: (i) $\varepsilon_{t,j} \perp\!\!\!\perp X_{t,\mathrm{pa}_r(j)}$ (exogeneity), (ii) the noise terms $\epsilon_{t,j}$ are mutually independent across both observations and variables (noise i.i.d.), and (iii) $\mathrm{Var}(\varepsilon_{t,j}) > 0$ (non-degeneracy).*

Assumption 3 (Faithfulness). *For every true edge $i \to j$ in $\mathcal{G}_r$, $X_i \not\perp\!\!\!\perp X_j \mid X_{\mathrm{pa}_r(j)\setminus\{i\}}$.*

Assumption 4 (Faithful basis representation). *For every true edge $i \to j$, $g_{ij} \not\equiv 0$ and g_{ij} lies in the span of φ. (Automatically satisfied when $p = 1$.)*

Assumption 5 (Signal separation). *For each true regime transition: $\|W_r^{\mathrm{true}} - W_{r+1}^{\mathrm{true}}\|_F \geq \Delta_{\min}$.*

Remark B.1 (Causal Markov condition). *Assumptions 1 and 2 together imply the causal Markov condition and mutually independent noise: $\varepsilon_{t,j} \perp\!\!\!\perp \varepsilon_{t,k}$ for $j \neq k$, since correlated errors would entail an unobserved common cause.*

B.2 Verification of Chambolle–Pock conditions

Condition (i): primal proximal. We enforce the no-self-loop constraint by optimizing over the reduced off-diagonal parameterization, in which the diagonal blocks $\beta_{t,jj}$ are excluded from the variables. For each window t and target variable j, let $\Phi_{t,-j}$ denote the design matrix obtained by removing the feature block corresponding to $X_{t,j}$. Then the proximal update decomposes over windows and targets as

$$\mathrm{prox}_{\tau g}(v_{t,-j,j}) = (I + \tau H_{t,j}^{(-j)})^{-1}(v_{t,-j,j} + \tau b_{t,j}^{(-j)}),$$

where

$$H_{t,j}^{(-j)} = \Phi_{t,-j}^{\top}\Phi_{t,-j}/n_w, \qquad b_{t,j}^{(-j)} = \Phi_{t,-j}^{\top}X_{t,j}/n_w.$$

The diagonal blocks are filled with zeros only when embedding the reduced solution back into the full coefficient matrix. Thus, the no-self-loop constraint is enforced by parameterization, and the proximal step remains exact.

Condition (ii): dual proximal. $\tilde{f}^*$ separates into independent components: the sparsity dual y_1 is updated by element-wise clipping to $[-\lambda_1, \lambda_1]$, and the TV dual y_2 by clipping to $[-\lambda_2, \lambda_2]$. Both are $O(m)$ with no inner iteration.

Condition (iii): bounded operator. $\|\tilde{K}\|^2 < 5$ independently of T, d, and p, so step sizes $\tau = \sigma = 1/\sqrt{5}$ are fixed a priori without data-dependent tuning.

All proximal operators admit closed-form solutions. Each iteration requires *no inner subroutine*—no matrix exponential, no line search, no penalty scheduling—yielding a deterministic algorithm.

B.3 Convergence

Theorem B.1 (Convergence — full statement). *The Stage 1 problem has the composite convex form $\min_w g(w) + \tilde{f}(\tilde{K}w)$. Let $\mathcal{L}(w, y) = g(w) + \langle \tilde{K}w, y\rangle - \tilde{f}^*(y)$. Under the Chambolle–Pock step-size condition $\tau\sigma\|\tilde{K}\|^2 \leq 1$, the ergodic iterates $(\bar{w}^N, \bar{y}^N)$ satisfy, for any w and y,*

$$\mathcal{L}(\bar{w}^N, y) - \mathcal{L}(w, \bar{y}^N) \leq \frac{\|w - w^0\|^2}{2\tau N} + \frac{\|y - y^0\|^2}{2\sigma N}.$$

Thus, the primal–dual saddle gap converges at rate $O(1/N)$. If g is additionally μ-strongly convex, the accelerated Chambolle–Pock variant yields the standard $O(1/N^2)$ ergodic rate.

The theorem establishes the standard ergodic saddle-point gap convergence of the Chambolle–Pock algorithm for the convex Stage 1 problem.

Proof. Algorithm 1 instantiates the standard proximal Chambolle–Pock scheme of (Chambolle & Pock, 2011), where both g and $\tilde{f}^*$ are handled via proximal steps. We verify the three conditions.

Condition 1: $\mathrm{prox}_{\tau g}$ is well-defined and closed-form. We enforce the no-self-loop constraint by optimizing over the reduced off-diagonal parameterization, in which the diagonal blocks $\beta_{t,jj}$ are excluded from the variables. For each target variable j, let $\Phi_{t,-j}$ denote the design matrix obtained

by removing the feature block corresponding to $X_{t,j}$, and let $v_{t,-j,j}$ collect the corresponding off-diagonal coefficients. Then the proximal update decomposes over windows and target variables as

$$\mathrm{prox}_{\tau g}(v_{t,-j,j}) = (I + \tau H_{t,j}^{(-j)})^{-1}(v_{t,-j,j} + \tau b_{t,j}^{(-j)}),$$

where

$$H_{t,j}^{(-j)} = \Phi_{t,-j}^{\top}\Phi_{t,-j}/n_w, \qquad b_{t,j}^{(-j)} = \Phi_{t,-j}^{\top}X_{t,j}/n_w.$$

The diagonal blocks are filled with zeros after embedding the reduced solution back into the full coefficient matrix. Thus, the no-self-loop constraint is enforced by parameterization, and the proximal update remains exact.

Condition 2: $\mathrm{prox}_{\sigma\tilde{f}^*}$ *is well-defined and closed-form.* $\tilde{f}(\tilde{K}w) = \lambda_1\Omega(w) + \lambda_2\|Kw\|_1$ is a sum of norms, hence convex, proper, and lsc. Its conjugate $\tilde{f}^*$ separates: for the ℓ_1 component ($p = 1$), $\mathrm{prox}_{\sigma\tilde{f}_1^*}$ is element-wise clipping to $[-\lambda_1, \lambda_1]$; for the group lasso component ($p > 1$), $\mathrm{prox}_{\sigma\tilde{f}_1^*}$ is block-wise projection onto ℓ_2 balls of radius λ_1. The TV dual proximal $\mathrm{prox}_{\sigma\tilde{f}_2^*}$ is element-wise clipping to $[-\lambda_2, \lambda_2]$. All dual proximals are closed-form.

Condition 3: $\tilde{K}$ *is a bounded linear operator.* $\tilde{K} = [I; K]$ where $K = D \otimes I_m$. The operator norm satisfies $\|\tilde{K}\|^2 = 1 + \|K\|^2$. For the first-order difference matrix $D \in \mathbb{R}^{(T-1)\times T}$, $D^{\top}D$ is the path-graph Laplacian with eigenvalues

$$\lambda_k = 2 - 2\cos(k\pi/T), \qquad k = 0, \ldots, T-1.$$

Hence,

$$\|D\|^2 = \lambda_{\max}(D^{\top}D) = 2 + 2\cos(\pi/T) < 4,$$

and therefore

$$\|\tilde{K}\|^2 = \|[I; K]\|^2 = 1 + \|K\|^2 < 5.$$

All three conditions are satisfied. Part (a) follows from Theorem 1 of (Chambolle & Pock, 2011): the standard proximal CP iteration with step sizes satisfying $\tau\sigma\|\tilde{K}\|^2 \le 1$ guarantees $O(1/N)$ ergodic convergence of the primal-dual gap. Part (b): when g is μ-strongly convex, the accelerated CP variant of (Chambolle & Pock, 2016) (Theorem 4) yields an $O(1/N^2)$ ergodic rate. The step-size admissibility depends on $\|\tilde{K}\|^2 < 5$, which is independent of T, d, and p. The accelerated rate constant additionally depends on μ, which may depend on the design matrices. □

Remark B.2 (Convergence diagnostic)**.** *The formal convergence result above is stated in terms of the ergodic saddle-point gap. In the experiments, we monitor the relative change in the primal objective* $|P^{k+1} - P^k|/|P^k|$ *as an empirical stopping criterion rather than evaluating the formal Fenchel duality gap.*

B.4 Population DAG target under standard SEM classes

Proposition B.2 (Population DAG target under standard SEM classes — full statement)**.** *Under Assumptions 1–4, the true DAG is identifiable from the joint population distribution in each of the following settings:*

- *(a)* ***Linear non-Gaussian*** *($p = 1$, $L = 0$): The DAG is identifiable by the LiNGAM result (Shimizu et al., 2006).*
- *(b)* ***Linear Gaussian with equal error variances*** *($p = 1$, $L = 0$): The DAG is identifiable from the joint Gaussian distribution (Peters & Bühlmann, 2013).*
- *(c)* ***Restricted nonlinear additive noise*** *($p > 1$, $L = 0$): Under causal minimality, the DAG is generically identifiable (Peters et al., 2014).*
- *(d)* ***With time lags*** *($L \ge 1$): For linear non-Gaussian SVAR models, the contemporaneous structure and structural lag matrices are jointly identifiable (Hyvärinen et al., 2010).*

Proof. Each part restates an existing result under the corresponding model class.

Part (a) follows from (Shimizu et al., 2006): in linear SEMs with non-Gaussian, mutually independent noise and no latent confounders (Assumptions 1–2), the DAG is uniquely determined from the joint distribution via independent component analysis.

Part (b) follows from Theorem 1 of (Peters & Bühlmann, 2013): in linear Gaussian SEMs where all error variances are equal, the DAG is identifiable from the joint Gaussian distribution.

Part (c) follows from Theorem 28 of (Peters et al., 2014): in restricted additive noise models satisfying causal minimality, the causal graph is generically identifiable from the joint distribution when the mechanisms are nonlinear.

Part (d) follows from (Hyvärinen et al., 2010): in linear non-Gaussian SVAR models, both contemporaneous and lagged structures are jointly identifiable from the population distribution of the observed time series.

In all cases, there exists a unique DAG consistent with the population distribution, which serves as the target that our two-stage method aims to recover. □

Remark B.3 (Relationship between identifiability and Stage 1 regression)**.** *The identifiability results above specify the population causal target. Stage 1 estimates directed candidate dependencies without enforcing acyclicity. Its output may contain non-parent edges and may omit weak true edges under finite-sample regularization. Thus, population identifiability alone does not establish support containment or recovery by the Stage 1 regression. Stage 2 enforces acyclicity, while Appendix B.5 gives an additional condition under which its ordering step preserves true input edges.*

Remark B.4 (Pooling within a regime)**.** *If the coefficients are shared across windows in a correctly identified regime, the corresponding data-fitting terms combine into a regression on the stacked window data. For equal window sizes, this is proportional to the pooled least-squares loss. This observation concerns the data-fitting term. The sparsity and between-regime TV penalties remain part of the objective. The resulting estimation accuracy also depends on the design, temporal dependence, and regularization bias.*

B.5 Stage-2 DAG Projection

Stage 1 estimates directed candidate dependencies rather than a DAG directly (Remark B.3). Stage 2 therefore serves a different role from the convex optimization: it maps the regime-level directed edge scores to an acyclic graph by extracting a greedy ordering and removing edges inconsistent with that ordering. We next clarify the criterion for evaluating this projection and give a sufficient condition under which it preserves the true DAG ordering.

Why the Stage-1 objective is not a projection criterion. Let $J(W)$ denote the Stage-1 objective in Eq. 5, and let $\widehat{W}$ be its global minimizer. By Theorem 3.1,

$$J(\widehat{W}) \le J(W) \qquad \text{for every feasible } W.$$

This inequality also holds for any DAG obtained after projection, as well as for the ground-truth DAG itself. Consequently, whether projection lowers the Stage-1 objective is not an appropriate criterion for evaluating its causal quality. As discussed in Remark B.3, the unconstrained population regression can assign nonzero coefficients to non-parent variables and thus favor a directed supergraph over the true DAG under the Stage-1 objective. The role of Stage 2 is instead to recover an acyclic representative of the population DAG target characterized in Proposition B.2.

We now give a sufficient condition for the greedy ordering used in Stage 2. Let $V_1 = V$ and let $\hat{v}_k$ denote the vertex selected by the greedy procedure at step k, with

$$V_{k+1} = V_k \setminus \{\hat{v}_k\}.$$

For a remaining vertex set V_k at greedy step k, define the population net-flow score

$$F_k^\star(v) = \sum_{j \in V_k} s^\star(v \to j) - \sum_{j \in V_k} s^\star(j \to v),$$

where $s^\star(i \to j)$ denotes the corresponding population edge score.

Lemma B.3 (Greedy ordering recovery under a net-flow margin)**.** *Let $G^\star$ be the true DAG within a regime. At every greedy step k, suppose that there exists a source v_k of the true induced subgraph on V_k such that*

$$F_k^\star(v_k) \geq F_k^\star(u) + \gamma$$

for every remaining non-source $u \in V_k$, for some margin $\gamma > 0$. Suppose further that the estimated net-flow scores satisfy

$$\max_{v \in V_k} \left| \widehat{F}_k(v) - F_k^\star(v) \right| < \frac{\gamma}{2}$$

at every greedy step. Then the greedy procedure selects a true source at each step. Consequently, the resulting ordering is a topological ordering of $G^\star$, and the subsequent back-edge removal deletes no true edge.

Proof. For any remaining non-source u,

$$\widehat{F}_k(v_k) > F_k^\star(v_k) - \frac{\gamma}{2} \geq F_k^\star(u) + \frac{\gamma}{2} > \widehat{F}_k(u).$$

Hence a source of the true induced subgraph attains a larger estimated net-flow score than every non-source and is selected by the greedy step. Removing a true source leaves the remaining induced graph acyclic, so the same argument applies recursively. Induction therefore yields a topological ordering of $G^\star$. Since every true edge is forward with respect to this ordering, the back-edge removal step deletes no true edge. □

Scope of the guarantee. Lemma B.3 is a sufficient ordering condition rather than an exact-support guarantee. In particular, it ensures that true edges are not removed when the estimated ordering is sufficiently accurate, but it does not imply that all spurious forward edges in the Stage-1 supergraph are eliminated. Exact support recovery would require additional conditions on the separation of true and spurious edge scores. We therefore evaluate the finite-sample structural effect of the projection empirically in Appendix F.

Connection to TV coupling. TV coupling and regime averaging aggregate information across windows before the ordering step. Within a correctly identified regime, this provides the motivation for using regime-level scores. Lemma B.3 applies when these scores satisfy its margin and estimation-error conditions. The empirical effect of regime-consistent processing is examined in the RC ablation in Table 6.

B.6 TV-INDUCED REGIME SEPARATION

Proposition B.4 (TV-induced regime recovery)**.** *Let $\{W_t^\circ\}_{t=1}^T$ be a piecewise-constant reference coefficient sequence with nonempty boundary set $\mathcal{B}^\circ = \{t \in \{1, \dots, T-1\} : W_{t+1}^\circ \neq W_t^\circ\}$, and minimum jump size $\Delta_{\mathrm{reg}} = \min_{t \in \mathcal{B}^\circ} \|W_{t+1}^\circ - W_t^\circ\|_F$. Suppose the Stage-1 estimates satisfy $\max_t \|\widehat{W}_t - W_t^\circ\|_F \leq e$ for some $e \geq 0$. If $\Delta_{\mathrm{reg}} > 4e$, then any threshold satisfying $2e < \epsilon_{\mathrm{cp}} < \Delta_{\mathrm{reg}} - 2e$ recovers the reference boundary set exactly: $\widehat{\mathcal{B}} = \{t : \|\widehat{W}_{t+1} - \widehat{W}_t\|_F > \epsilon_{\mathrm{cp}}\} = \mathcal{B}^\circ$.*

Proof. For $t \notin \mathcal{B}^\circ$, we have $W_{t+1}^\circ = W_t^\circ$. Hence,

$$\|\widehat{W}_{t+1} - \widehat{W}_t\|_F \leq \|\widehat{W}_{t+1} - W_{t+1}^\circ\|_F + \|\widehat{W}_t - W_t^\circ\|_F \leq 2e.$$

For $t \in \mathcal{B}^\circ$, the reverse triangle inequality gives

$$\|\widehat{W}_{t+1} - \widehat{W}_t\|_F \geq \|W_{t+1}^\circ - W_t^\circ\|_F - 2e \geq \Delta_{\mathrm{reg}} - 2e.$$

The prescribed threshold therefore excludes every non-boundary and includes every boundary. □

The lower bound suppresses spurious within-regime differences, while the upper bound prevents adjacent true regimes from being fused. When this interval is nonempty, regime boundaries can be inferred by thresholding $\|\hat{W}_{t+1} - \hat{W}_t\|_F$.

The lower bound is finite-sample in nature and reflects stochastic fluctuations of the windowed regression estimates. The upper bound is a signal-separation condition: if λ_2 is too large relative to the smallest structural change, TV regularization may merge adjacent regimes. A formal finite-sample localization theorem would require additional restricted-eigenvalue and concentration assumptions for the design-coupled multivariate regression. We therefore use this condition to motivate the TV-based regime recovery mechanism and verify the resulting behavior empirically.

Remark B.5 (Finite-sample behavior)**.** *In finite samples, the Stage 1 objective is a design-coupled multivariate regression with TV coupling, which is more complex than the scalar signal denoising setting analyzed in (Harchaoui & Lévy-Leduc, 2010). A formal finite-sample localization theorem would additionally require assumptions on the design regularity, restricted eigenvalues, and concentration for the windowed regressions. We verify empirically that change-point detection achieves high accuracy across all tested configurations (Section 4).*

Remark B.6 (Primal vs. dual change-point detection)**.** *The TV penalty encourages adjacent coefficient estimates to fuse. Primal differences are zero within an estimated fused block, but these blocks need not coincide with the underlying regimes. Under the conditions of Proposition B.4, thresholding the primal differences recovers the reference boundary set.*

A nonzero TV dual variable need not imply a nonzero primal difference. We therefore use primal differences for boundary detection and dual magnitudes as a complementary diagnostic.

Remark B.7 (Complementary role of y_1)**.** *The sparsity dual helps monitor optimization, while edges are selected from the estimated coefficients. A dual variable reaching its bound does not necessarily indicate an edge.*

C Time-Lagged Extension

The main text presents DACORD for the contemporaneous model ($L = 0$), where each variable depends only on other variables at the same time point. Here we extend the framework to incorporate lagged causal effects ($L \geq 1$), where variable j at time s may depend on variables at times $s - 1, \ldots, s - L$.

C.1 Data construction

For each observation at time index s within window t (where $s \geq L+1$), we construct the augmented feature vector by concatenating the basis-expanded current and lagged observations:

$$\Phi_t^{\text{aug}}[s,:] = \big[\underbrace{\varphi(x_{s,1}),\ldots,\varphi(x_{s,d})}_{\text{lag 0 (contemporaneous)}},\ \underbrace{\varphi(x_{s-1,1}),\ldots,\varphi(x_{s-1,d})}_{\text{lag 1}},\ \ldots,\ \underbrace{\varphi(x_{s-L,1}),\ldots,\varphi(x_{s-L,d})}_{\text{lag } L}\big], \tag{8}$$

yielding $\Phi_t^{\text{aug}} \in \mathbb{R}^{(n_w-L)\times(L+1)dp}$. The first L observations in each window are discarded as they lack sufficient history, giving $n_w' = n_w - L$ usable samples per window. The target matrix remains $Y_t = X_t[L+1:,:] \in \mathbb{R}^{n_w' \times d}$.

The augmented weight matrix is partitioned by lag:

$$W_t = \begin{bmatrix} W_t^{(0)} \\ W_t^{(1)} \\ \vdots \\ W_t^{(L)} \end{bmatrix} \in \mathbb{R}^{(L+1)dp\times d}, \tag{9}$$

where $W_t^{(0)} \in \mathbb{R}^{dp\times d}$ encodes contemporaneous effects and $W_t^{(\ell)} \in \mathbb{R}^{dp\times d}$ ($\ell = 1,\ldots,L$) encodes lag-ℓ effects. The SEM becomes

$$Y_{t,j} = \sum_{i\neq j} g_{ij}^{(t,0)}(X_{t,i}) + \sum_{\ell=1}^{L}\sum_{i=1}^{d} g_{ij}^{(t,\ell)}(X_{t-\ell,i}) + \varepsilon_{t,j}, \tag{10}$$

where the contemporaneous sum excludes $i = j$ (no instantaneous self-loops), while the lagged sums include $i = j$ (autoregressive effects $X_{s-\ell,j} \to X_{s,j}$ are permitted, as they cannot form instantaneous cycles). An edge $i \xrightarrow{\ell} j$ at lag ℓ exists if and only if $\beta^{(\ell)}_{t,ij} \neq \mathbf{0}_p$.

The data fidelity retains the same quadratic form:

$$\ell(W_t; Y_t) = \frac{1}{2n'_w} \|Y_t - \Phi^{\text{aug}}_t W_t\|^2_F, \tag{11}$$

which remains convex in W_t since Φ^{aug}_t is a fixed data-dependent matrix.

C.2 Algorithmic changes

The extension from $L = 0$ to $L \geq 1$ requires three modifications to Algorithm 1. All other components—the Chambolle–Pock iteration, the operator $\tilde{K} = [I;\, K]$, the step-size condition, and the convergence guarantees—remain unchanged.

Modification 1: Dimension change. The vectorized weight dimension changes from $m = dp \cdot d$ to $m = (L+1)dp \cdot d$. The operator $K = D \otimes I_m$ and augmented operator $\tilde{K} = [I;\, K]$ use the new m, but $\|\tilde{K}\|^2 = 1 + \|K\|^2 < 5$ remains independent of L, so the step sizes $\tau = \sigma = 1/\sqrt{5}$ are unchanged.

Modification 2: Diagonal mask. The mask M is applied block-wise across lag levels:

$$M^{(\ell)}_{ij} = \begin{cases} 0 & \text{if } \ell = 0 \text{ and } i = j \quad \text{(no contemporaneous self-loops)}, \\ 1 & \text{otherwise} \quad \text{(lagged self-effects permitted)}. \end{cases} \tag{12}$$

This ensures that $X_{s,j}$ cannot appear as its own contemporaneous parent, while autoregressive effects $X_{s-\ell,j} \to X_{s,j}$ ($\ell \geq 1$) are allowed.

Modification 3: MFAS scope. In Stage 2, the regime-consistent MFAS projection is applied only to the contemporaneous block $W^{(0)}_t$. The lagged blocks $W^{(\ell)}_t$ for $\ell \geq 1$ are not subject to acyclicity projection, since the arrow of time prevents instantaneous cycles: $X_{s-\ell}$ cannot be caused by X_s. Formally, for each regime r:

- Compute $\bar{W}^{(0)}_r = \frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} W^{(0)}_t$ and apply MFAS to obtain the DAG $\hat{\mathcal{G}}_r$.
- Retain $W^{(\ell)}_t$ for $\ell \geq 1$ without modification.

Computational cost. The per-iteration cost increases from $O(T(dp)^2 d)$ to $O(T((L+1)dp)^2 d) = O(T(L+1)^2 d^3 p^2)$. For typical settings ($L \leq 2$, $p = 1$), this is at most $9\times$ the $L = 0$ cost. The precomputation cost for $(I + \tau H_t)^{-1}$ increases from $O(T(dp)^3)$ to $O(T((L+1)dp)^3)$.

C.3 Sparsity and lag selection

The maximum lag L is user-specified. A larger L increases model flexibility and estimation cost. The sparsity penalty encourages inactive lag coefficients to vanish, while correct lag selection requires additional signal and design conditions. The sparsity penalty operates uniformly across all lags:

$$\Omega(W_t) = \sum_{\ell=0}^{L} \Omega_\ell(W^{(\ell)}_t), \qquad \Omega_\ell(W^{(\ell)}_t) = \begin{cases} \|W^{(\ell)}_t\|_1 & (p = 1), \\ \sum_{i,j} \|\beta^{(\ell)}_{t,ij}\|_2 & (p > 1). \end{cases} \tag{13}$$

C.4 Convergence

The convergence of Theorem 3.1 extends directly to $L \geq 1$:

Corollary C.1. *For any fixed maximum lag $L \geq 0$, Algorithm 1 with the augmented design matrix satisfies the same $O(1/k)$ ergodic saddle-point gap bound as in Theorem 3.1, under the same optimization assumptions.*

Proof. The proof of Theorem 3.1 relies on three properties: (i) g is convex (quadratic in W_t), (ii) $\tilde{f}$ is convex (norm + composition with linear operator), and (iii) $\|\tilde{K}\|^2 < 5$. All three hold for any $L \geq 0$: increasing L changes the dimension m but not the problem structure. The operator norm bound $\|\tilde{K}\|^2 = 1 + \|K\|^2 < 5$ depends only on the eigenvalues of $D^\top D$, which are determined by T alone. □

D Additional Experimental Setup

D.1 Dataset and metrics

Dataset. Due to the absence of reliable ground-truth causal graphs in real-world non-stationary settings, we evaluate on synthetic data under both linear and nonlinear SEMs. The causal graph is sampled from an Erdős–Rényi model with $d \in \{5, \ldots, 200\}$ nodes and an expected degree of 4 (Lopez et al., 2022). For each generated edge, the edge weight is uniformly sampled from a specified range (e.g., $[-2, -0.5] \cup [0.5, 2]$) with equal probability (0.5). Each regime consists of 250 samples; at regime transitions, 20% of the edges are randomly modified (10% added, 10% removed). Time-lagged dependencies follow a VAR structure with lag $L \in \{1, 2, 3\}$, using bounded activations (e.g., $\tanh$) in nonlinear settings to prevent divergence.

The data are generated by a SEM whose causal structure may change over time as detailed in Eq. 7 where $\mathrm{pa}^{(0)}(i, j)$ denotes the contemporaneous parents of variable j at observation i, $\mathrm{pa}^{(\ell)}(i, j)$ for $\ell \geq 1$ denotes the lag-ℓ parents, and $\varepsilon_j^{(i)}$ is independent additive Gaussian noise ($\sigma = 1.0$). Setting $L = 0$ recovers the contemporaneous model.

Metrics. For causal structure recovery, we report Precision, Recall, F1, NHD, and NHD-ratio. With TP, FP, FN, and TN denoting edge-level confusion counts, these metrics are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1} = \frac{2\,\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We define $\text{NHD} = (\text{FP} + \text{FN})/d^2$, i.e., the average adjacency-entry error rate. To account for graph sparsity, we further report $\text{NHD-ratio} = \text{NHD}/\text{NHD}_{\text{base}}$, where NHD_{base} is the worse NHD obtained by the empty-graph and complete-graph predictors. Thus, $\text{NHD-ratio} < 1$ indicates improvement over this trivial baseline. Higher RegAcc, Precision, Recall, and F1 are better, while lower NHD and NHD-ratio are better.

The Area Under the Precision–Recall Curve (AUPRC) evaluates ranking quality across all possible edge-selection thresholds, rather than relying on a single fixed threshold (Davis & Goadrich, 2006). It is computed by varying the threshold over the predicted edge scores, calculating Precision and Recall at each operating point, and integrating the resulting curve: $\text{AP} = \sum_n (R_n - R_{n-1})\, P_n$.

D.2 Baselines and hyperparameters

Baselines. We compare against two categories of baselines to evaluate regime segmentation and causal structure recovery separately.

For regime segmentation, we include dedicated regime detection methods that do not perform causal discovery: K-Means (Arthur & Vassilvitskii, 2007) and Spectral clustering (Ng et al., 2001) applied directly to the raw time series, as well as T-Rep (Fraikin et al., 2024) and TS2Vec (Yue et al., 2022), which first learn temporal embeddings of the full dataset. We then apply K-Means or Spectral clustering to the learned representations (denoted T-Rep$_K$, T-Rep$_S$, TS2Vec$_K$, and TS2Vec$_S$, respectively).

For causal structure recovery, we include both non-stationary and stationary causal discovery methods. Non-stationary methods—RPCMCI (Saggioro et al., 2020), FANTOM (Rahmani & Frossard, 2025), and SpaceTime (Mameche et al., 2025)—are evaluated end-to-end. Since the number of existing non-stationary causal discovery methods remains limited, we additionally include stationary methods—CD-NOD (Huang et al., 2020), DYNOTEARS (Pamfil et al., 2020), NTS-NOTEARS (Sun et al., 2023), DiffAN (Sanchez et al., 2023), and DyCAST (Cheng et al., 2025)—by providing the ground-truth regime segmentation and running each method independently within each regime. This gives

stationary baselines a substantial advantage by eliminating regime detection errors entirely, making the comparison conservative for DACORD.

Hyperparameters. We aim to maintain comparable hyperparameter configurations across different methods to ensure fair evaluation. We initially attempted to align model capacities (e.g., network width, batch size, and training epochs) as closely as possible across all models. We also conducted ablation studies over hyperparameters to evaluate their sensitivity and robustness.

To ensure fair comparisons, all methods are evaluated under the same global configurations. We use 20 independent runs with different random seeds (ranging from 42 onward), unless computational constraints such as out-of-memory (OOM) errors or excessive runtime prevent repeated trials. A fixed threshold of 0.1 is used across all methods to determine the presence of edges. All experiments are conducted on a single NVIDIA RTX 3090 GPU. To mitigate OOM issues as the dimensionality increases, we adaptively scale model parameters depending on the number of variables d. Despite this, certain models still require manual adjustments for stability. The detailed hyperparameters are as shown in Table 5.

DACORD We use an ℓ_1 sparsity penalty $\lambda_1 = 0.1$ and a total variation penalty $\lambda_2 = 0.01$. The nonlinear component is modeled using RBF features with 10 components and bandwidth $\gamma = 1.0$. The window size is set to $\max(2d, 125)$, and the convergence tolerance is 10^{-6}.

DiffAN. The model is trained for 3000 epochs with a learning rate of 0.001. The batch size is set to 1024 for $d \leq 50$, 64 for $d = 100$, and 32 for $d \geq 200$.

FANTOM. We run a maximum of 3 EM iterations with a minimum regime size of 50. The batch size is set to 32 for $d < 100$, 16 for $d = 100$, and 8 for $d = 200$. Learning rates are set to 0.005, 0.002, and 0.001, respectively. The encoder and decoder hidden dimensions are reduced to $[100, 100]$, $[80, 80]$, and $[64, 64]$ accordingly. Other hyperparameters include a reconstruction loss weight of 0.5, Augmented Lagrangian iterations of 100 (inner) and 2–3 (outer), and a maximum of 50 Pinet rounds to prevent infinite loops.

DYNOTEARS. We use regularization parameters $\lambda_w = \lambda_a = 0.1$. Optimization is performed using L-BFGS-B with tolerance 10^{-8} and a maximum of 500 iterations.

NTS-NOTEARS. We set a Multi Layer Perceptron with dimensions $[d, 10, 1]$. Regularization parameters are $\ell_1 = 0.005$ and $\ell_2 = 0.01$, with tolerance 10^{-8} and maximum 500 iterations.

DyCAST. The model is trained for 500 epochs with learning rate 0.001. The latent dimension is 64 and lag is set to 1. Regularization includes $\ell_1 = 0.01$, diagonal penalty 0.01, and DAG penalty 1.0.

CD-NOD. We use significance level $\alpha = 0.05$. Independence testing is performed using Kernel Conditional Independence to capture non-linearity and fisherz for fast learning.

RPCMCI. We use Partial Correlation (ParCorr) as the conditional independence test with $\alpha = 0.01$, PC skeleton significance 0.2, and regime switching threshold 0.05. We also tested a GPU-accelerated Gaussian Process Distance Correlation (GPDCtorch) as the conditional independence test, but excluded it from the final evaluation as it became prohibitively slow beyond $d = 50$ with similar performance to ParCorr. The maximum lag is set to 1 (or 3 in lag experiments), with instantaneous edges enabled ($\tau_{\min} = 0$). We run 20 EM-style iterations with 5 annealing rounds. The number of regimes is provided as oracle input.

SpaceTime. We use minimum duration min_dur $= 50$ and a Gaussian process-based ST-GP method. The default lag is 1, adjusted depending on the dataset. Optimization includes 10 interleaved iterations and a change-point convergence tolerance of 3.

However, in high-dimensional settings, many methods frequently encountered out-of-memory (OOM) and excessively long runtimes when using comparable configurations as shown in Table 2. As a result, we adaptively adjusted certain hyperparameters (e.g., batch size, hidden dimensions, and

Table 5: Summary of method-specific hyperparameters.

Method	Hyperparameters
DACORD	$\lambda_1 = 0.1$, $\lambda_2 = 0.01$, RBF basis=10, $\gamma = 1.0$, window=$\max(2d, 125)$
DiffAN	3000 epochs, lr=10^{-3}, adaptive batch size by d
FANTOM	EM iterations=3, minimum regime size=50, adaptive batch size and hidden dimensions
DYNOTEARS	$\lambda_w = \lambda_a = 0.1$, L-BFGS-B, max iter=500
NTS-NOTEARS	MLP $[d, 10, 1]$, $\ell_1 = 0.005$, $\ell_2 = 0.01$
DyCAST	500 epochs, lr=10^{-3}, latent dimension=64
CD-NOD	$\alpha = 0.05$, Kernel Conditional Independence test
RPCMCI	ParCorr, $\alpha = 0.01$, 20 EM iterations, 5 annealing rounds, oracle R
SpaceTime	min_dur=50, ST-GP, interleave iterations=10

learning rates) for each model to ensure stable training and computational feasibility. Despite these adjustments, we kept configurations as close as possible to the original or recommended settings to preserve fairness.

We systematically evaluated performance across varying problem scales by increasing the dimensionality d from 8 to 128 and the batch size from 8 to 32. During this process, training-related hyperparameters such as the number of epochs and batch sizes were carefully tuned to maintain stability and avoid OOM issues, while still ensuring sufficient model convergence. The proposed model is implemented in Python 3.11 (Python Software Foundation License v2). Our experimental environment is built upon PyTorch ($\geq$ 2.2.0, the BSD-style/BSD 3-Clause license), leveraging NumPy ($\geq$ 1.26.0, the BSD 3-Clause license) for numerical computations and Pandas ($\geq$ 1.5.0, the BSD 3-Clause license) for efficient data manipulation.

D.3 Determinism and reproducibility

Our algorithm contains no stochastic component: no mini-batch sampling, no dropout, no random initialization, and no adaptive line search. The objective in Eq. 5 is convex and the deterministic update sequence converges to a global minimizer. The initialization is fixed ($w^0 = 0$), the step sizes are determined a priori by $\|\tilde{K}\|$, and all updates use the full dataset. Consequently, the optimization procedure is algorithmically deterministic given fixed data, hyperparameters, numerical precision, and implementation settings. This stands in contrast to augmented Lagrangian methods for nonconvex DAG learning, where multiple local minima, adaptive line search, and progressive penalty scheduling can yield different solutions across runs, as well as neural-network-based approaches that additionally introduce stochasticity through mini-batch optimization and random initialization.

E Additional Experimental Results

We conduct extensive ablation studies and sensitivity analyses to better understand the behavior of our method under various conditions. Due to space constraints, additional experimental results and full tables are provided in the Supplementary Material (additional_experimental_results.pdf). We visualize the key differences between our proposed method and the baseline models through visualization examples on a synthetic dataset as in Figure 5.

Additional metrics. Table 7 reports SHD, NHD, and NHD ratio under R=$3, 5, 10$ to complement the precision/recall analysis in the main text. Raw SHD values are large across all methods due to the high number of possible edges; the normalized metric provides a more interpretable comparison. DACORD achieves the lowest NHD in all completed settings (0.117, 0.058, and 0.027 for d=$50, 100, 200$), demonstrating that per-edge error rate decreases as d grows. The NHD ratio further contextualizes these results against a sparsity-based trivial baseline: DACORD's ratios (0.028–0.135) indicate substantial improvement, whereas FANTOM's ratios (0.977–0.989) show negligible improvement over the baseline, consistent with its near-trivial all-edges prediction pattern observed in Table 1. At d=200, DACORD is the only method that produces valid results in $R = 5, 10$.

To verify that DACORD's advantage is not an artifact of the fixed edge-selection threshold of 0.1, we additionally report threshold-free AUPRC in Table 9, computed from the unthresholded edge scores

Table 6: Ablation results for DACORD variants. Results are averaged over all available regimes and variable counts for each ablation setting. RC denotes regime-consistent projection.

Family	Variant	Runs	Precision	Recall	F1	SHD	NHD	Time (s)	GPU Mem.
MFAS/RC, no lag	v1	24	0.549	0.372	0.431	18.688	0.174	0.42	8.39
	v1 + R^2	24	0.506	0.434	0.458	18.986	0.187	0.43	8.39
	v1 + RC	24	0.528	0.436	0.473	19.128	0.175	0.42	8.39
	v2 (R^2 + RC)	24	0.465	0.494	0.473	20.490	0.200	0.43	8.39
MFAS/RC, contemporaneous	v1	72	0.717	0.401	0.493	15.144	0.140	0.44	9.24
	v1 + R^2	72	0.717	0.401	0.493	15.144	0.140	0.44	9.24
	v1 + RC	72	0.676	0.501	0.561	15.082	0.136	0.44	9.24
	v2 (R^2 + RC)	72	0.543	0.559	0.539	17.079	0.170	0.45	9.24
MFAS/RC, lagged	v1	72	0.750	0.292	0.397	11.759	0.087	0.44	9.24
	v1 + R^2	72	0.750	0.292	0.397	11.759	0.087	0.44	9.24
	v1 + RC	72	0.750	0.293	0.398	11.753	0.086	0.44	9.24
	v2 (R^2 + RC)	72	0.751	0.290	0.395	11.762	0.087	0.45	9.24
Debiasing	without debiasing	7	0.687	0.289	0.395	177.367	0.111	2.15	16.86
	with debiasing	7	0.521	0.387	0.435	255.238	0.119	1.92	16.86
Basis	linear	6	0.385	0.281	0.314	401.278	0.094	2.42	0.00
	polynomial degree 2	6	0.308	0.331	0.307	510.250	0.111	6.44	0.00
	polynomial degree 3	6	0.279	0.348	0.299	563.278	0.120	6.92	0.00
	RBF, 5 bases	6	0.264	0.368	0.300	576.533	0.130	7.70	0.00
	RBF, 10 bases	6	0.414	0.321	0.352	359.400	0.094	10.40	0.00
Adaptive basis	adaptive RBF	72	0.384	0.326	0.347	574.546	0.129	4.82	130.18
Uniform RBF	uniform RBF10	72	0.415	0.274	0.318	510.044	0.134	4.99	138.26
Solver	Condat–Vũ, linear	48	0.566	0.297	0.378	212.048	0.111	1.55	12.80
Window size	window $\times 0.33$	3	0.587	0.306	0.399	397.822	–	5.18	0.00
	window $\times 0.50$	3	0.610	0.299	0.397	394.489	–	5.11	0.00
	window $\times 1.00$	3	0.623	0.339	0.435	391.000	–	4.86	0.00
	window $\times 1.50$	3	0.604	0.371	0.459	391.333	–	4.78	0.00
	window $\times 2.00$	3	0.604	0.371	0.459	391.333	–	4.78	0.00
RBF10 grid	$\lambda_1 = 0, \lambda_2 = 0$	4	0.084	0.319	0.130	1055.583	0.334	1.13	49.48
	$\lambda_1 = 0, \lambda_2 = 0.01$	4	0.086	0.360	0.137	1087.617	0.361	1.05	49.48
	$\lambda_1 = 0, \lambda_2 = 0.05$	4	0.079	0.389	0.129	1241.558	0.409	1.05	49.48
	$\lambda_1 = 0, \lambda_2 = 0.1$	4	0.076	0.410	0.126	1315.717	0.438	1.08	49.48
	$\lambda_1 = 0, \lambda_2 = 0.5$	4	0.066	0.402	0.112	1585.217	0.495	2.01	49.48
	$\lambda_1 = 0, \lambda_2 = 1$	4	0.065	0.418	0.111	1634.517	0.514	3.01	49.48
	$\lambda_1 = 0.01, \lambda_2 = 0$	4	0.146	0.174	0.155	329.092	0.145	1.08	49.48
	$\lambda_1 = 0.01, \lambda_2 = 0.01$	4	0.202	0.226	0.199	314.867	0.135	1.17	49.48
	$\lambda_1 = 0.01, \lambda_2 = 0.05$	4	0.236	0.394	0.280	348.017	0.145	1.20	49.48
	$\lambda_1 = 0.01, \lambda_2 = 0.1$	4	0.219	0.346	0.252	345.842	0.145	1.16	49.48
	$\lambda_1 = 0.01, \lambda_2 = 0.5$	4	0.144	0.250	0.173	358.050	0.166	1.22	49.48
	$\lambda_1 = 0.01, \lambda_2 = 1$	4	0.136	0.237	0.162	358.292	0.168	1.29	49.48
	$\lambda_1 = 0.05, \lambda_2 = 0$	4	0.316	0.287	0.270	284.492	0.108	1.19	49.48
	$\lambda_1 = 0.05, \lambda_2 = 0.01$	4	0.434	0.379	0.399	181.508	0.088	1.02	49.48
	$\lambda_1 = 0.05, \lambda_2 = 0.05$	4	0.491	0.400	0.437	170.050	0.079	0.95	49.48
	$\lambda_1 = 0.05, \lambda_2 = 0.1$	4	0.430	0.334	0.374	198.483	0.085	0.93	49.48
	$\lambda_1 = 0.05, \lambda_2 = 0.5$	4	0.284	0.193	0.220	225.325	0.101	1.06	49.48
	$\lambda_1 = 0.05, \lambda_2 = 1$	4	0.264	0.186	0.210	225.883	0.103	1.24	49.48
	$\lambda_1 = 0.1, \lambda_2 = 0$	4	0.512	0.315	0.386	165.550	0.077	0.97	49.48
	$\lambda_1 = 0.1, \lambda_2 = 0.01$	4	0.523	0.324	0.398	154.400	0.077	0.92	49.48
	$\lambda_1 = 0.1, \lambda_2 = 0.05$	4	0.524	0.283	0.362	161.008	0.076	0.89	49.48
	$\lambda_1 = 0.1, \lambda_2 = 0.1$	4	0.449	0.219	0.289	172.733	0.081	0.87	49.48
	$\lambda_1 = 0.1, \lambda_2 = 0.5$	4	0.296	0.109	0.154	181.100	0.091	1.02	49.48
	$\lambda_1 = 0.1, \lambda_2 = 1$	4	0.288	0.106	0.149	181.200	0.091	1.21	49.48
	$\lambda_1 = 0.5, \lambda_2 = 0$	4	0.246	0.025	0.045	170.708	0.082	1.03	49.48
	$\lambda_1 = 0.5, \lambda_2 = 0.01$	4	0.243	0.024	0.042	170.608	0.082	1.02	49.48
	$\lambda_1 = 0.5, \lambda_2 = 0.05$	4	0.094	0.005	0.010	171.492	0.083	1.04	49.48
	$\lambda_1 = 0.5, \lambda_2 = 0.1$	4	0.078	0.004	0.007	170.833	0.082	1.17	49.48
	$\lambda_1 = 0.5, \lambda_2 = 0.5$	4	0.003	0.000	0.000	170.808	0.082	1.43	49.48
	$\lambda_1 = 0.5, \lambda_2 = 1$	4	0.003	0.000	0.000	170.783	0.082	1.78	49.48
	$\lambda_1 = 1, \lambda_2 = 0$	4	0.000	0.000	0.000	167.050	0.080	3.30	49.48
	$\lambda_1 = 1, \lambda_2 = 0.01$	4	0.000	0.000	0.000	166.900	0.080	3.31	49.48
	$\lambda_1 = 1, \lambda_2 = 0.05$	4	0.000	0.000	0.000	166.950	0.080	3.30	49.48
	$\lambda_1 = 1, \lambda_2 = 0.1$	4	0.000	0.000	0.000	166.950	0.080	3.32	49.48
	$\lambda_1 = 1, \lambda_2 = 0.5$	4	0.000	0.000	0.000	166.900	0.080	3.91	49.48
	$\lambda_1 = 1, \lambda_2 = 1$	4	0.000	0.000	0.000	166.850	0.080	5.13	49.48

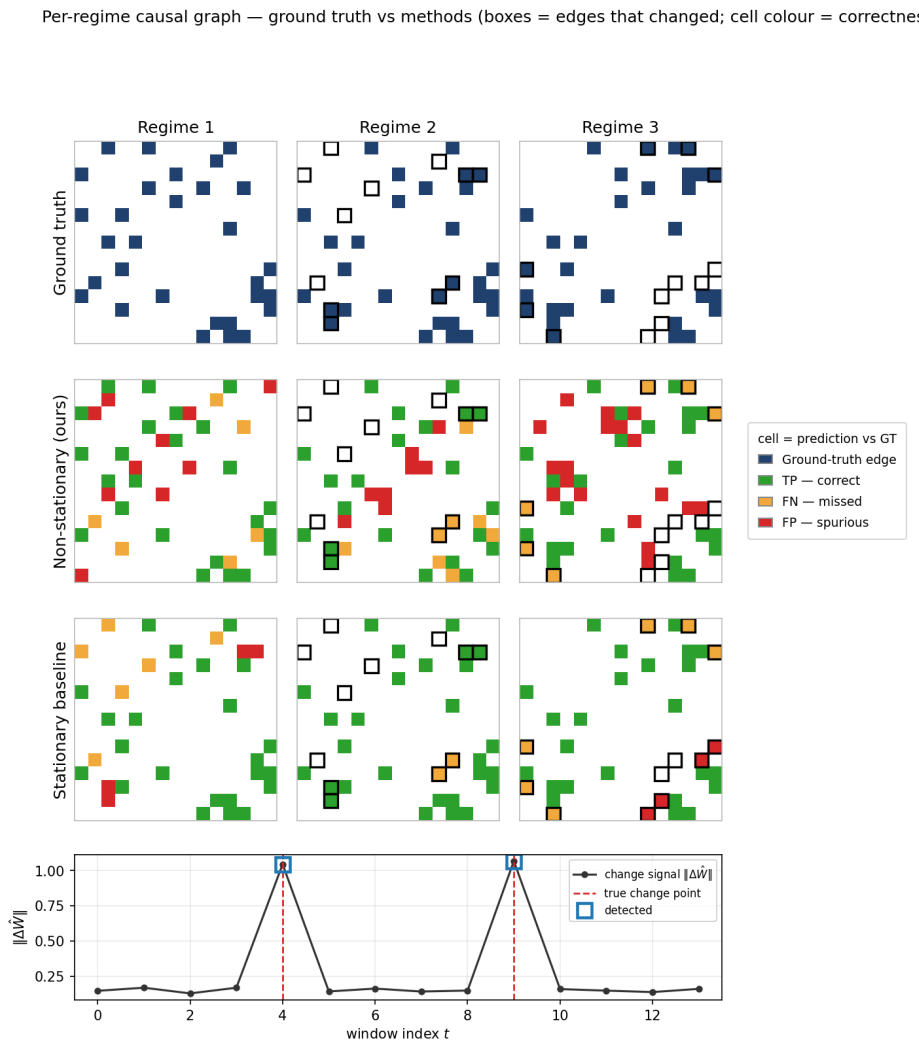

Figure 5: Visualization of Comparison between Ours and the baselines on synthetic dataset.

(Saito & Rehmsmeier, 2015). DACORD achieves the highest AUPRC in every completed (d, R) configuration, confirming that its advantage is also reflected in the ranking quality of the learned edge scores. RPCMCI is competitive at small scale, approaching DACORD at $d = 50$ and $R = 3$ (.211 vs. .216), but DACORD remains consistently higher as d increases and RPCMCI becomes unavailable in larger high-dimensional configurations. FANTOM's AUPRC remains low (.040–.136), indicating limited separation between true and false edges in its score ranking. Thus, the thresholded Precision/Recall/F1 results are not solely driven by the particular threshold choice.

Table 7: Additional structural-distance metrics under $R = 10$ across different numbers of variables. SHD, NHD, NHD ratio, and baseline NHD are reported for the same settings as Table 1. Missing entries indicate unavailable runs. "× too slow" did not complete within budget or unavailable.

d	Method	SHD	NHD	NHD Ratio	Baseline NHD
50	**DACORD**	**292.3**	**0.117**	**0.135**	0.883
	FANTOM	2158.5	0.863	0.977	0.883
	RPCMCI	583.1	0.233	0.269	0.883
	SpaceTime		× NaN (numerical failure)		
100	**DACORD**	**580.9**	**0.058**	**0.062**	0.939
	FANTOM	9293.0	0.929	0.989	0.939
	RPCMCI	1558.5	0.156	0.168	0.932
	SpaceTime		× NaN (numerical failure)		
200	**DACORD**	**1086.4**	**0.027**	**0.028**	0.970
	FANTOM		× NaN (numerical failure)		
	RPCMCI		× too slow		
	SpaceTime		× too slow		

Table 8: Semi-synthetic DREAM4 benchmark ($R = 5$ regimes, each regime a real DREAM4 topology). RegAcc is reported only for methods that perform regime detection; "oracle" denotes stationary methods run per regime with ground-truth boundaries. "too slow" = did not complete within budget.

d	Method	RegAcc	Prec	Rec	F1
10	DACORD	1.00	0.49	0.37	0.41
	CD-NOD	0.76	0.20	0.27	0.23
	FANTOM	—	0.16	1.00	0.27
	DYNOTEARS (oracle)	—	0.68	0.68	0.67
	NTS-NOTEARS (oracle)	—	0.16	0.30	0.21
	SpaceTime	—	× too slow		
100	DACORD	1.00	0.67	0.63	0.65
	CD-NOD	0.89	0.05	0.10	0.07
	FANTOM	—	0.02	1.00	0.04
	DYNOTEARS (oracle)	—	× too slow		
	RPCMCI / NTS-NOTEARS / SpaceTime	—	× too slow		

Semi-synthetic dataset: DREAM4. We construct a semi-synthetic benchmark whose regime graphs are real DREAM4 gene-regulatory topologies: five networks are used as five consecutive regimes. Table 8 reports the results. DACORD detects every regime boundary and is the only non-stationary method that completes with a usable graph at $d = 100$. DYNOTEARS attains the best graph F1 at $d = 10$, but only when given the ground-truth regime boundaries as an oracle, and it does not complete at $d = 100$. This is consistent with the positioning in Section 5: stationary solvers can be more accurate in small, segmented settings, yet they do not scale to the setting we target. We note that the real GRN topologies are cyclic, so no DAG method can recover them fully.

Case study on real dataset: S&P We apply DACORD to ten market-adjusted sector portfolios built from 58 S&P constituents continuously listed over 2000–2024. No reference causal graph is available, so we examine the estimated sector couplings descriptively.

Using the same knee rule as in our synthetic experiments, DACORD selects four change points: January 2009, May 2009, May 2011, and January 2020 (Figure 6A). The two 2009 detections bracket the March market trough and the expansion of the Federal Reserve's asset purchases. The January 2020 detection occurs shortly before the pre-COVID market peak, while the May 2011 detection falls within the euro-area sovereign debt crisis. No change point is selected around the 2002 market trough.

Separately, we examine six fixed historical periods, distinct from the five automatically detected regimes (Figure 6B–C). The row-normalised coupling scores identify InfoTech as the most prominent

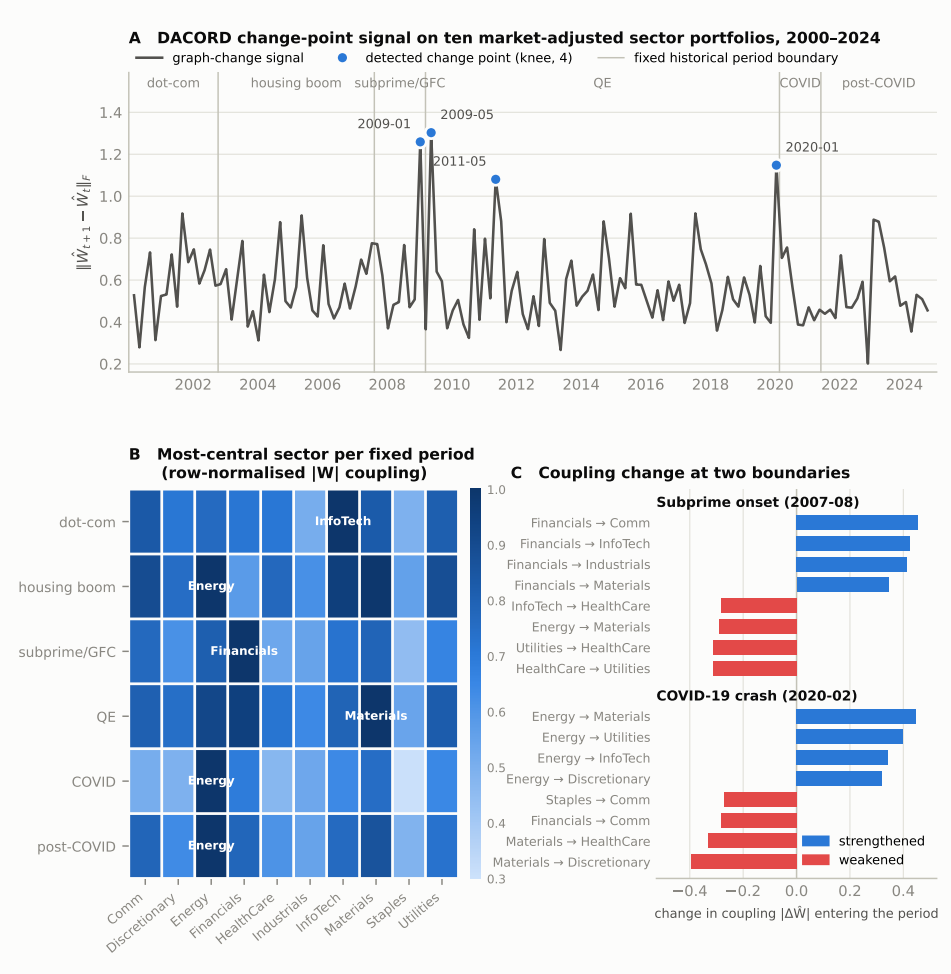

Figure 6: Visualization of Comparison On S&P dataset.

sector during the dot-com period, Financials during the subprime/GFC period, and Energy during COVID-19. Energy and Materials are prominent during the housing boom. Compared with the preceding fixed period, estimated outgoing couplings from Financials to Communication Services, Information Technology, and Industrials strengthen in the subprime period. Outgoing couplings from Energy to Materials, Utilities, Information Technology, and Consumer Discretionary strengthen in the COVID-19 period. These descriptive patterns are broadly consistent with previous sector-level connectedness findings (Laborda & Olmo, 2021).

Change-point detection. Using the same knee rule as in the synthetic experiments, DACORD selects four change points from the data alone. Three lie close to documented events—two in early 2009 bracketing the GFC bottom and QE1, and one at the pre-COVID market peak—while the fourth precedes the European debt crisis by about a quarter. The signal does not react at lower-volatility transitions such as the 2002 dot-com trough.

Regime-level structure. To avoid circularity, we then fix six regimes at documented dates and read, per regime, which sector carries the most coupling and which couplings change as each regime begins. Two readings are clean enough to state. InfoTech carries the cross-sector dynamics in the dot-com regime and recedes at the following boundary as Energy's couplings come forward. In the subprime regime, Financials is the most coupled sector, and at its onset the couplings running from Financials to Communication Services, InfoTech and Industrials strengthen together: financial stress becomes the channel through which shocks reach the rest of the market, rather than merely a stressed

sector. This is consistent with sector-level connectedness estimates, in which banking is the dominant transmission channel in 2007–2009 (Laborda & Olmo, 2021; Billio et al., 2012). Energy dominates the COVID regime, with couplings radiating outward at the crash—again the period Laborda & Olmo (2021) identify for Energy. The housing-boom regime is dominated by Energy and Materials, where one might have expected Financials; we report the discrepancy rather than fit a narrative to it.

Table 9: AUPRC across d and R. Values are mean (std) over nonlinear datasets. "×" = method failed.

		R=3			R=5			R=10		
Method	d	AUPRC	std	n	AUPRC	std	n	AUPRC	std	n
DACORD	50	**0.216**	0.006	4	**0.227**	0.012	3	**0.216**	0.003	3
	100	**0.183**	0.027	3	**0.215**	0.033	3	**0.177**	.006	3
	200	**0.170**	0.008	3	**0.205**	0.046	3	**0.213**	.023	3
FANTOM	50	0.109	0.039	5	0.108	0.042	3	.136	0.040	3
	100	0.055	0.020	3	0.070	0.020	3	.040	0.001	3
	200				× NaN					
RPCMCI	50	0.211	0.031	4	0.209	0.002	3	0.199	—	1
	100	0.136	0.023	3	0.167	0.017	3	0.140		
	200				× too slow					

Single regime setting (Stationary time series). To evaluate the effect of non-stationarity, we first consider a simplified setting with a single regime. This allows us to isolate the impact of regime changes and verify that the proposed method reduces to a standard causal discovery framework when temporal heterogeneity is absent.

Table A in the Supplementary Materials reports results under a single regime to evaluate DACORD when non-stationarity is absent. In this setting, stationary methods—particularly DYNOTEARS in the linear case—achieve substantially higher F1, as expected: with a single regime and sufficient samples, the augmented Lagrangian formulation can fully exploit the acyclicity constraint without scalability pressure. DACORD's TV penalty, designed to detect structural transitions, provides no benefit when the causal graph is constant, and the short effective window length reduces the sample size available for estimation. Nevertheless, DACORD maintains competitive precision in nonlinear settings (e.g., 0.571 at d=200) and remains orders of magnitude faster than all baselines. These results confirm that DACORD is specifically designed for the non-stationary, high-dimensional regime where existing methods become infeasible, rather than as a general-purpose replacement for stationary solvers.

Condat–Vũ optimization. The Stage 1 problem also admits a Condat–Vũ splitting that handles g via its gradient, avoiding the matrix inversion and yielding faster per-iteration cost (Condat, 2013). However, in our experiments the Chambolle–Pock formulation consistently achieves higher accuracy. (Table B in the Supplementary Materials) The gradient step only linearly approximates the quadratic subproblem that the proximal step solves exactly, and its convergence rate depends on $L_g = \max_t \|\Phi_t^\top \Phi_t / n_w\|$, which grows with d and the basis dimension—whereas the proximal rate depends only on $\|\tilde{K}\|^2 < 5$. Furthermore, the Chambolle–Pock splitting was preferred for computational reasons. Its primal proximal step solves the quadratic data-fidelity subproblem exactly, while the nonsmooth sparsity and TV terms are handled by simple dual projections. By contrast, the Condat–Vũ variant uses a gradient step on the smooth term, whose convergence depends on the Lipschitz constant $L_g = \max_t \|\Phi_t^\top \Phi_t / n_w\|$, which can grow with d and the basis dimension. Both splittings optimize the same composite objective; the difference lies in the numerical update structure rather than in a different statistical target.

Linear dataset. Table C in the Supplementary Materials reports results under linear and nonlinear SEMs. DYNOTEARS achieves the highest F1 across all completed settings (0.652–0.955), which is expected as it directly optimizes the linear acyclicity-constrained objective. However, its runtime exceeds one hour at d=50 and it fails to complete at $d \geq 100$. Among the remaining methods, DACORD consistently achieves the highest precision and F1 at $d \geq 50$ across all regime counts, while maintaining sub-minute runtimes even at d=200, R=10 (35 s vs. 8584 s for DYNOTEARS at d=50). These results show that specialized stationary methods retain an advantage in low-dimensional

settings—DYNOTEARS exploits the acyclicity constraint via augmented Lagrangian optimization in linear SEMs, while CD-NOD benefits from sufficient samples for reliable conditional independence testing at small d. In contrast, neural-network-based methods such as DiffAN and NTS-NOTEARS consistently underperform across both linear and nonlinear settings, with precision below 0.06 and 0.15 respectively at $d \geq 50$. FANTOM exhibits Recall $= 1.0$ with precision as low as .076, confirming the near-trivial prediction pattern observed in the main experiments.

Dense structures. We evaluate robustness under higher graph density by approximately doubling the expected number of edges per node relative to the default setting as shown in Table D of the Supplementary Materials. The same performance patterns observed in the main experiments persist. DYNOTEARS achieves higher precision than DACORD at d=50 but requires over one hour per run and does not scale beyond d=50. Across all settings where results are available, DACORD generally achieves the highest precision, while FANTOM continues to exhibit near-trivial prediction behavior with Recall ≈ 1.0 and precision below 0.16. Notably, DACORD's precision remains stable or even improves as d increases, confirming that the method's advantage is not sensitive to graph sparsity assumptions.

Without acyclicity. We also analyze the behavior when acyclicity is not enforced during optimization. Key results are presented in Table 10. This provides insight into how much performance depends on explicit DAG constraints.

Applying Stage 2 increases precision while decreasing recall, exhibiting a typical precision–recall trade-off. However, the higher recall without Stage 2 is largely attributable to predicting substantially more edges ($|\hat{E}|$): for example, at d=200 and R=3, removing Stage 2 increases the predicted edge count from 458 to 735 (+60%), inflating recall through indiscriminate edge inclusion rather than improved structure recovery as shown in Table 11. The precision gap widens with dimensionality—reaching $+0.14$ at d=200—indicating that the DAG projection becomes increasingly effective in high-dimensional settings where unconstrained regression produces more spurious cyclic edges.

Table 10: Effect of DAG projection (Stage 2 MFAS) on DACORD. Values are means over nonlinear datasets. $|\hat{E}|$ denotes the number of predicted edges.

Variant	d	R=3				R=5				R=10			
		P	**R**	**F1**	$\lvert\hat{E}\rvert$	**P**	**R**	**F1**	$\lvert\hat{E}\rvert$	**P**	**R**	**F1**	$\lvert\hat{E}\rvert$
With MFAS	50	0.490	0.237	0.318	143	0.504	0.244	0.327	144	0.509	0.245	0.328	138
	100	0.542	0.289	0.374	242	0.538	0.289	0.373	238	0.552	0.291	0.379	236
	200	0.661	0.364	0.469	458	0.687	0.382	0.490	464	0.694	0.374	0.486	454
Without MFAS	50	0.438	0.544	0.485	235	0.452	0.568	0.503	238	0.454	0.547	0.496	228
	100	0.496	0.492	0.494	393	0.495	0.493	0.493	395	0.509	0.499	0.504	388
	200	0.518	0.483	0.499	735	0.519	0.499	0.507	756	0.525	0.496	0.509	744

Table 11: What the Stage 2 projection removes. Stage 1 output is never acyclic: two thirds of its arcs belong to a two-cycle, i.e. the model predicts both $i \to j$ and $j \to i$. The projection deletes the order-violating arcs, and the majority of what it deletes is not in the ground truth at all. Means over $R \in \{3, 5, 10\}$ and three seeds, nonlinear datasets; counts are per regime.

d	Acyclic		Arcs			Removed arcs were…			Hedged true edges
	before	after	before	after	removed	absent in GT	re-oriented	lost	
50	0%	100%	197	103	95 (48%)	59%	24%	18%	70%
100	0%	100%	393	239	154 (39%)	63%	30%	7%	67%
200	0%	100%	747	460	287 (38%)	65%	28%	6%	65%

Regularization sensitivity. We conduct a sensitivity analysis over the two regularization parameters: the sparsity coefficient λ_1 and the temporal smoothness coefficient λ_2. This experiment evaluates how DACORD behaves under different levels of edge sparsity and temporal smoothing.

The results (Table E in the Supplementary Materials) show a clear regularization trade-off. When λ_1 is too small, the learned graphs become overly dense, leading to many false-positive edges and low precision. Conversely, overly large λ_1 values remove too many edges, reducing recall and F1. The TV coefficient λ_2 controls the temporal smoothness of the graph trajectory. Moderate TV regularization suppresses noisy window-to-window fluctuations and improves regime consistency, whereas excessively large λ_2 can over-smooth structural changes and merge adjacent regimes. DACORD performs best in a moderate regularization range, confirming that sparsity and TV regularization play complementary roles: λ_1 controls edge selection, while λ_2 controls temporal regime consistency.

Window size and regime length. Table F in the Supplementary Materials reports performance under varying samples-per-window (SPW). When SPW is too small (e.g., ≤ 50), each window contains insufficient samples for reliable estimation, causing recall to drop sharply. As SPW increases, F1 improves and stabilizes once each window contains enough samples to estimate the per-window regression reliably; beyond this point, further increases yield diminishing returns. SPW=200 and SPW=250 produce identical results in all tested configurations, as both yield the same number of effective windows T_{win}. DACORD is robust to window size selection across a wide range, with the default setting $n_w = \max(2d, 125)$.

At d=200, we also report results with the default window size of 250 samples per regime as shown in Table 12. Under this setting, RPCMCI completes at R=3 but still achieves substantially lower precision and F1 than DACORD. DACORD's performance decreases compared to the larger window configuration, but remains at a reasonable level, demonstrating robustness to reduced per-regime sample size.

Table 12: Precision (P), Recall (R), and F1 at d=200 across R. Values are mean over $d = 200$ dataset with 250 samples per regime. "× too slow" = method did not complete within budget (over 10 hours per run).

Method	R=3			R=5			R=10		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
DACORD	**.661**	**.364**	**.469**	**.687**	**.382**	**.490**	**.694**	**.374**	**.486**
FANTOM	× NaN (numerical failure)								
RPCMCI	.112	.337	.153	× too slow					
SpaceTime	× too slow (Gaussian process)								

Dataset seed. To assess robustness to the data-generating process, we repeat all experiments with three different dataset seeds (0, 1, 2), each producing a distinct random DAG and observation sequence (Table 13). Since DACORD is deterministic (Appendix D.3), the reported standard deviations reflect variability across datasets, not across algorithm runs. DACORD exhibits low variance (std ≤ 0.08 in F1) and consistently achieves the highest precision and F1 across all (d, R) configurations, confirming that the performance patterns observed in the main experiments are not artifacts of a particular dataset realization.

Table 13: Precision (P), Recall (R), and F1 across d and R. Values are mean (std) over nonlinear dataset across different dataset-generation seeds. "×" = method failed or did not complete within budget.

		R=3			R=5			R=10		
Method	d	P	R	F1	P	R	F1	P	R	F1
DACORD	50	0.490 (0.06)	0.237 (0.06)	0.318 (0.07)	0.504 (0.07)	0.244 (0.07)	0.327 (0.08)	0.509 (0.05)	0.245 (0.06)	0.328 (0.07)
	100	0.542 (0.05)	0.289 (0.08)	0.374 (0.08)	0.538 (0.06)	0.289 (0.07)	0.373 (0.07)	0.552 (0.04)	0.291 (0.07)	.379 (0.07)
	200	0.597 (0.03)	0.307 (0.06)	0.403 (0.06)	0.605 (0.03)	0.314 (0.06)	0.411 (0.06)	0.609 (0.03)	0.307 (0.06)	0.406 (0.06)
FANTOM	50	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)	0.119 (0.05)	1.000 (0.00)	0.210 (0.07)
	100	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)	0.061 (0.02)	1.000 (0.00)	0.115 (0.04)
	200	× NaN (numerical failure)								
RPCMCI	50	0.203 (0.01)	0.496 (0.12)	0.256 (0.05)	0.198 (0.02)	0.429 (0.11)	0.230 (0.04)	0.135 (0.11)	0.319 (0.25)	0.171 (0.13)
	100	0.147 (0.03)	0.315 (0.08)	0.155 (0.04)	0.141 (0.02)	0.450 (0.04)	0.196 (0.03)	0.092 (0.08)	0.262 (0.23)	0.126 (0.11)
	200	0.112 (0.03)	0.337 (0.08)	0.153 (0.04)	× too slow					
SpaceTime	50–200	× too slow (Gaussian process)								

Time lag. When time-lagged dependencies are present ($L \in \{1, 2, 3\}$), the same performance patterns persist: DACORD consistently achieves the highest precision across all baselines at $d \geq 50$,

Table 14: Temporal-penalty comparisons. Results are averaged over $d \in \{50, 100\}$, $R = 5$, and three seeds, using each variant's selected λ_2. Margin denotes the smallest boundary jump divided by the largest within-regime jump. All four variants attain perfect segmentation in these settings.

Temporal penalty	Margin	Plateau width	Graph F1
Elementwise ℓ_1	2.90	0.58	0.40
Time-block group	3.50	0.63	0.36
Edge-block group	4.20	0.66	0.41
Squared ℓ_2	1.98	0.45	0.34

showing results consistent with the contemporaneous experiments. The reported metrics are averaged over both contemporaneous and lagged edges.

Basis sensitivity. We investigate the effect of fixing the number of RBF basis functions. Empirically, we observe that fixing the number of basis functions limits performance on simpler patterns (underfitting), while still performing reasonably well on more complex patterns (Table H in the Supplementary Materials). We compare five basis functions: `linear`, `poly2` ($\phi(x) = [x, x^2]^\top$, p=2), `poly3` ($\phi(x) = [x, x^2, x^3]^\top$, p=3), `rbf5` and `rbf10` (Gaussian RBF with 5 and 10 equally spaced centers, respectively).

RBF bases yield the best and most consistent empirical performance across both linear and nonlinear settings, with smaller variation between them. This is consistent with prior work showing that basis expansions provide a tractable way to model nonlinear additive mechanisms in nonparametric DAGs, and that RBFs are flexible nonlinear approximators with localized behavior (Zheng et al., 2020). Polynomial bases of degree 2 and 3 improve over the linear basis, but their global monomial features become less stable as d grows, likely due to increased feature correlation and estimation variance in the expanded parameter space. In contrast, RBF bases provide localized approximations, which appear to scale more gracefully under the additive model as the number of candidate edge functions grows with d (Bühlmann et al., 2014).

ε_{cp} sweep and temporal consistency. The knee-selected threshold falls inside the plateau. Within the plateau the detected boundary set is identical for every admissible ϵ_{cp}, so the estimated regimes, the regime averages, and the per-regime DAGs are identical as well. Outside the plateau the two directions behave differently. Below it, the selector over-segments ($\hat{R} = 11.2$ against a true mean of 5.7). The spurious boundaries split a stable regime into near-duplicate graphs rather than corrupting it. The Hamming distance between estimated regimes that lie inside the same true regime is 8.0, whereas it is 40.0 between genuinely distinct regimes, a five-fold separation. Above the plateau, the selector merges distinct regimes and assigns a single averaged DAG across incompatible structures. Over-segmentation is therefore the benign failure mode and merging is the harmful one.

Regime accuracy. To broaden the comparison, we additionally evaluate dedicated regime detection methods alongside non-stationary causal discovery baselines in Table 16. DACORD achieves the highest regime accuracy across all d settings, outperforming not only causal discovery baselines but also dedicated regime detection methods including T-Rep and TS2Vec variants. Notably, regime accuracy peaks at d=50 (0.95) and decreases at d=100 (0.68). This reflects a trade-off inherent to our formulation: as d grows, each regime transition affects more edge parameters, producing a stronger TV signal that aids detection; however, the number of parameters to estimate also grows as $O(d^2)$, increasing estimation noise per entry. Despite this trade-off, DACORD maintains the highest accuracy across all tested dimensionalities.

Regime accuracy at the knee, default vs. corrected window. With 250 samples per regime, the default window rule $n_w = \max(2d, 125)$ leaves a single window per regime once $d \geq 100$. No within-regime difference $\hat{W}_{t+1} - \hat{W}_t$ then exists, and boundary detection degenerates. Fixing $n_w = 125$ restores two windows per regime and yields perfect regime accuracy at every d except $d = 10$ (Table 15). To isolate the cause, we hold the SEM fixed and vary only the window rule, and vice versa. The window rule accounts for $+0.41$ and $+0.46$ in regime accuracy at $d = 100$ and $d = 200$, whereas changing the SEM accounts for -0.11 and $+0.03$. The window effect is

Table 15: Regime accuracy at the knee-selected threshold under the default and corrected window rules. Parentheses give the number of windows per regime (wpr) under each rule.

Protocol	Window rule	$d = 50$	$d = 100$	$d = 200$
Linear basis	default $n_w = \max(2d, 125)$	1.000 (2)	0.467 (1)	0.600 (1)
Linear basis	corrected $n_w = 125$	1.000 (2)	1.000 (2)	1.000 (2)
RBF-10, linear SEM	default $\rightarrow$ corrected	1.00 $\rightarrow$ 1.00	0.49 $\rightarrow$ 1.00	0.57 $\rightarrow$ 1.00
RBF-10, nonlinear SEM	default $\rightarrow$ corrected	1.00 $\rightarrow$ 1.00	0.70 $\rightarrow$ 1.00	0.51 $\rightarrow$ 1.00

Table 16: Regime accuracy across d. Values are mean (std) over linear/nonlinear $\times$ R$\in \{3, 5, 10\}$. "—" = unavailable / out-of-memory; std omitted where per-seed measurements were not available.

	Causal Discovery				Regime Detection					
d	**Ours**	FANTOM	RPCMCI	SpaceTime	K-Means	Spectral	T-Rep$_K$	T-Rep$_S$	TS2Vec$_K$	TS2Vec$_S$
5	**0.65**	0.59 (0.20)	0.18	0.56 (0.12)	0.29	0.29	0.41	0.47	0.37	0.38
10	0.64	0.59 (0.22)	0.39	**0.66** (0.32)	0.28	0.28	0.41	0.46	0.34	0.37
20	**0.78**	0.51 (0.23)	0.50 (0.25)	—	0.29	0.30	0.45	0.48	0.38	0.40
50	**0.95**	0.35 (0.20)	0.59 (0.20)	—	0.39	0.45	0.56	0.63	0.52	0.57
100	**0.68**	0.36 (0.24)	0.57 (0.15)	—	0.45	0.53	0.61	0.64	0.54	0.57
200	**0.66**	—	0.60 (0.08)	—	0.49	0.58	0.58	0.62	0.54	0.56

therefore roughly an order of magnitude larger. A genuine dimension effect would not recover fully at $d = 200$, but this one does. The same saturation appears elsewhere. The window-size ablation in supplementary material about window size gives identical results at $\times 1.5$ and $\times 2.0$, and reports SPW$= 200$ and SPW$= 250$ as indistinguishable. In both cases the enlarged window yields the same reduced number of effective windows per regime.

Edge-fate analysis. The edge-fate analysis separates orientation errors from skeleton loss. Some removed true directions retain their reverse edges, whereas others remove the corresponding adjacency entirely. Forward false positives remain after projection, consistent with the scope of Lemma B.3.

We classify the fate of every Stage 1 edge as shown in Table 17. Stage 1 output is 0% acyclic, and 68–70% of its edges belong to bidirectional pairs. Moreover, 66–71% of the true edges it recovers arrive as such pairs, that is, without a directional claim. The projection resolves these pairs. Most of the recall it gives up is that resolution going the wrong way rather than structure loss. On the undirected skeleton, the F1 difference between projected and unprojected output is 0.076 at $d = 50$ and 0.02 at $d = 100$ and $d = 200$, and skeleton precision is unchanged (0.847 to 0.855 at $d = 200$). The retained false positives are order-aligned spouse edges, and their share decreases with d. The gap between the Stage 1 guarantee and final-DAG correctness is empirically an orientation gap rather than a support gap.

K-SVARM

Time and memory comparison. Table 19 reports wall-clock time and memory usage for all methods, including stationary baselines whose runtimes reflect the total cost of R independent runs across all ground-truth regimes. DACORD is the fastest method across all (d, R) configurations, completing in under 33 seconds even at d=200, R=10. Note that these totals exclude the cost of regime detection, which stationary methods do not perform; even under this favorable protocol, DACORD remains faster in all settings. Among non-stationary baselines, RPCMCI requires up to 4.7 hours at d=200 and times out at R=10; FANTOM fails numerically at d=200; and SpaceTime does not complete in any setting. Among stationary baselines given oracle segmentation, NTS-NOTEARS requires 77 hours at d=200, R=3 and DiffAN runs out of GPU memory at d=200. In terms of memory, DACORD uses at most 824 MB GPU and 1 GB CPU, whereas DiffAN requires over 15 GB GPU and DyCAST scales to 3.7 GB GPU at d=200, R=10.

Table 17: Composition of removed and retained edges under the Stage 2 projection, nonlinear datasets. Removed-edge percentages are ranges over seeds and (d, R) settings.

	$d = 50$	$d = 100$	$d = 200$
Of edges removed			
Absent from ground truth in that direction	58–60%	61–65%	64–66%
True direction of a two-cycle whose reverse survives	21–25%	28–32%	28–29%
True edge lost from the skeleton	17–19%	6–7%	6–7%
Of edges retained			
Not in ground truth (forward false positives)	45%	41%	37%

Table 18: DACORD vs. kernel SVARM at a common fixed edge threshold (K-SVARM at its documented default $\mathrm{rel}{=}0.05$). "oracle" fits one graph per ground-truth regime; "pooled" fits a single graph on the whole series. $|\hat{E}|$ is predicted arcs per regime and "% compl." is $|\hat{E}|$ as a fraction of the $d(d-1)$ possible arcs; $|E^\star|$ is the number of true arcs. K-SVARM's recall is obtained by predicting almost the complete graph, which contains every true edge by construction.

Benchmark	Method	d	Prec	Rec	F1	$\lvert\hat{E}\rvert$	% compl.	$\lvert E^\star\rvert$
DREAM4	**DACORD**	10	**0.476**	0.378	**0.417**	11.3	12.6%	14.2
	K-SVARM (oracle)	10	0.159	0.992	0.273	88.8	98.7%	14.2
	K-SVARM (pooled)	10	0.160	0.977	0.275	86.7	96.3%	14.2
	DACORD	100	**0.643**	0.648	**0.643**	205.7	2.1%	204.8
	K-SVARM (oracle)	100	0.021	0.995	0.042	9501.6	96.0%	204.8
	K-SVARM (pooled)	100	0.021	1.000	0.041	9897.0	100.0%	204.8
Synthetic nonlinear	**DACORD**	50	**0.563**	0.309	**0.399**	104.1	4.2%	185.0
	K-SVARM (oracle)	50	0.077	0.997	0.142	2406.0	98.2%	185.0
	K-SVARM (pooled)	50	0.076	1.000	0.140	2450.0	100.0%	185.0
	DACORD	100	**0.588**	0.353	**0.440**	238.1	2.4%	405.0
	K-SVARM (oracle)	100	0.045	0.963	0.086	8784.0	88.7%	405.0
	K-SVARM (pooled)	100	0.042	0.991	0.080	9651.0	97.5%	405.0
	DACORD	200	**0.631**	0.372	**0.467**	465.5	1.2%	769.0
	K-SVARM (oracle)	200	0.020	0.985	0.039	37804.8	95.0%	769.0
	K-SVARM (pooled)	200	0.019	1.000	0.038	39712.0	99.8%	769.0

F ALGORITHMS

The full algorithm is given in Algorithm 1.

Table 19: Wall-clock time and memory usage across d and R, mean (std) over linear/nonlinear $\times$ density. "—" = not measured. "$\times$ too slow" = method did not complete within budget.

		R=3			R=5			R=10		
Method	d	Time	GPU	CPU	Time	GPU	CPU	Time	GPU	CPU
DACORD	50	1.1s (.02)	42MB	798MB (0.7)	1.4s (.05)	61MB	804MB (1.1)	2.4s (.11)	110MB	820MB (1.0)
	100	2.1s (.04)	78MB	809MB (0.7)	3.0s (.06)	116MB	821MB (0.9)	5.6s (.15)	216MB	851MB (1.4)
	200	10.4s (.16)	266MB	866MB (0.6)	16.6s (.15)	428MB	903MB (1.2)	32.4s (.27)	824MB	1004MB (0.9)
DYNOTEARS	50	52.3m (13.99)	—	—	1.6h (0.25)	—	—	2.4h (0.00)	—	—
	100					$\times$ too slow				
	200					$\times$ too slow				
NTS-NOTEARS	50	38.6m (5.61)	23M (0.01)	1.3G (0.00)	1.0h (0.16)	23M (0.00)	1.3G (0.00)	2.2h (0.36)	23M (0.00)	1.3G (0.00)
	100	6.5h (0.35)	33M (0.00)	1.5G (0.00)	10.9h (1.04)	33M (0.00)	1.5G (0.00)	20.8h (0.64)	33M (0.00)	1.5G (0.00)
	200	77.1h (0.00)	70M (0.00)	2.0G (0.00)			$\times$ too slow			
RPCMCI	50	4.5m (0.46)	—	777M (3.20)	16.7m (0.90)	—	782M (0.17)	1.4h (0.24)	—	785M (0.93)
	100	22.3m (4.39)	—	788M (3.28)	56.8m (12.43)	—	797M (7.95)	3.6h (0.61)	—	813M (6.09)
	200					$\times$ too slow				
CD-NOD (fisherz)	50	9.5s (1.9)	—	806M (5)	11.7s (0.5)	—	816M (2)	17.0s (3.6)	—	832M (3)
	100	1.9m (0.59)	—	1.1G (0.09)	4.3m (0.62)	—	1.5G (0.14)	34.3m (28.4)	—	6.3G (4.10)
	200	24.9m (15.4)	—	4.6G (3.05)	55.4m (23.3)	—	10.0G (5.82)	2.0h (0.79)	—	19.9G (6.88)
CD-NOD (KCI)	50	21.86h (13.05)	—	869M (46)		$\times$ too slow			$\times$ too slow	
	100					$\times$ too slow				
	200					$\times$ too slow				
FANTOM	50	1.3m (0.09)	150M (0.00)	1.7G (0.00)	2.1m (0.19)	240M (0.00)	1.7G (0.00)	3.9m (0.27)	463M (0.00)	1.7G (0.00)
	100	1.3m (0.14)	411M (0.00)	1.7G (0.00)	2.0m (0.30)	672M (0.00)	1.7G (0.00)	4.0m (0.25)	1.3G (0.00)	1.7G (0.00)
	200					$\times$ NaN (numerical failure)				
DiffAN	50	11.2s (1.48)	15.1G (0.00)	1.6G (0.01)	18.3s (1.70)	15.1G (0.00)	1.6G (0.01)	39.5s (3.85)	15.1G (0.00)	1.6G (0.01)
	100	31.3s (4.08)	15.8G (0.00)	1.6G (0.01)	53.3s (5.31)	15.8G (0.00)	1.6G (0.01)	1.9m (0.22)	15.8G (0.00)	1.6G (0.01)
	200					$\times$ OOM (GPU)				
DyCAST	50	22.0s (2.52)	235M (0.00)	1.6G (0.00)	40.8s (2.03)	267M (0.00)	1.6G (0.00)	1.5m (0.12)	343M (0.00)	1.7G (0.00)
	100	23.0s (2.11)	794M (0.00)	1.7G (0.00)	42.8s (1.64)	873M (0.00)	1.7G (0.00)	1.6m (0.04)	1.0G (0.00)	1.9G (0.00)
	200	30.9s (1.33)	2.9G (0.00)	3.3G (0.00)	55.7s (2.46)	3.1G (0.00)	3.3G (0.00)	2.0m (0.03)	3.7G (0.00)	3.4G (0.00)
SpaceTime	50–200					$\times$ too slow (Gaussian process)				

Algorithm 1 DACORD

Require: Data $\{X_t, \Phi_t\}_{t=1}^{T}$, parameters $\lambda_1, \lambda_2, \epsilon_{\text{cp}}, \tau = \sigma = 1/\sqrt{5}, \theta = 1$

— Stage 1: Convex Optimization —

For each window t, precompute $H_t = \Phi_t^\top \Phi_t / n_w$, $B_t = \Phi_t^\top X_t / n_w$, and $P_t = (I_{dp} + \tau H_t)^{-1}$. For each target j, precompute $\Gamma_{t,j} = P_t[:, J_j](P_t[J_j, J_j])^{-1}$. ▷ $O(Td^3)$

Initialize $w^0 = \bar{w}^0 = 0,\ y_1^0 = 0,\ y_2^0 = 0$

for $k = 0, 1, \ldots$ until convergence **do**

  $y_1^{k+1} \leftarrow \text{clip}(y_1^k + \sigma \bar{w}^k,\ \pm\lambda_1)$ ▷ sparsity dual

  $y_2^{k+1} \leftarrow \text{clip}(y_2^k + \sigma K \bar{w}^k,\ \pm\lambda_2)$ ▷ TV dual

  Compute $v^k = w^k - \tau \widetilde{K}^* y^{k+1}$ and embed it as $\{V_t^k\}$ with zero self-coefficient blocks. For each window t, compute

$$U_t^k = P_t(V_t^k + \tau B_t).$$

  For each target j, update

$$W_t^{k+1}[:, j] = U_t^k[:, j] - \Gamma_{t,j} U_t^k[J_j, j].$$

  Form w^{k+1} from the free coordinates of $\{W_t^{k+1}\}_{t=1}^{T}$.

  $\bar{w}^{k+1} \leftarrow w^{k+1} + \theta(w^{k+1} - w^k)$ ▷ extrapolation

end for

— Stage 2: Regime-Consistent DAG Projection —

Detect regime boundaries: $\mathcal{B} = \{t : \|\hat{W}_{t+1} - \hat{W}_t\|_F > \epsilon_{\text{cp}}\}$

Partition $[T]$ into regimes $\mathcal{T}_1, \ldots, \mathcal{T}_{\hat{R}}$ using $\mathcal{B}$

for each regime $r = 1, \ldots, \hat{R}$ **do**

  $\bar{W}_r \leftarrow \frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} \hat{W}_t$ ▷ regime averaging

  Compute edge scores $s(i \to j) = |\bar{W}_{r,ij}|$

  Extract a greedy ordering by iteratively selecting the vertex with the largest net-flow score on the remaining induced subgraph

  Remove back-edges to obtain DAG $\hat{\mathcal{G}}_r$ ▷ MFAS

  Assign $\hat{\mathcal{G}}_r$ to all $t \in \mathcal{T}_r$ ▷ regime-consistent

end for

return DAGs $\{\hat{\mathcal{G}}_r\}_{r=1}^{\hat{R}}$, regime boundaries $\mathcal{B}$

Self-loop exclusion. We enforce $\beta_{t,jj} = \mathbf{0}_p$ by optimizing directly in reduced off-diagonal coordinates. The shared-inverse correction enforces the zero self-coefficient blocks. For $p > 1$, this embedding is broadcast over the p basis coefficients in each block. Thus, diagonal blocks are never updated as optimization variables, and the proximal step remains exact.

Per-iteration complexity. Each iteration is dominated by the primal step: one matrix multiplication per window with the precomputed $(I + \tau H_t)^{-1}$, costing $O(d^3)$ per window and $O(Td^3)$ in total. While this asymptotic cost matches existing methods, three structural advantages yield a 20–40$\times$ practical speedup: (1) a single matrix multiplication versus $\sim$26 for the matrix exponential and its gradient; (2) no line search, since step sizes are fixed a priori by $\|\tilde{K}\|$; (3) a single optimization loop without nested penalty scheduling.

G LIMITATIONS

A limitation of our evaluation is the absence of real-world non-stationary datasets with reliable ground-truth causal graphs. We therefore complement controlled synthetic experiments with a semi-synthetic DREAM4 benchmark, which provides realistic graph topologies with known regime structure, and an S&P sector case study, which evaluates whether the recovered regime changes and dependencies admit plausible domain interpretation. The latter does not provide ground-truth graph accuracy and should therefore be viewed as qualitative validation rather than causal verification. Additionally, the ℓ_1 TV penalty is designed for abrupt structural breaks; under gradual concept drift, it may impose discrete regime boundaries rather than tracking smooth parameter evolution. The predominant formulation in the non-stationary causal discovery literature models structural change as discrete regime transitions—where the causal mechanism remains stable within each regime and shifts at identifiable change points (Rahmani & Frossard, 2025). Our method directly targets this setting, and extending the framework to smooth transitions (e.g., via ℓ_2 or group-fused TV penalties) is a natural direction for future work.

H BROADER IMPACTS

The proposed method is applicable to domains where causal mechanisms change over time, including economics, healthcare, and process optimization (Glymour et al., 2019). Importantly, DACORD achieves the highest precision across all completed settings. In high-stakes domains such as medicine, false-positive causal edges can trigger costly or harmful interventions. Our precision profile thus reduces the risk of misguided decision-making driven by spurious causal claims (Soh & Chandrasekaran, 2017). As a general-purpose optimization algorithm for causal structure learning, our method does not directly enable domain-specific harmful applications.