RESP: RELAXED SPARSEST-PERMUTATION FOR CAUSAL DISCOVERY AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the growing availability of large datasets, causal discovery remains computationally prohibitive at scale. We introduce Relaxed Sparsest-Permutation (ReSP), a support-constrained formulation for linear structural equation models (SEMs) that retains the number of edges in the induced directed acyclic graph (DAG) as the selection criterion while relaxing exact all-entry Cholesky evaluation. ReSP requires the factorization equality only within a precision-support mask, and the ordering-dependent lower-triangular factor must be zero outside it, preventing fill outside the screened graph during candidate ordering evaluation. We establish two population-level guarantees for Gaussian linear SEMs under no-cancellation and sparsest Markov representation assumptions using the true precision-support mask. With a suitable candidate ordering set, every ReSP minimizer induces a DAG in the true Markov equivalence class; even without such candidates, all feasible solutions induce edge sets within the moralized skeleton. We demonstrate ReSP's practical potential through a simple instantiation using a single approximate minimum degree ordering and local refinement. Ablation studies show that masking improves structural accuracy and reduces downstream computational cost. This instantiation scales to 10^4 variables with $n < p$ and improves recovery as the sample size increases to 10^6 in weak-signal experiments. Full-feature TCGA analysis further yields greater directional agreement with validated biological regulatory knowledge than feature-partitioned approaches.

1 INTRODUCTION

Learning a directed acyclic graph (DAG) and its Markov equivalence class (MEC) from observational data is a central goal in causal discovery. In applications such as genomics, neuroscience, and socio-economic systems, this task increasingly involves tens of thousands of variables (Linnarsson & Teichmann, 2016; Trapnell et al., 2014). Recovering weak causal effects can also require large sample sizes, making scalability in both dimension and sample size essential. At these scales, repeated conditional independence (CI) tests and likelihood-based searches can become computationally prohibitive (Kalisch & Bühlmann, 2007; Chickering, 2002).

Existing methods have improved scalability in different ways. PC (Kalisch & Bühlmann, 2007), GES (Chickering, 2002), and GSP (Solus et al., 2021) are well established but require increasingly expensive statistical queries or searches. Continuous-optimization methods such as NOTEARS (Zheng et al., 2018) and GOLEM (Ng et al., 2020) avoid discrete search but repeat dense computations. Recent methods such as BOSS (Andrews et al., 2023) and FLOP (Wienöbst et al., 2026) accelerate ordering search but remain costly as p grows, with FLOP additionally requiring $n > p$. Precision-based sparse-Cholesky pipelines extend to larger problems, but RFD (Squires et al., 2020a) remains limited to moderate sizes and FROSTY's (Bang & Oh, 2023) comparative advantage in scalability comes at the cost of being sensitive to ordering-dependent fill.

Sparsest-permutation (SP) learning (Raskutti & Uhler, 2018) offers a complementary approach to scalable causal discovery, using graph sparsity rather than numerical fit as its population criterion. Under the sparsest Markov representation (SMR) assumption, which is weaker than faithfulness, this criterion identifies the true MEC. We introduce Relaxed Sparsest-Permutation (ReSP) to retain this criterion while improving scalability by replacing exact ordering-specific factorization with support-constrained evaluation.

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.

Table 1: Representative methods, structural criteria, and core computations in causal discovery.

Approach	Criterion and constraints	Core computation	Examples
Constraint-based	CI relations	Repeated CI tests	PC
Score-based search	Penalized DAG/MEC fit	Scoring and search	GES, BOSS, FLOP
Continuous opt.	Smooth objective; acyclicity	Optimization updates	NOTEARS, GOLEM
Sparse Cholesky	Ordering heuristics; exact factorization	Order + factorize	FROSTY, RFD
SP	Factor sparsity; $\Omega_\pi = L_\pi L_\pi^\top$	Exact evaluation	GSP, GRaSP
ReSP	**Factor sparsity; masked equality/support**	Masked evaluation	SCOPE

Our key insight is that, for support-level structure recovery, enforcing the exact Cholesky identity entrywise is stronger than necessary. ReSP instead restricts candidate triangular factors to precision support, reducing repeated candidate-evaluation cost without changing the ordering search space. Conceptually, ReSP relaxes exact SP by

$$\text{Sparsest Permutation (SP):} \quad \min_{\pi} \|L_\pi\|_0 \ \text{ s.t. } \ \Omega_\pi = L_\pi L_\pi^\top,$$
$$\text{Relaxed Sparsest-Permutation (ReSP):} \quad \min_{\pi} \|L_\pi\|_0 \ \text{ s.t. } \ \Omega_\pi \approx (L_\pi L_\pi^\top),$$

where equality is required only at nonzero entries of Ω_π, and L_π is constrained to be zero elsewhere. The exact formulation of ReSP is developed in Section 3.

ReSP specifies support constraints and a sparsity criterion for evaluating candidate orderings, allowing different algorithms to generate and search over orderings. With a suitable candidate set containing a reverse topological ordering, every minimizer induces a DAG in the true MEC under the SMR assumption. For arbitrary candidate sets, every feasible solution induces an edge set contained in the moralized skeleton. This support confinement also allows finite-sample algorithms to refine candidate structures within the screened graph. Table 1 places ReSP among representative causal discovery approaches by outlining their structural criteria and core computations.

We instantiate ReSP with SCOPE (Sparse-ChOlesky PipelinE), a one-pass implementation using a single approximate minimum-degree (AMD) ordering (Amestoy et al., 1996), masked factor construction, and local refinement. The high-dimensional ablation studies isolate the contribution of masking, while simulations across strong-signal, weak-signal, and high-dimensional regimes show robust performance. A full-feature TCGA analysis illustrates the practical value of this approach for joint structure learning for a large set of variables.

Our contributions are summarized as follows:

- **A support-constrained SP formulation.** ReSP relaxes SP through masked equality and precision-support constraints, retaining its sparsity objective and opening a path to scalable causal discovery algorithms.
- **Reliable structural recovery.** Under the stated assumptions, ReSP minimizers recover DAGs in the true MEC for suitable candidate sets, while support confinement holds for every feasible solution regardless of the candidate set. Ablation studies demonstrate masking's finite-sample accuracy benefits.
- **Expansion of the feasible regime.** One-pass SCOPE illustrates ReSP's computational reach: high-dimensional experiments with $p = 10^4$ and $n < p$, weak-signal experiments with $n = 10^6$, and full-feature TCGA analysis with $p = 16{,}325$.

2 Preliminaries

2.1 Linear Structural Equation Models

The model structure is represented by a DAG $G = (V, E)$, where $V = \{1, \ldots, p\}$ is the set of variable labels and $E \subseteq V \times V$ is the set of directed edges. We write (j, k) for an edge from node j to node k. For a node $k \in V$, let $\mathrm{Pa}(k) := \{j \in V : (j, k) \in E\}$ and $\mathrm{Ch}(k) := \{j \in$

$V : (k, j) \in E\}$ denote its parents and children. The moralized graph G^m is the undirected graph obtained by connecting any two parents of a common child and then dropping all edge directions. We represent a skeleton by the symmetric set of ordered adjacent pairs, excluding the diagonal, and write $\Delta_p := \{(i, i) : i \in V\}$.

Let $\pi = (\pi_1, \ldots, \pi_p) \in \mathbb{S}_p$ be an ordering, where $\mathbb{S}_p$ is the set of all permutations and $\pi(i) := \pi_i \in V$ denotes the variable at position i. We extend π to sets of index pairs by applying it to both coordinates. The permutation matrix P_π reorders $A \in \mathbb{R}^{p\times p}$ as $A_\pi := P_\pi A P_\pi^\top$, with $(A_\pi)_{ij} = A_{\pi(i),\pi(j)}$. Thus, $\mathrm{supp}(A) = \pi(\mathrm{supp}(A_\pi))$ expresses the support in the original variable labels.

An ordering π is a (not necessarily unique) topological ordering if $i < j$ (parents before children) whenever $(\pi_i, \pi_j) \in E$ and reverse topological if $i > j$ (parents after children) whenever $(\pi_i, \pi_j) \in E$. A topological ordering is guaranteed to exist for any DAG.

A linear SEM associates $X \in \mathbb{R}^p$ with a DAG G via a weighted adjacency matrix $B = [B_{kj}]$:

$$X = BX + \epsilon, \tag{1}$$

where $B_{kj} \neq 0$ if and only if (j, k) is an edge in G. The noise vector $\epsilon = (\epsilon_1, \ldots, \epsilon_p)^\top$ has independent components with $\mathbb{E}[\epsilon_j] = 0$ and $\mathrm{Var}(\epsilon_j) = \sigma_j^2 > 0$. Write $\Sigma_\epsilon := \mathrm{Var}(\epsilon) = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. The population precision matrix is $\Omega := \mathrm{Var}(X)^{-1} = (I_p - B)^\top \Sigma_\epsilon^{-1} (I_p - B)$.

The MEC $\mathcal{M}(G)$ consists of all DAGs that entail the same CI relations as G (Spirtes et al., 2000). It is represented by a completed partially directed acyclic graph (CPDAG), where an edge is directed if and only if its orientation is invariant across all DAGs in $\mathcal{M}(G)$ (Andersson et al., 1997).

We consider i.i.d. samples from the linear SEM (1) and assume Gaussian errors for the theoretical analysis. We work with data-generating DAGs with small in-degrees, so parent sets remain sparse while moralized degrees can be large, allowing for hub nodes. These graph characteristics are consistent with networks observed in genomics, neuroscience, and socio-economic systems, which often exhibit sparse local dependencies with prominent hubs and substantial degree heterogeneity (Barabási & Oltvai, 2004; Jeong et al., 2000; Bullmore & Sporns, 2009; Newman, 2003).

2.2 Sparsest-Permutation Approaches

SP learning for Gaussian linear SEMs can be seen as a variable-ordering problem with the objective

$$\min_{\pi\in\mathbb{S}_p} \|L_\pi\|_0 \quad \text{s.t.} \quad \Omega_\pi = L_\pi L_\pi^\top, \tag{2}$$

where L_π is the lower-triangular Cholesky factor of $\Omega_\pi := P_\pi \Omega P_\pi^\top$. The off-diagonal support of L_π induces a directed edge set under π. Specifically, define

$$\widetilde{E}(\pi, L_\pi) := \{\, (\pi(k), \pi(j)) : \; k > j, \; (L_\pi)_{kj} \neq 0 \,\},$$

and let $G_\pi = (V, \widetilde{E}(\pi, L_\pi))$ be the induced DAG. Thus, a nonzero in row k, column j represents $\pi(k) \to \pi(j)$: parents follow their children in this reverse topological ordering π_0 of the SEM,

$$L_{\pi_0} = (I_p - B_{\pi_0})^\top \Sigma_{\epsilon,\pi_0}^{-1/2},$$

which is lower triangular and induces the true edge directions. By construction, the off-diagonal nonzeros of L_π are in one-to-one correspondence with $\widetilde{E}(\pi, L_\pi)$, so $\|L_\pi\|_0 = p + |\widetilde{E}(\pi, L_\pi)|$.

Identifiability for MEC recovery can be obtained under the SMR assumption, which is typically weaker than faithfulness; see Raskutti & Uhler (2018). Existing SP methods differ in how they search the permutation space and evaluate candidate orderings.

Exact SP characterization (CI-based). For Gaussian linear SEMs, the ordering-induced factor has the CI characterization (Raskutti & Uhler, 2018)

$$(L_\pi)_{kj} = 0 \quad \Longleftrightarrow \quad X_{\pi(j)} \perp X_{\pi(k)} \mid X_{C_{jk}(\pi)}, \qquad k > j,$$

where $C_{jk}(\pi) := \{\pi(\ell) : \ell > j, \; \ell \neq k\}$. These conditioning variables follow j in the elimination ordering, or equivalently precede it in the corresponding topological ordering of the induced DAG G_π. The conditioning sets can grow linearly with p; SP compares the resulting factor supports over orderings, with exhaustive optimization requiring evaluation across the permutation space.

Greedy SP variants. Greedy variants replace global optimization over $\mathbb{S}_p$ with local search through neighboring orderings (Teyssier & Koller, 2005; Lam et al., 2022). Starting from an initial ordering, they accept moves that improve a sparsity or likelihood-based score and stop at a local optimum. Representative examples include GSP (Solus et al., 2021), GRaSP (Lam et al., 2022), and ARCS (Ye et al., 2021). Recent order-search methods such as BOSS (Andrews et al., 2023) and FLOP (Wienöbst et al., 2026) accelerate ordering search, though FLOP requires $n > p$.

Limitations. Both exhaustive and greedy SP methods require repeated ordering-dependent evaluations through CI tests or likelihood-based scores. Although local search avoids enumerating all $p!$ orderings, its set of candidate moves still grows with p, without simplifying each candidate evaluation. Repeated over many iterations, these costs can remain prohibitive at large p. Sparse Cholesky offers an apparent shortcut, but it does not directly optimize the SP objective, as we explain next.

2.3 Sparse Cholesky vs. SP objectives

Sparse Cholesky factorization computes a lower-triangular factor L satisfying $A = LL^\top$ for a sparse symmetric positive definite matrix A (Davis, 2006; Saad, 2003). Its primary goal is to choose an ordering π so that factoring the permuted matrix $A_\pi := P_\pi A P_\pi^\top$ introduces as little *fill-in* as possible, i.e., nonzeros created during elimination beyond the original sparsity pattern of A_π.

Fill-reducing approaches use what is referred to as a symbolic elimination graph (see Scott & Tůma, 2023, Chapter 3), which predicts possible factor entries from the sparsity pattern without accounting for numerical cancellations. Let $\mathcal{P}(\pi)$ denote the symmetrized symbolic Cholesky pattern of A_π and define its fill set by

$$\mathcal{F}(\pi) := \mathcal{P}(\pi) \setminus \operatorname{supp}(A_\pi).$$

The numerical factor satisfies $\operatorname{supp}(L_\pi + L_\pi^\top) \subseteq \mathcal{P}(\pi)$, potentially strictly when entries cancel. For Gaussian linear SEMs, we specialize to $A = \Omega$, whose numerical factor sparsity defines the SP objective (Raskutti & Uhler, 2018).

Objective comparison. Although both sparse Cholesky ordering and SP learning are phrased through a triangular factor, they optimize different criteria:

$$\text{Sparse Cholesky: } \min_{\pi \in \mathbb{S}_p} |\mathcal{F}(\pi)| \quad \text{vs.} \quad \text{Sparsest Permutation: } \min_{\pi \in \mathbb{S}_p} \|L_\pi\|_0.$$

This mismatch arises because fill reduction is defined on an *undirected* elimination graph, whereas $\|L_\pi\|_0$ reflects *directed* sparsity from conditional independencies. Figure 1 illustrates how the skeletons obtained under fill-reducing Cholesky orderings and the SP criterion differ relative to $\operatorname{supp}(\Omega)$.

Sparse Cholesky factorization methods are designed to reduce the set difference S1 in Figure 1 when computing an ordering π and the corresponding factor L_π. For a given ordering π, the symmetrized triangular support $\pi(\operatorname{supp}(L_\pi + L_\pi^\top))$ may extend beyond $\operatorname{supp}(\Omega)$ commonly referred to as *fill-ins* in the sparse linear algebra literature.

Beyond reducing fill-ins, SP learning imposes an even stricter sparsity objective. Aside from the set difference S1, the set difference S2 highlights additional nonzeros that vanish under a reverse topological ordering π_0, associated with moralized edges arising from v-structures under suitable regularity conditions. Accordingly, SP learning accounts for sparsity both outside and within $\operatorname{supp}(\Omega)$.

An ordering with zero symbolic fill can still yield a denser numerical factor than one with positive symbolic fill, because numerical cancellations can remove entries predicted by the elimination graph (Appendix A). Thus, fill reduction alone is insufficient to minimize the SP objective. A misspecified ordering may introduce edges outside precision support (S1) and retain moralization edges within it (S2). ReSP excludes S1 by construction while retaining sparsity-based comparison within the mask to eliminate S2.

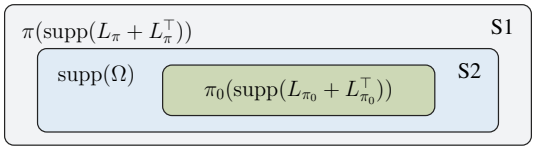


Figure 1: Nested support patterns: S1 fill beyond $\operatorname{supp}(\Omega)$, and S2 additional sparsity under π_0.

3 RELAXED SPARSEST-PERMUTATION FORMULATION

This section formalizes the relaxation that makes SP-style causal discovery scalable. The key idea is to evaluate candidate orderings only within a precision-support screen, rather than through exact all-entry representations. This reduces per-candidate evaluation cost, while feasibility under the mask confines the induced edges of even a misspecified ordering to the moralized skeleton. Detailed proofs of all theoretical results are deferred to Appendix C.

$$\min_{\pi \in \mathbb{T}_p} \|L_\pi\|_0 \quad \text{s.t.} \quad (\Omega_\pi - L_\pi L_\pi^\top) \circ M_\pi = \mathbf{0}, \quad L_\pi \circ (\mathbf{1} - M_\pi) = \mathbf{0}. \tag{3}$$

Here, each L_π is lower triangular with positive diagonal. M denotes the precision-support mask,

$$M := \mathbb{I}\{\Omega \neq 0\} \in \{0,1\}^{p \times p}, \quad M_{ij} = \mathbb{I}\{\Omega_{ij} \neq 0\},$$

and $M_\pi := P_\pi M P_\pi^\top$ for each ordering π. The set $\mathbb{T}_p \subseteq \mathbb{S}_p$ represents the candidate orderings evaluated during optimization. It may be chosen from prior knowledge, local search proposals, or a heuristic ordering, and in scalable implementations it can be very small, even a singleton. Hence, ReSP is an optimization formulation over a user-specified candidate set, combining support-constrained evaluation with sparsity-based selection.

The mask M yields an ordering-agnostic screening graph, restricting feasible supports to $\mathrm{supp}(\Omega)$ under every ordering. It suppresses the set difference S1 in Figure 1 by forbidding any fill outside $\mathrm{supp}(\Omega)$, while the formulation compares orderings by the sparsity of L_π within the mask.

We evaluate (3) via masked zero-fill incomplete Cholesky factorization, often denoted as IC(0) in the literature Lin & Moré (1999); Meijerink & van der Vorst (1977); Saad (2003). For each π we compute the IC(0) factor $L_\pi = \mathrm{IC0}(\Omega_\pi; M_\pi)$, which satisfies

$$(\Omega_\pi - L_\pi L_\pi^\top) \circ M_\pi = \mathbf{0}, \quad L_\pi \circ (\mathbf{1} - M_\pi) = \mathbf{0}. \tag{4}$$

Our focus is the support of L_π, not numerical approximation of Ω_π by $L_\pi L_\pi^\top$.

In linear SEMs, the precision matrix entries combine direct and shared-child effects. To make this explicit, index variables so that B is strictly lower triangular. Then, for any $j < k$, the population precision matrix admits the decomposition

$$\Omega_{kj} = -\sigma_k^{-2} B_{kj} + \sum_{\ell > k} \sigma_\ell^{-2} B_{\ell j} B_{\ell k}.$$

The first term corresponds to the directed edge (j, k), while the summation aggregates contributions from shared children of j and k. Consequently, absent exact algebraic cancellations, any adjacency in the moralized graph G_0^m yields $\Omega_{ij} \neq 0$. This motivates the following no-cancellation condition, under which $\mathrm{supp}(\Omega)$ provides a valid ordering-agnostic screening mask.

Assumption 1 (No-cancellation Loh & Bühlmann (2014))**.** *Assume the linear SEM* (1) *with independent noise, and without loss of generality index variables so that B is strictly lower triangular. For all $j < k$,*

$$\Omega_{kj} = 0 \;\Rightarrow\; B_{kj} = 0 \;\; \textit{and} \;\; B_{\ell j} B_{\ell k} = 0, \; \forall\, \ell > k.$$

Lemma 3.1 (Mask validity under no-cancellation)**.** *Assume Assumption 1. Then for the true DAG G_0 and any $\pi \in \mathbb{S}_p$,*

$$\mathrm{skel}(G_0) \subseteq \pi\big(\mathrm{supp}(\Omega_\pi) \setminus \Delta_p\big) = \mathrm{skel}(G_0^m).$$

Lemma 3.1 shows that, uniformly over all orderings, the precision-support mask restricts the locations of nonzero entries without removing true parent-child adjacencies, and rules out the set difference S1 (fill beyond $\mathrm{supp}(\Omega)$). Even under misspecified orderings, the mask confines induced edges to the moralized skeleton. Hence, precision-support masking offers a scalable mechanism for suppressing false discoveries and is well suited to large graphs.

Furthermore, ReSP retains the MEC-recovery guarantee of SP under the SMR assumption, while also ensuring support confinement under ordering misspecification. For the soundness analysis, we impose an admissibility condition on the candidate set. We call $\mathbb{T}_p$ IC(0)-admissible if, for every $\pi \in \mathbb{T}_p$, masked IC(0) is well-defined and its output satisfies

$$(L_\pi L_\pi^\top) \circ (\mathbf{1} - M_\pi) = \mathbf{0}, \quad \forall\, \pi \in \mathbb{T}_p. \tag{5}$$

This condition allows SP's sparsity-based recovery argument to carry over to ReSP (Lemma C.1), whereas support confinement does not require it.

Algorithm 1 SCOPE: Sparse-Cholesky Pipeline

Require: Data $X \in \mathbb{R}^{n\times p}$, regularization λ, refinement level α, candidate set $\mathbb{T}_p$

1: $\widehat{\Omega} \leftarrow \textsc{EstimatePrecision}(X;\lambda)$
2: $\widehat{M} \leftarrow \textsc{BuildMask}(\widehat{\Omega})$
3: **for** $\pi \in \mathbb{T}_p$ **do**
4: $\quad \widehat{\Omega}_\pi \leftarrow P_\pi \widehat{\Omega} P_\pi^\top; \quad \widehat{M}_\pi \leftarrow P_\pi \widehat{M} P_\pi^\top$
5: $\quad$ *Factor construction under the ReSP constraints* (3)*:* $\widehat{L}_\pi \leftarrow \mathrm{IC0}(\widehat{\Omega}_\pi; \widehat{M}_\pi)$
6: $\quad$ *Finite-sample refinement:* $\widetilde{L}_\pi \leftarrow \textsc{RefitTest}(X_\pi; \mathrm{supp}(\widehat{L}_\pi), \alpha)$
7: **end for**
8: $\widehat{\pi} \in \arg\min_{\pi\in\mathbb{T}_p} \|\widetilde{L}_\pi\|_0$
9: **return** $\widehat{E} \leftarrow \widetilde{E}(\widehat{\pi}, \widetilde{L}_{\widehat{\pi}})$

Assumption 2 (Sparsest Markov Representation, Raskutti & Uhler (2018))**.** *Let π_0 be a reverse topological ordering of G_0. For each $\pi \in \mathbb{S}_p$, let L_π be the Cholesky factor of Ω_π and G_π its induced DAG. Assume that*

$$G_\pi \notin \mathcal{M}(G_0) \quad \Rightarrow \quad \|L_\pi\|_0 > \|L_{\pi_0}\|_0, \quad \forall\, \pi \in \mathbb{S}_p.$$

Theorem 3.2 (MEC recovery and support confinement)**.** *Consider a Gaussian linear SEM* (1) *satisfying Assumption 1, and let $M = \mathbb{I}\{\Omega \neq 0\}$ be a given precision-support mask.*

- ***(Soundness).*** *Suppose $\mathbb{T}_p$ is IC(0)-admissible under M, and contains a reverse topological ordering π_0 of G_0, and that Assumption 2 holds. Then every minimizer $(\hat{\pi}, L_{\hat{\pi}})$ of the ReSP objective* (3) *satisfies $G_{\hat{\pi}} \in \mathcal{M}(G_0)$.*
- ***(Support confinement).*** *For any candidate set $\mathbb{T}_p$, every feasible pair (π, L_π) for* (3) *satisfies $\widetilde{E}(\pi, L_\pi) \subseteq \mathrm{skel}(G_0^m)$.*

Theorem 3.2 establishes two complementary guarantees: ReSP recovers a DAG in the true MEC under the soundness conditions, while support confinement holds even under ordering misspecification. These guarantees motivate algorithms that construct and refine candidate structures within precision support.

4 SCOPE: A Scalable Instantiation of ReSP

ReSP provides a theoretical formulation for causal structure learning upon which practical algorithms can be built. At a high level, implementing ReSP requires constructing factors under the masked equality and support constraints and comparing their sparsity across candidate orderings. In this section, we present SCOPE as a simple, finite-sample implementation of ReSP.

SCOPE solves the finite-sample counterpart of (3):

$$\min_{\pi\in\mathbb{T}_p} \|\widetilde{L}_\pi\|_0 \text{ s.t. } \begin{cases} (\widehat{\Omega}_\pi - \widehat{L}_\pi \widehat{L}_\pi^\top) \circ \widehat{M}_\pi = \mathbf{0}, \\ \widehat{L}_\pi \circ (\mathbf{1} - \widehat{M}_\pi) = \mathbf{0}, \\ \widetilde{L}_\pi = \textsc{RefitTest}(X_\pi; \mathrm{supp}(\widehat{L}_\pi), \alpha). \end{cases} \tag{6}$$

Here, α is the refinement significance level, $X_\pi := X P_\pi^\top$, $\widehat{\Omega}_\pi := P_\pi \widehat{\Omega} P_\pi^\top$, and $\widehat{M}_\pi := P_\pi \widehat{M} P_\pi^\top$, where $\widehat{M}$ is an estimated precision-support mask constructed from $\widehat{\Omega}$. The masked factorization and final selection implement the ReSP constraints and sparsity objective, respectively.

Specifically, SCOPE uses graphical lasso, masked IC(0), and refit-and-test pruning; the one-pass version uses only an AMD ordering. Refinement only removes candidate edges, so the final support remains within $\widehat{M}$. Algorithm 1 summarizes the procedure, with each stage detailed in Appendix B and the experimental choice of λ specified in Appendix E.

4.1 Computational Complexity of One-Pass SCOPE

One-pass SCOPE uses a singleton candidate set $\mathbb{T}_p = \{\hat{\pi}_{\mathrm{AMD}}\}$, obtained by applying approximate minimum-degree (AMD), a classical and inexpensive fill-reducing heuristic, to $\widehat{\Omega}$ (Amestoy et al., 1996; George, 1989; Berman & Schnitger, 1990). In a DAG, nodes with few or no children often have lower degree in the moralized graph, so AMD can favor children before parents and thereby approximate a reverse ordering (Squires et al., 2020a). As with SP, more elaborate algorithms can optimize ReSP; we analyze the simplest implementation here, one-pass SCOPE.

We separate the cost of constructing a precision estimate, shared by all candidates, from the work performed for each ordering. Let T_{GL} denote the number of graphical-lasso solver iterations, and let $d_m \geq 1$ bound the maximum degree of the screened graph induced by $\widehat{M}$.

Theorem 4.1 (Complexity of one-pass SCOPE)**.** *Fix a singleton candidate set $\mathbb{T}_p = \{\pi\}$. Suppose $\widehat{\Omega}$ is estimated by graphical lasso at a given λ, $d_m < n$, and the masked factor construction and local regressions are well-defined. One pass of Algorithm 1, starting from X, runs in*

$$T(\pi) = O\big(np^2 + T_{\mathrm{GL}}p^3 + pnd_m^2\big)\,.$$

The first two terms account for forming the sample covariance and estimating precision. Given $\widehat{\Omega}$ and $\widehat{M}$, masked IC(0) costs $O(pd_m^2)$ per ordering and local refinement costs $O(pnd_m^2)$. The one-pass AMD heuristic is computed once; its ordering cost is separate from this fixed-ordering bound. For K specified candidate orderings, the shared estimation cost is paid once and the evaluation cost scales as $O(Kpnd_m^2)$.

ReSP's precision-support constraint prevents ordering-dependent fill from enlarging the local neighborhoods used in factor construction and refinement. For bounded d_m, the evaluation procedure above is therefore linear in the data size np, after precision estimation. This provides a concrete route to scalable causal discovery algorithms that combine masked evaluation with different ordering strategies. Appendix D compares computational costs with representative methods.

5 Numerical Experiments

We evaluate the finite-sample behavior of ReSP through one-pass SCOPE. The ablation studies in Section 5.1 isolate the effect of precision-support masking while varying refinement and ordering. Section 5.2 evaluates performance across regimes with large p, weak signals with large n, and $n < p$.

Experimental setup. For Tables 2 and 3, we generate data from linear SEMs (1) with independent Gaussian errors and in-degree bounded by 5. The three regimes in Table 3 examine complementary challenges. The weak-signal regime increases sample size to $n = 10^6$ at fixed $p = 500$ to examine recovery of small effects. The low-dimensional regime considers strong signals and abundant samples at moderate dimension, up to $p = 2000$. The high-dimensional regime considers strong signals at large dimension, reaching $p = 10^4$ with $n < p$.

CPDAG-skeleton F_1 and normalized structural Hamming distance (nSHD) for CPDAG recovery (Yang et al., 2022; Ban et al., 2023), averaged over completed trials, are used as performance metrics. For precision-based pipelines (SCOPE, FROSTY, and RFD), we measure runtime with $\widehat{\Omega}$ given, because precision-estimation time varies widely across solvers and tuning parameters. Appendix E provides experimental details and results under additional noise distributions.

5.1 Isolating the Effect of ReSP Masking

High-dimensional ablations illustrate the usefulness of ReSP's precision-support masking.

Masking improves accuracy and reduces downstream computation. Masking improves CPDAG-skeleton F_1 and nSHD under both orderings, with and without refinement. At $p = 10^4$ with AMD ordering, adding the mask to the refined pipeline raises F_1 from 0.213 to 0.582 and reduces discovery time from 222.4 to 15.5 s. Without refinement, masked factor construction itself takes longer than the unmasked implementation (8.2 versus 2.9 s), but already raises F_1 from 0.040 to 0.486. The computational speedup comes from restricting the candidate edges passed to downstream refinement, whose cost otherwise grows with fill.

Table 2: Effect of ReSP masking across refinement and ordering configurations at $n = 0.8p$. Entries report CPDAG-skeleton F_1, nSHD (in parentheses), and runtime in seconds (s) with $\widehat{\Omega}$ given.

Configuration			p=5000		p=10^4		Configuration			p=5000		p=10^4	
mask	refit	ordering	F_1 (nSHD)	s	F_1 (nSHD)	s	mask	refit	ordering	F_1 (nSHD)	s	F_1 (nSHD)	s
–	–	AMD	0.058 (27.5)	0.8	0.040 (46.2)	2.9	–	–	π_0	0.016 (105)	1.7	0.015 (124)	5.1
–	✓	AMD	0.299 (4.22)	40.0	0.213 (7.33)	222.4	–	✓	π_0	0.424 (3.11)	255.9	0.492 (2.86)	888.2
✓	–	AMD	0.596 (1.45)	2.2	0.486 (2.34)	8.2	✓	–	π_0	0.596 (1.79)	2.4	0.486 (2.73)	8.0
✓	✓	AMD	**0.668** (**1.17**)	4.2	**0.582** (**1.74**)	15.5	✓	✓	π_0	**0.780** (**1.18**)	5.7	**0.820** (**1.26**)	22.2

Table 3: CPDAG-skeleton F_1 and nSHD (in parentheses) across three regimes. Best values are bolded; second-best values are shaded in columns where at least three methods complete. ✗: timeout (1800 s); —: structurally inapplicable.

	(a) Weak signal (p=500)			(b) Low-dim (n=20p)				(c) High-dim (n=0.8p)		
Method	n=10^5	2×10^5	10^6	p=200	800	1200	2000	p=2000	5000	10^4
SCOPE	0.56 (1.62)	0.86 (0.67)	0.91 (0.52)	0.67 (1.20)	0.64 (1.25)	0.61 (1.26)	**0.64 (1.26)**	0.60 **(0.90)**	**0.67 (1.17)**	**0.58 (1.74)**
FROSTY	0.05 (36.65)	0.20 (7.75)	0.23 (6.41)	0.20 (7.66)	0.07 (22.90)	0.06 (23.57)	0.06 (27.42)	0.18 (5.15)	0.06 (27.47)	0.04 (46.24)
RFD	0.02 (95.89)	0.03 (68.73)	0.17 (10.26)	0.09 (21.36)	✗	✗	✗	✗	✗	✗
PC	0.58 (1.49)	0.65 (1.44)	0.69 (1.39)	0.70 (0.93)	0.68 (1.04)	✗	✗	**0.66** (1.00)	✗	✗
GES	**0.73 (0.77)**	**0.87 (0.65)**	**0.99 (0.49)**	0.50 (2.34)	✗	✗	✗	✗	✗	✗
BOSS	0.50 (0.87)	✗	✗	**0.99 (0.02)**	✗	✗	✗	✗	✗	✗
FLOP	0.01 (1.00)	0.18 (0.98)	✗	**0.99** (0.04)	**0.99 (0.03)**	**0.99 (0.04)**	✗	—	—	—
NOTEARS	0.00 (1.00)	0.00 (1.00)	0.00 (1.00)	✗	✗	✗	✗	✗	✗	✗
GOLEM	0.00 (1.00)	✗	✗	0.42 (0.79)	✗	✗	✗	✗	✗	✗
GSP	0.43 (2.54)	0.48 (2.49)	0.50 (2.45)	0.35 (3.97)	0.12 (15.24)	0.09 (19.71)	✗	✗	✗	✗

Masking remains beneficial under oracle ordering. With the oracle ordering π_0, adding masking to the refined pipeline raises F_1 from 0.492 to 0.820 at $p = 10^4$ and reduces discovery time from 888.2 to 22.2 s. This shows that masking remains beneficial even when the ordering is known, by controlling spurious fill induced by finite-sample estimation error. The oracle ordering also improves F_1 over AMD's 0.582 in the same pipeline, indicating the opportunity for better candidate orderings. These finite-sample results complement the population soundness guarantee in Theorem 3.2.

5.2 Performance Across Sample Regimes

Table 3 reports both accuracy metrics across the three regimes.

Weak-signal regime. At fixed $p = 500$, we increase n to examine whether the methods can use large sample sizes to recover weak effects. The CPDAG-skeleton F_1 of SCOPE increases from 0.56 to 0.86 and then 0.91, with nSHD decreasing at each step. GES leads this regime, while SCOPE achieves comparable accuracy at the larger sample sizes. These results show that one-pass SCOPE translates larger sample sizes into improved structural recovery under weak signals.

Low-dimensional regime. With abundant samples and strong signals, BOSS and FLOP attain near-perfect recovery where they complete. One-pass SCOPE has lower accuracy, but its skeleton F_1 remains between 0.61 and 0.67 across the four dimensions. At $p = 2000$, it retains $F_1 = 0.64$, whereas the evaluated search-based implementations are in the timeout category.

High-dimensional regime. At $n = 0.8p$, one-pass SCOPE produces structural estimates at every evaluated dimension through $p = 10^4$. At $p = 2000$, PC has higher F_1, whereas SCOPE has lower nSHD. The reported discovery times are 0.6 s for SCOPE with $\widehat{\Omega}$ given and 887 s for PC. At the larger dimensions, only SCOPE and FROSTY return results among the evaluated implementations.

Implications for ReSP. The leading baseline changes with the regime, whereas one-pass SCOPE achieves competitive accuracy and returns estimates in all ten settings. The $p = 10^4$ and $n = 10^6$ results demonstrate scalability in both dimension and sample size. Together, these results highlight the potential of support-constrained evaluation for making use of the information available in larger samples and broader variable sets.

Table 4: Directional agreement of estimated CPDAGs with the reference TF regulatory network.

	One-step Paths		Two-step Paths			One-step Paths		Two-step Paths	
Method	Fwd/Rev	Ratio	Fwd/Rev	Ratio	Method	Fwd/Rev	Ratio	Fwd/Rev	Ratio
PC	0.028/0.024	1.155	0.052/0.045	1.147	FLOP	0.035/0.024	1.435	0.057/0.046	1.230
GES	0.028/0.024	1.183	0.056/0.045	1.247	SCOPE	0.078/0.045	**1.723**	0.107/0.060	**1.785**

6 Breast Cancer Data Analysis

The TCGA study demonstrates the practical value of ReSP's support-constrained evaluation for joint structure learning over the full variable set, at a scale beyond the practical reach of most evaluated methods. We apply one-pass SCOPE to the TCGA mRNA expression dataset of invasive ductal breast carcinoma (The Cancer Genome Atlas Network, 2012), processed as in Koh et al. (2019), with $(n, p) = (599, 16325)$. Because transcription factors (TFs) regulate target-gene expression, we evaluate its directed CPDAG edges against the validated TF regulatory network of Koh et al. (2019).

We selected PC, GES, and FLOP for their strong performance across different simulation regimes. Because they did not scale to the full data dimensionality, we applied them to random disjoint subsets of approximately 800, 400, and 544 features, respectively, and took the union of their directed edges. These sizes permit the subset analyses to run while satisfying FLOP's $n > p$ requirement; details are provided in Appendix F.

We assess whether directed CPDAG paths follow TF→target directions. The forward (Fwd) and reverse (Rev) rates count supported pairs in each direction, normalized by the number of directed edges. Their ratios use unrounded rates. Reverse paths have the wrong direction but still indicate biologically meaningful associations.

Table 4 shows that full-feature SCOPE achieves the strongest directional agreement with the TF regulatory network, with the largest forward-to-reverse ratios for both path lengths. Similar results hold across additional random partitions (Appendix F). Random feature splitting can separate TFs from their targets and, through split-wise marginalization, distort the CI relations used for orientation or introduce spurious dependencies due to omitted variables. Joint analysis of all $16{,}325$ features avoids losses caused by partitioning, illustrating the practical value of the scale reached by this ReSP instantiation.

7 Discussion

ReSP is a causal discovery paradigm based on a support-constrained formulation of SP. It retains the sparsity objective while relaxing all-entry Cholesky equality, restricting factor construction and candidate evaluation to precision support. Under no-cancellation and SMR, ReSP minimizers induce DAGs in the true MEC when the candidate set is IC(0)-admissible and contains a reverse topological ordering. For arbitrary candidate sets, ReSP ensures graph support lies within the moralized skeleton.

One-pass SCOPE demonstrates the practical potential of ReSP. The ablation studies show that masking improves accuracy and reduces downstream computational cost. SCOPE delivers reliable performance across challenging regimes: weak signals with large sample sizes and high-dimensional settings reaching $p = 10^4$ with $n < p$. The full-feature TCGA analysis further illustrates the benefit of joint structure learning at scale. Together, these findings demonstrate the breadth attainable with a simple ReSP instantiation and motivate the development of more effective ReSP algorithms.

ReSP opens several directions for developing algorithms around support-constrained candidate evaluation. The oracle-ordering ablations motivate better ordering strategies, including swap- and tuck-type moves, multiple initial orderings, and the incorporation of prior knowledge. These strategies can share a precision estimate across candidate evaluations. Finite-sample performance also depends on the estimated precision-support mask, motivating alternative precision estimators and support-preserving refinement procedures. Further extensions include interventional and latent-variable settings, together with the structural assumptions and guarantees (Squires et al., 2020b; 2022).

AI USE STATEMENT

We used generative AI tools primarily to assist with manuscript drafting, organization, and language polishing, as well as literature checks. The authors reviewed and revised the AI-assisted material and take responsibility for the final content of this work.

ETHICS STATEMENT

This work develops causal discovery methodology and evaluates it using synthetic data and an existing TCGA gene-expression dataset processed and released by Koh et al. (2019). It involves no new participant recruitment or collection of human data. The inferred regulatory structures are evaluated for methodological purposes and are not presented as clinically validated findings.

REPRODUCIBILITY STATEMENT

The model assumptions and ReSP formulation are specified in Sections 2 and 3, with proofs in Appendix C. Section 4 and Appendix B describe the SCOPE implementation, while Appendices E and F document the experimental settings and evaluation procedures. We will publicly release a reproducibility package containing the implementation, experiment configurations, reproduction scripts, and instructions for data access and preprocessing.

REFERENCES

Patrick R. Amestoy, Timothy A. Davis, and Iain S. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905, 1996.

Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

Bryan Andrews, Joseph Ramsey, Ruben Sanchez-Romero, Jazmin Camchong, and Erich Kummerfeld. Fast scalable and accurate discovery of DAGs using the best order score search and grow-shrink trees. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. Causal structure learning supervised by large language model. *arXiv preprint arXiv:2311.11689*, 2023.

Joshua Bang and Sang-Yun Oh. Frosty: A high-dimensional scale-free bayesian network learning method. *Journal of Data Science*, 21(2):354–367, 2023. doi: 10.6339/23-JDS1097.

Albert-László Barabási and Zoltán N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004. doi: 10.1038/nrg1272.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6):2369–2429, 2012.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Inference on treatment effects after selection among high-dimensional controls. *Review of Economic Studies*, 81(2):608–650, 2014.

Piotr Berman and Georg Schnitger. On the performance of the minimum degree ordering for Gaussian elimination. *SIAM Journal on Matrix Analysis and Applications*, 11(1):83–88, 1990. doi: 10.1137/0611005.

Matthias Bollhöfer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk. Large-scale sparse inverse covariance matrix estimation. *SIAM Journal on Scientific Computing*, 41(1):A380–A401, 2019. doi: 10.1137/17M1147615.

Edward Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10:186–198, 2009. doi: 10.1038/nrn2575.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

Pedro Cisneros-Velarde, Alexander Petersen, and Sang-Yun Oh. Distributionally robust formulation and model selection for the graphical lasso. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 756–765. PMLR, 2020. URL https://proceedings.mlr.press/v108/cisneros20a.html.

Timothy A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Alan George. The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19, 1989.

Cho-Jui Hsieh, Matyas A. Sustik, Inderjit S. Dhillon, Pradeep Ravikumar, and Russell A. Poldrack. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems 26*, pp. 3165–3173, 2013.

Cho-Jui Hsieh, Matyas A. Sustik, Inderjit S. Dhillon, and Pradeep Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15(106):2911–2947, 2014.

Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltán N. Oltvai, and Albert-László Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000. doi: 10.1038/35036627.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. doi: 10.18637/jss.v047.i11.

Hiromi W. L. Koh, Damian Fermin, Christine Vogel, Kwok Pui Choi, Rob M. Ewing, and Hyungwon Choi. iOmicsPASS: Network-based integration of multiomics data for predictive subnetwork discovery. *npj Systems Biology and Applications*, 5(1):22, July 2019. ISSN 2056-7189. doi: 10.1038/s41540-019-0099-y.

Wai-Yin Lam, Bryan Andrews, and Joseph D. Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 180 of *Proceedings of Machine Learning Research*, pp. 1052–1062. PMLR, 2022.

Chih-Jen Lin and Jorge J Moré. Incomplete cholesky factorizations with limited memory. *SIAM Journal on Scientific computing*, 21(1):24–45, 1999.

Sten Linnarsson and Sarah A. Teichmann. Single-cell genomics: coming of age. *Genome Biology*, 17:97, 2016. doi: 10.1186/s13059-016-0960-x.

Han Liu, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 15(88):3065–3105, 2014.

J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31(137):148–162, 1977. doi: 10.1090/S0025-5718-1977-0438681-4.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. doi: 10.1137/S003614450342480.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and DAG constraints for learning linear DAGs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 17943–17954, 2020.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011. doi: 10.1214/11-EJS631.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2 edition, 2003.

Jennifer Scott and Miroslav Tůma. *Algorithms for Sparse Linear Systems*. Nečas Center Series. Springer International Publishing, Cham, 2023. ISBN 978-3-031-25819-0 978-3-031-25820-6. doi: 10.1007/978-3-031-25820-6. URL https://link.springer.com/10.1007/978-3-031-25820-6.

Liam Solus, Yuhao Wang, and Caroline Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021.

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

Chandler Squires. `causaldag`*: creation, manipulation, and learning of causal models*, 2018. URL https://github.com/uhlerlab/causaldag.

Chandler Squires, Joshua Amaniampong, and Caroline Uhler. Efficient permutation discovery in causal DAGs. *arXiv preprint arXiv:2011.03610*, 2020a.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In Jonas Peters and David Sontag (eds.), *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 1039–1048. PMLR, 03–06 Aug 2020b. URL https://proceedings.mlr.press/v124/squires20a.html.

Chandler Squires, Annie Yun, Eshaan Nichani, Raj Agrawal, and Caroline Uhler. Causal structure discovery between clusters of nodes induced by latent factors. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 669–687. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/squires22a.html.

Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 584–590, Arlington, Virginia, USA, 2005. AUAI Press.

The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, October 2012. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature11412.

C. Tran, Pedro Cisneros-Velarde, Alexander Petersen, and Sang-Yun Oh. robust-selection: Robust selection (robsel) for graphical lasso regularization. PyPI, 2020. URL https://pypi.org/project/robust-selection/. Version 0.0.8.

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014. doi: 10.1038/nbt.2859.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The Max-Min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

Larry Wasserman and Kathryn Roeder. High dimensional variable selection. *Annals of Statistics*, 37(5A):2178–2201, 2009.

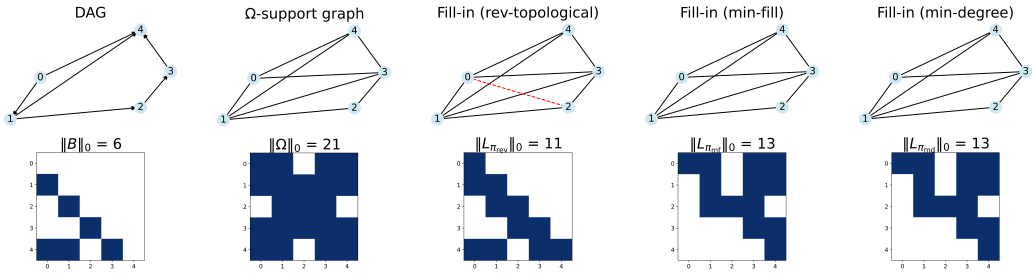
Marcel Wienöbst, Leonard Henckel, and Sebastian Weichwald. Embracing discrete search: A reasonable approach to causal structure learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Yuqin Yang, Mohamed S Nafea, AmirEmad Ghassami, and Negar Kiyavash. Causal discovery in linear structural causal models with deterministic relations. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang (eds.), *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 944–993. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/yang22a.html.

Qiaoling Ye, Arash A Amini, and Qing Zhou. Optimizing regularized cholesky score for order-based learning of bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3555–3572, 2021.

Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gCastle: A python toolbox for causal discovery. arXiv preprint arXiv:2111.15155, 2021.

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



(a) DAG (b) Moral graph (c) rev-topo (d) min-fill (e) min-degree

Figure 2: DAG, moral graph, and triangulations under rev-topo, min-fill, and min-degree orderings, with sparsity patterns of B, Ω, and L. Red dashed edges indicate fill-in.

A A Mismatch Between Fill Reduction and SP Sparsity

This example shows that minimizing fill in the undirected elimination graph need not yield a sparse triangular factor under the SP criterion.

Figure 2 shows the DAG, its moral graph, and the factors obtained from the reverse-topological, minimum-fill, and minimum-degree orderings $\pi_{\mathrm{rev}} = [4, 3, 2, 1, 0]$, $\pi_{\mathrm{mf}} = [0, 2, 1, 3, 4]$, and $\pi_{\mathrm{md}} = [2, 0, 1, 3, 4]$. The common child at node 4 induces the moral edges $(0, 3)$ and $(1, 3)$, which are absent from the DAG skeleton. The reverse ordering incurs the fill edge $(0, 2)$ but yields the sparsest factor, with $\|L_{\pi_{\mathrm{rev}}}\|_0 = 11$; the other two orderings achieve zero elimination-graph fill but yield $\|L_{\pi_{\mathrm{mf}}}\|_0 = \|L_{\pi_{\mathrm{md}}}\|_0 = 13$.

Under π_{rev}, node 4 is removed first, after which the coupling between node 3 and $\{0, 1\}$ mediated through their common child vanishes numerically. The undirected elimination graph retains the corresponding moral edges and therefore predicts fill that does not materialize in the numerical factor. Thus, elimination-graph fill can overestimate Cholesky support, and fill reduction alone is not a reliable surrogate for SP sparsity. This mismatch motivates restricting feasibility to $\mathrm{supp}(\Omega)$ before comparing sparsity across orderings.

B Finite-Sample Implementation Details for SCOPE

This section details the finite-sample implementation of SCOPE summarized in Algorithm 1.

B.1 Stage 0: Precision Estimation and Mask Construction

Stage 0 supplies an estimated precision-support screen for ReSP. SCOPE uses graphical lasso followed by hard-thresholding; other precision estimators can be substituted without changing the ReSP formulation. Let $\widehat{\Sigma} := n^{-1} \sum_{i=1}^{n} X_i X_i^\top$ be the empirical covariance of the centered observations. The graphical lasso estimator solves

$$\widehat{\Omega}(\lambda) \in \arg\min_{K \succ 0} \left\{ \mathrm{tr}(K\widehat{\Sigma}) - \log\det(K) + \lambda \|K\|_{1,\mathrm{off}} \right\}. \tag{7}$$

We use the SQUIC solver (Bollhöfer et al., 2019); the regularization protocol is specified in Appendix E.

For $c_{\mathrm{thr}} > 0$, define

$$\widehat{\Omega}^{\mathrm{thr}}_{ij} := \widehat{\Omega}(\lambda)_{ij} \mathbb{I}\{|\widehat{\Omega}(\lambda)_{ij}| \geq c_{\mathrm{thr}}\lambda\}, \quad i \neq j, \qquad \widehat{\Omega}^{\mathrm{thr}}_{ii} := \widehat{\Omega}(\lambda)_{ii}. \tag{8}$$

We set $c_{\mathrm{thr}} = 1$ and use $\widehat{M} := \mathbb{I}\{\widehat{\Omega}^{\mathrm{thr}} \neq 0\}$, including the diagonal, unless otherwise stated. The thresholded matrix determines the mask; the graphical-lasso estimate supplies the values for masked factorization. Existing support-estimation results motivate thresholding when estimation error is separated from nonzero signal strength (Ravikumar et al., 2011; Liu et al., 2012). Here, precision estimation is an input stage of the finite-sample implementation, while the ReSP recovery theorem concerns the population mask.

B.2 Stage 1: Masked IC(0) Factor Construction

For a fixed ordering π, Stage 1 computes a masked IC(0) factor $\widehat{L}_\pi = \mathrm{IC0}(\widehat{\Omega}_\pi; \widehat{M}_\pi)$, a lower-triangular matrix satisfying the masked feasibility constraints in (4) with (Ω_π, M_π) replaced by $(\widehat{\Omega}_\pi, \widehat{M}_\pi)$.

B.3 Stage 2: Finite-Sample Refinement

Stage 2 applies refit-and-test pruning to the support of $\widehat{L}_\pi$. Consistent with the edge convention in Section 2.2, define

$$\widehat{\mathrm{Pa}}_\pi(j) := \{i > j : (\widehat{L}_\pi)_{ij} \neq 0\}.$$

These are the ordering positions of candidate parents of variable $\pi(j)$. For each j, we fit

$$\widehat{\beta}_{j,\widehat{\mathrm{Pa}}_\pi(j)} \in \arg\min_{\beta \in \mathbb{R}^{|\widehat{\mathrm{Pa}}_\pi(j)|}} \left\| X_{\pi(j)} - X_{\pi(\widehat{\mathrm{Pa}}_\pi(j))}\beta \right\|_2^2,$$

where the design matrix contains the columns indexed by $\{\pi(i) : i \in \widehat{\mathrm{Pa}}_\pi(j)\}$. Let $p_{j,i}$ be the usual two-sided regression t-test p-value for the coefficient of candidate parent $\pi(i)$ in this regression. We prune the factor according to

$$(\widetilde{L}_\pi)_{ij} := (\widehat{L}_\pi)_{ij}\mathbb{I}\{i \in \widehat{\mathrm{Pa}}_\pi(j),\ p_{j,i} \leq \alpha\}, \qquad i > j,$$

leaving the diagonal unchanged. Thus, refinement preserves acyclicity and cannot introduce an edge outside $\widehat{M}$. This local pruning follows post-selection refitting and screen-and-clean ideas (Belloni et al., 2012; 2014; Wasserman & Roeder, 2009).

B.4 Stage 3: Sparsity-Based Selection

Finally, SCOPE selects an ordering

$$\widehat{\pi} \in \arg\min_{\pi \in \mathbb{T}_p} \|\widetilde{L}_\pi\|_0$$

and returns the directed edge set $\widehat{E} := \widetilde{E}(\widehat{\pi}, \widetilde{L}_{\widehat{\pi}})$ induced as in Section 2.2. The criterion may admit ties because sparsity objectives can be indifferent among orderings within the same MEC; any minimizer may then be selected (Raskutti & Uhler, 2018; Solus et al., 2021).

C Proofs of Theoretical Results

C.1 Proof of Lemma 3.1

Under Assumption 1, the direct-edge and shared-child decomposition of the precision entries gives (Loh & Bühlmann, 2014, Theorem 2 and Assumption 1)

$$\mathrm{supp}(\Omega) \setminus \Delta_p = \mathrm{skel}(G_0^m).$$

Permutation only relabels indices. With the convention $\pi(i)$ denoting the variable at ordering position i,

$$\pi\big(\mathrm{supp}(\Omega_\pi) \setminus \Delta_p\big) = \mathrm{supp}(\Omega) \setminus \Delta_p = \mathrm{skel}(G_0^m).$$

Combining this identity with $\mathrm{skel}(G_0) \subseteq \mathrm{skel}(G_0^m)$ proves the claim.

C.2 Proof of Theorem 3.2

C.2.1 Proof for Soundness

We first record a simple consequence of precision-support masking and IC(0)-admissibility.

Lemma C.1 (Exact factorization under precision-support masking and IC(0)-admissibility)**.** *Assume $M = \mathbb{I}\{\Omega \neq 0\}$ (with the diagonal included) and let $\mathbb{T}_p$ be IC(0)-admissible. Then, for every $\pi \in \mathbb{T}_p$, the masked IC(0) output $L_\pi = \mathrm{IC0}(\Omega_\pi; M_\pi)$ satisfies*

$$\Omega_\pi = L_\pi L_\pi^\top, \qquad \textit{where} \quad \Omega_\pi := P_\pi \Omega P_\pi^\top,\ M_\pi := P_\pi M P_\pi^\top.$$

Proof. Fix $\pi \in \mathbb{T}_p$. Since $M = \mathbb{I}\{\Omega \neq 0\}$, we have $\Omega \circ (\mathbf{1} - M) = \mathbf{0}$, and relabeling yields

$$\Omega_\pi \circ (\mathbf{1} - M_\pi) = \mathbf{0}.$$

By the masked IC(0) invariants (4),

$$(\Omega_\pi - L_\pi L_\pi^\top) \circ M_\pi = \mathbf{0}.$$

Moreover, IC(0)-admissibility (5) gives

$$(L_\pi L_\pi^\top) \circ (\mathbf{1} - M_\pi) = \mathbf{0}.$$

Combining the last two equations and splitting over M_π and $(\mathbf{1} - M_\pi)$,

$$\Omega_\pi - L_\pi L_\pi^\top = \big[(\Omega_\pi - L_\pi L_\pi^\top) \circ M_\pi\big] + \big[(\Omega_\pi - L_\pi L_\pi^\top) \circ (\mathbf{1} - M_\pi)\big] = \mathbf{0}.$$

Here, the second term vanishes because

$$(\Omega_\pi - L_\pi L_\pi^\top) \circ (\mathbf{1} - M_\pi) = \Omega_\pi \circ (\mathbf{1} - M_\pi) - (L_\pi L_\pi^\top) \circ (\mathbf{1} - M_\pi) = \mathbf{0} - \mathbf{0}.$$

Hence $\Omega_\pi = L_\pi L_\pi^\top$. □

Lemma C.1 implies that for every $\pi \in \mathbb{T}_p$, the constraints in (3) reduce to the exact factorization constraint

$$L_\pi L_\pi^\top = \Omega_\pi.$$

Hence (3) is equivalent to minimizing the sparsity of the exact Cholesky factor of Ω_π over candidate orderings:

$$\min_{\pi \in \mathbb{T}_p} \|L_\pi\|_0 \quad \text{s.t.} \quad L_\pi L_\pi^\top = \Omega_\pi.$$

This is the same sparsity-based comparison underlying SP learning in Raskutti & Uhler (2018, Def. 2.2 and Thm. 2.3) so the remaining steps follow the standard SMR argument.

Let $(\hat{\pi}, L_{\hat{\pi}})$ be any minimizer of (3) over $\mathbb{T}_p$. By feasibility of (π_0, L_{π_0}) and optimality,

$$\|L_{\hat{\pi}}\|_0 \leq \|L_{\pi_0}\|_0. \tag{9}$$

If $G_{\hat{\pi}} \notin \mathcal{M}(G_0)$, then Assumption 2 implies

$$\|L_{\hat{\pi}}\|_0 > \|L_{\pi_0}\|_0,$$

which contradicts (9). Therefore $G_{\hat{\pi}} \in \mathcal{M}(G_0)$.

C.2.2 Proof of Support Confinement

Fix an arbitrary candidate set $\mathbb{T}_p$ and any feasible pair (π, L_π) for (3). Recall the population mask $M := \mathbb{I}\{\Omega \neq 0\}$ and its permuted version $M_\pi := P_\pi M P_\pi^\top = \mathbb{I}\{\Omega_\pi \neq 0\}$, where $\Omega_\pi := P_\pi \Omega P_\pi^\top$.

By feasibility, L_π satisfies the support constraint

$$L_\pi \circ (1 - M_\pi) = 0. \tag{10}$$

Consequently, for any $k > j$,

$$(L_\pi)_{kj} \neq 0 \implies (M_\pi)_{kj} = 1 \implies (\Omega_\pi)_{kj} \neq 0.$$

Since $(\Omega_\pi)_{kj} = \Omega_{\pi(k),\pi(j)}$ and Ω is symmetric, the condition $(\Omega_\pi)_{kj} \neq 0$ is equivalent to both $\Omega_{\pi(k),\pi(j)} \neq 0$ and $\Omega_{\pi(j),\pi(k)} \neq 0$, i.e.,

$$(L_\pi)_{kj} \neq 0 \implies (\pi(k), \pi(j)), (\pi(j), \pi(k)) \in \mathrm{supp}(\Omega).$$

By Assumption 1, $\mathrm{supp}(\Omega) \setminus \Delta_p = \mathrm{skel}(G_0^m)$. Therefore every directed edge induced by a nonzero entry of L_π connects a pair adjacent in $\mathrm{skel}(G_0^m)$, and hence

$$\widetilde{E}(\pi, L_\pi) := \big\{\, (\pi(k), \pi(j)) : \; k > j, \; (L_\pi)_{kj} \neq 0 \,\big\} \;\subseteq\; \mathrm{skel}(G_0^m),$$

which proves the claim.

C.3 Proof of Theorem 4.1

We prove Theorem 4.1 by decomposing the runtime across stages.

Fix an ordering π and consider one pass of Algorithm 1 (Stages 0–2). Let $d_m \geq 1$ bound the maximum degree of the screened graph encoded by the Stage 0 mask $\widehat{M}$ and let $\widehat{M}_\pi := P_\pi \widehat{M} P_\pi^\top$ be the permuted mask. We bound the runtime by summing the stagewise costs.

Stage 0 (precision estimation). Forming the dense empirical covariance from $X \in \mathbb{R}^{n\times p}$ costs $O(np^2)$. At a given regularization level λ, graphical lasso with T_{GL} solver iterations costs $O(T_{\mathrm{GL}}p^3)$ in the dense worst-case analysis (Friedman et al., 2008; Hsieh et al., 2013; 2014). Constructing the thresholded mask takes at most $O(p^2)$, which is absorbed by covariance construction. Thus,

$$T_0 = O\big(np^2 + T_{\mathrm{GL}}p^3\big)\,.$$

Per-dataset bootstrap calibration, if used, adds $O(Bnp^2)$ for B covariance replicates; it is excluded from the bound conditional on λ.

Stage 1 (masked IC(0)). We show that masked IC(0) runs in $O(p\,d_m^2)$ time under the degree bound d_m. Write $\widehat{L}_\pi = \mathrm{IC0}(\widehat{\Omega}_\pi; \widehat{M}_\pi)$ for Stage 1 output. For each row index $k \in [p]$, define the active predecessor set

$$S_k := \{j < k : \ (\widehat{M}_\pi)_{kj} = 1\}.$$

By definition of d_m as a maximum degree bound for the screened graph, we have $|S_k| \leq d_m$ for all k.

Masked IC(0) computes, for each k, the entries $\{(\widehat{L}_\pi)_{kj} : j \in S_k\}$ and the diagonal $(\widehat{L}_\pi)_{kk}$ using only indices in S_k. Concretely, for each fixed pair $j \in S_k$, the IC(0) update for $(\widehat{L}_\pi)_{kj}$ is of the generic form

$$(\widehat{L}_\pi)_{kj} \ = \ \frac{1}{(\widehat{L}_\pi)_{jj}}\Big((\widehat{\Omega}_\pi)_{kj} - \sum_{\ell \in S_k \cap S_j} (\widehat{L}_\pi)_{k\ell}(\widehat{L}_\pi)_{j\ell}\Big),$$

and the diagonal update has the form

$$(\widehat{L}_\pi)_{kk}^2 \ = \ (\widehat{\Omega}_\pi)_{kk} - \sum_{\ell \in S_k} (\widehat{L}_\pi)_{k\ell}^2,$$

with the breakdown condition corresponding to $(\widehat{L}_\pi)_{kk}^2 \leq 0$.

We now count arithmetic operations. For each k:

- Computing $(\widehat{L}_\pi)_{kk}$ requires a sum over $t \in S_k$ and thus costs $O(|S_k|) \leq O(d_m)$ operations.
- For each $j \in S_k$, computing $(\widehat{L}_\pi)_{kj}$ requires summing over $S_k \cap S_j$. Since $S_k \cap S_j \subseteq S_k$, this costs $O(|S_k \cap S_j|) \leq O(|S_k|) \leq O(d_m)$ operations. There are $|S_k| \leq d_m$ such indices j.

Hence the total cost at row k is

$$O(d_m) + \sum_{j \in S_k} O(d_m) \ = \ O(d_m) + O(|S_k|\, d_m) \ = \ O(d_m^2).$$

Summing over $k = 1, \ldots, p$ yields

$$T_1 = \sum_{k=1}^{p} O(d_m^2) = O(p\, d_m^2).$$

Stage 2 (RefitTest). Stage 2 performs p local regressions. For each node k, the candidate predictor set is the parent-candidate set implied by the support of the Stage 1 factor, hence its size is at most the screened degree bound:

$$s_k := |\widehat{\mathrm{Pa}}_\pi(k)| \leq d_m.$$

Table 5: Representative computational bounds. SCOPE and FROSTY include sample covariance construction and a graphical-lasso solve at a given λ; SCOPE uses one fixed ordering. The conditioning and scope of the remaining bounds are described below.

Method	Category	Cost or search bound
SCOPE	ReSP instantiation	$O\big(np^2 + T_{\mathrm{GL}}p^3 + pnd_m^2\big)$
FROSTY	Sparse Cholesky	$O\big(np^2 + T_{\mathrm{GL}}p^3\big)$
RFD	Sparse Cholesky	$O(p^{w+3})$
PC	Constraint-based	$O(p^{q+2})$
GSP	SP search	$O(p^{q+2})$
GES	Score-based	$O(p^2 2^p)$
MMHC	Score-based	$O(p^2 2^p)$
NOTEARS	Continuous opt.	$O(T(n \vee p)p^2)$
GOLEM	Continuous opt.	$O(np^2 + Tp^3)$

Consider the regression of $X_{\pi(k)}$ on the columns indexed by $\{\pi(i) : i \in \widehat{\mathrm{Pa}}_\pi(k)\}$, with s_k predictors. Forming the Gram matrix and cross-product costs

$$X^\top X : O(ns_k^2), \qquad X^\top y : O(ns_k).$$

Solving the resulting $s_k \times s_k$ linear system by a generic dense method costs $O(s_k^3)$. Under the stated regime $d_m < n$, we have $s_k \le d_m < n$, so $s_k^3 \le s_k^2 n$ and the solve is dominated by $O(ns_k^2)$. Computing residuals and t-statistics adds at most $O(ns_k + s_k^2)$ and is also dominated by $O(ns_k^2)$. Therefore, the total cost per node is $O(ns_k^2) \le O(nd_m^2)$ for each k. Summing over $k = 1, \dots, p$ gives

$$T_2 = O(p\, n\, d_m^2).$$

Combining the stagewise bounds,

$$T(\pi) = T_0 + T_1 + T_2 = O\big(np^2 + T_{\mathrm{GL}}\, p^3 + p\, d_m^2 + p\, n\, d_m^2\big)\,.$$

Since $n \ge 1$, the term $p\, d_m^2$ is absorbed by $p\, n\, d_m^2$, yielding

$$T(\pi) = O\big(np^2 + T_{\mathrm{GL}}\, p^3 + p\, n\, d_m^2\big)\,,$$

which proves the theorem.

D Computational Bounds

In Table 5, q is the maximum conditioning-set size, w the RFD search depth, and T the number of NOTEARS or GOLEM iterations (Kalisch & Bühlmann, 2007; Solus et al., 2021; Chickering, 2002; Tsamardinos et al., 2006; Zheng et al., 2018; Ng et al., 2020; Squires et al., 2020a; Bang & Oh, 2023). All bounds treat tuning parameters (significance levels, penalties, and regularization levels) as given. The RFD entry is a search bound conditional on a precision estimate. The PC and GSP entries summarize conditioning-set search counts, and the GES and MMHC entries summarize graph-search bounds; they do not include all implementation-dependent costs of individual statistical queries. The SCOPE bound separates shared data processing from evaluation and retains explicit dependence on n and d_m. BOSS and FLOP are omitted because their papers do not give end-to-end worst-case bounds; FLOP reports only the $O(k^2)$ cost of a dynamic Cholesky update for a parent set of size k (Andrews et al., 2023; Wienöbst et al., 2026).

E Numerical Experiments

Implementation details. For precision estimation, we solve graphical lasso using SQUIC (Bollhöfer et al., 2019). With $n \le p$, we calibrate λ per dataset using the DRO-based Robust Selection (RobSel) rule with its significance level set to 0.01, implemented by the `robust-selection` package (Cisneros-Velarde et al., 2020; Tran et al., 2020). In the $n > p$ sweeps, per-dataset DRO calibration tended to produce overly dense masks. We therefore hold

a reference-calibrated λ fixed within each sweep: $\lambda = 0.10$ for the low-dimensional regime, calibrated at $(p, n) = (200, 4000)$, and $\lambda = 0.005$ for the weak-signal regime, calibrated at $(p, n) = (500, 10^6)$. Stage 2 uses $\alpha = 0.01$ throughout the simulations. If SQUIC fails, the corresponding trial is excluded from the reported averages. All algorithms use four BLAS threads and a per-algorithm timeout of 1800 seconds. Unless otherwise stated, remaining settings follow the implementations below.

Evaluation. For methods that output a DAG, we convert the estimated DAG to a CPDAG using the `causaldag` Python package (Squires, 2018). We define normalized structural Hamming distance (nSHD) as the SHD between the estimated and true CPDAGs divided by the number of true edges. If this conversion exceeds 60 seconds for a dense output, we terminate it and use the DAG skeleton as a CPDAG proxy. Reported runtimes exclude conversion; the dense exact-Cholesky configurations in Table 2 reach this cap at every scale. For attempted settings, we target 30 replicates with $p < 10^4$ and 10 with $p = 10^4$. Accuracy and runtime means use completed trials; timed-out runs and failed precision estimates are excluded.

Failure modes in Table 3. The ✗ symbol denotes the timeout category, which combines budget exhaustion with larger settings not attempted after a timeout at an easier size. The — symbol denotes structural inapplicability: FLOP requires a positive-definite sample covariance, which is unavailable for $n < p$.

Synthetic linear SEM generation. Synthetic data is generated from a linear SEM (1) with independent errors. To match the sparsity regime considered throughout the paper, we sample bounded in-degree DAGs by letting each node select at most d parents among the earlier nodes in a random ordering. For graphs with $p > 1000$, we generate DAGs in a blockwise manner using blocks of size at most 1000. Edge coefficients are drawn uniformly from $\pm[0.6, 0.8]$ in the low- and high-dimensional regimes and from $[0.005, 0.02]$ in the weak-signal regime. In the low-dimensional sweep, $n = 20p$ without a sample-size cap for every method. Error variances are sampled independently from $[0.8, 1.0]$. To assess robustness beyond Gaussianity, the noise terms are drawn from one of the following distributions: standard normal, Student-t with 10 degrees of freedom, or uniform. In all cases, the noise is centered to have mean zero and then rescaled so that its variance matches the sampled value in $[0.8, 1.0]$.

We summarize below the software used for each baseline and the corresponding user-specified hyperparameters, with all unspecified options left at their package defaults.

- **PC** (Kalisch & Bühlmann, 2007).
  Implementation: `pcalg::pc` (R) via `rpy2` (Kalisch et al., 2012).
  The algorithm uses the Gaussian Fisher-z CI test with $\alpha = 0.01$. We set the maximum conditioning-set size to `m.max`$= 3$ and use the original skeleton routine without the stable variant to keep the computational cost practical at our problem sizes.
- **GES** (Chickering, 2002).
  Implementation: `pcalg::ges` (R) with `GaussL0penObsScore` via `rpy2` (Kalisch et al., 2012).
  We use the standard ℓ_0-penalized Gaussian score with penalty $\lambda = 0.5\log(n)$ (i.e., `lambda_mult`$= 0.5$), and otherwise keep defaults.
- **GSP** (Solus et al., 2021).
  Implementation: `causaldag` (Python) (Squires, 2018).
  We run GSP with a partial-correlation CI tester using $\alpha = 0.01$. We keep the remaining options at their defaults.
- **GRaSP** (Lam et al., 2022).
  Implementation: `causal-learn` version 0.1.4.3, using its Python `grasp` routine.
  We call `grasp(X, score_func="local_score_BIC")` with the remaining arguments at their defaults, including `depth`$= 3$. In this version, the score is evaluated from a covariance matrix computed once, with `lambda_value`$= 2$; its local criterion is $n\log(\widehat{\sigma}^2) + 2\log(n)|\mathrm{PA}|$. No seed is supplied for the routine's internal Python random shuffling. The `pc_alpha` argument in the experimental wrapper is unused by GRaSP.

Table 6: Large-p performance at $p = 10^4$ under t-distributed noise; entries are mean (standard error).

(a) t noise ($d = 1$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.994** (0.004)	**0.050** (0.043)
$0.5p$	FROSTY	0.642 (0.007)	1.116 (0.034)
p	SCOPE	**0.951** (0.012)	**0.264** (0.050)
p	FROSTY	0.635 (0.008)	1.152 (0.041)

(b) t noise ($d = 5$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.614** (0.117)	**0.917** (0.046)
$0.5p$	FROSTY	0.024 (0.001)	78.041 (4.258)
p	SCOPE	**0.633** (0.031)	**1.322** (0.189)
p	FROSTY	0.023 (0.001)	85.968 (2.966)

GRaSP exceeded the per-trial time budget in every setting we attempted, so it is omitted from Table 3.

- **BOSS** (Andrews et al., 2023).
  Implementation: `Boss` with `PermutationSearch` in Tetrad, accessed through `py-tetrad`/JPype.
  We use the `SemBicScore` with penalty discount 2.0 and `num_starts`$= 5$, and otherwise keep defaults.
- **FLOP** (Wienöbst et al., 2026).
  Implementation: the authors' `flopsearch` package (Python/Rust).
  We use the standard BIC penalty $\lambda_{\mathrm{BIC}} = 0.5 \log n$ with `restarts`$= 5$ and otherwise keep defaults. `flopsearch` returns a CPDAG directly, which we translate from its output encoding without an intermediate DAG conversion.
- **NOTEARS** (Zheng et al., 2018).
  Implementation: `gCastle Notears` (Python) (Zhang et al., 2021).
  The reported runs use `lambda1`$= 0.1$, `loss_type="l2"`, and `max_iter`$= 20$, with `h_tol`$= 10^{-8}$ and `rho_max`$= 10^{16}$. The final `w_threshold`$= 0.3$ is applied by gCastle before returning the binary `causal_matrix`. If that binary graph contains a cycle, the wrapper deletes edges until it is acyclic; all nonzero edge magnitudes are tied at this stage. Remaining options use the package defaults.
- **GOLEM** (Ng et al., 2020).
  Implementation: `gCastle GOLEM` (Python) (Zhang et al., 2021).
  The reported runs use `lambda_1`$= 0.1$, `lambda_2`$= 0.1$, `equal_variances=False`, and `num_iter`$= 1000$, with Adam learning rate 10^{-3}, seed 1, and no supplied initialization. The package post-processing applies `graph_thres`$= 0.3$, removes remaining cycles using weighted edges, and returns a binary DAG. Remaining options use the package defaults.
- **FROSTY** (Bang & Oh, 2023) and **RFD** (Squires et al., 2020a).
  Implementation: Python implementations provided by the authors.
  Both methods take a precision estimate as input. We use the same SQUIC solver and regime-specific regularization protocol described above for SCOPE.

E.1 High-Dimensional Performance

Tables 6 and 7 report additional $p = 10^4$ results at $n \in \{0.5p, p\}$ under t, normal, and uniform noise, complementing the $n = 0.8p$ grid in the main text (Table 3). Across all three distributions and both values of d, one-pass SCOPE outperforms FROSTY in both metrics, with the largest gaps when $d = 5$.

F Breast Cancer Data Analysis

Data description. We use the TCGA invasive ductal breast carcinoma mRNA data processed by Koh et al. (2019), with $(n, p) = (599, 16325)$; the processed files are available at https://github.com/Hiromikwl/DataCodes_iOmicsPASS. The data were acquired through the Genomics Data Commons, aligned to GRCh38, and mapped from ENSEMBL identifiers to HGNC gene symbols (The Cancer Genome Atlas Network, 2012; Koh et al., 2019). We evaluate directed paths against the accompanying regulatory network of 2,486 transcription factors and 14,796 target genes.

Table 7: Large-p performance at $p = 10^4$ under normal and uniform noise; entries are mean (standard error).

(a) Normal noise ($d = 1$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.965** (0.006)	**0.163** (0.078)
$0.5p$	FROSTY	0.627 (0.010)	1.193 (0.050)
p	SCOPE	**0.911** (0.005)	**0.345** (0.109)
p	FROSTY	0.626 (0.008)	1.196 (0.042)

(b) Normal noise ($d = 5$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.580** (0.204)	**1.166** (0.074)
$0.5p$	FROSTY	0.022 (0.002)	88.658 (12.876)
p	SCOPE	**0.562** (0.021)	**1.864** (0.109)
p	FROSTY	0.021 (0.000)	93.024 (1.093)

(c) Uniform noise ($d = 1$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.948** (0.003)	**0.268** (0.062)
$0.5p$	FROSTY	0.622 (0.011)	1.218 (0.056)
p	SCOPE	**0.903** (0.007)	**0.400** (0.075)
p	FROSTY	0.624 (0.011)	1.207 (0.056)

(d) Uniform noise ($d = 5$)

n	Method	F_1	nSHD
$0.5p$	SCOPE	**0.642** (0.020)	**1.241** (0.059)
$0.5p$	FROSTY	0.022 (0.000)	85.750 (0.638)
p	SCOPE	**0.497** (0.175)	**1.835** (0.302)
p	FROSTY	0.020 (0.002)	97.246 (10.738)

Table 8: Directional agreement of estimated CPDAGs with the reference TF regulatory network across random feature partitions.

			One-step Paths		Two-step Paths	
Method	Random partition	# of directed edges	Fwd/Rev	Ratio	Fwd/Rev	Ratio
PC	1	62,422	0.028/0.024	1.155	0.052/0.045	1.147
PC	2	62,515	0.026/0.024	1.094	0.051/0.047	1.066
PC	3	62,708	0.027/0.025	1.068	0.051/0.046	1.098
GES	1	308,533	0.028/0.024	1.183	0.056/0.045	1.247
GES	2	308,707	0.028/0.022	1.276	0.055/0.045	1.215
GES	3	307,769	0.029/0.022	1.308	0.056/0.045	1.238
FLOP	1	99,619	0.035/0.024	1.435	0.057/0.046	1.230
FLOP	2	99,562	0.032/0.026	1.238	0.058/0.045	1.301
FLOP	3	99,422	0.032/0.025	1.282	0.060/0.044	1.348
SCOPE		271,066	0.078/0.045	**1.723**	0.107/0.060	**1.785**

Implementation details. For this $n < p$ dataset, we calibrate the graphical lasso regularization parameter λ per dataset via Robust Selection (RobSel). Following prior work, we set RobSel's significance level to 0.99 because RobSel tends to select a conservative λ in this regime (Tran et al., 2020; Bang & Oh, 2023).

Because this setting already yields strong regularization, we use $\widehat{\Omega}(\lambda)$ directly rather than the thresholded $\widehat{\Omega}^{\mathrm{thr}}$ in (8). All other SCOPE, PC, and GES details follow Appendix E. FLOP uses the authors' `flopsearch` implementation with default settings and subsets of approximately 544 features to satisfy its $n > p$ requirement.

Results on additional random partitions. Table 8 compares PC, GES, and FLOP across random feature partitions against full-feature SCOPE; partition 1 matches Table 4. The split-based results vary modestly across partitions but retain lower forward-to-reverse ratios than full-feature SCOPE for both one- and two-step paths.