SCALING TEMPORAL CAUSAL DISCOVERY TO MILLIONS OF VARIABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal causal discovery becomes prohibitively expensive as the number of variables grows: with N series and L lags, a method may need to consider N^2L candidate relationships. Yet many real temporal systems are sparse, with each variable depending directly on only a small fraction of the system. We exploit this structure with VL-PCMCI, a scalable causal discovery framework that separates cheap global screening from expensive causal reasoning. VL-PCMCI first screens all lagged pairs using GPU-accelerated matrix operations, performs causal discovery independently within sparse blocks, and then reconciles relationships across blocks. We characterize when this decomposition preserves causal decisions and identify three requirements: true causes must survive screening, tested non-edges must be given separating sets, and statistical testing must account for screening. Experiments show that VL-PCMCI maintains high accuracy while scaling substantially beyond existing temporal causal discovery methods, recovering nearly every true edge in planted systems with millions of variables. At this scale, computation is no longer the only bottleneck: false positives from data-dependent screening grow approximately quadratically unless the final tests account for selection. These results show that temporal causal discovery at millions of variables is computationally feasible while identifying the statistical conditions required for reliable discovery at this scale.

1 INTRODUCTION

Modern datasets can contain hundreds of thousands or millions of interacting time series, while temporal causal discovery methods remain designed for much smaller systems. With N series and L lags, constraint-based discovery may consider N^2L candidate lagged relationships and use conditional-independence (CI) tests to distinguish direct influence from autocorrelation and indirect paths. Even when the true causal graph is sparse, finding a few parents among NL candidates for every target remains expensive. In the experiments reported here, for example, PCMCI (Runge et al., 2019) did not finish within a day at 2,000 series and 30 lags.

Sparsity suggests screening and decomposition, but applying them naively can change the causal problem. Marginal screening can discard a true parent when direct and indirect effects cancel. Partitioning variables can remove a variable needed to separate a spurious relationship. Moreover, screening and testing on the same data selects unusually correlated pairs, invalidating a test level that ignores the screening stage. Existing work addresses screening, restricted search, and divide-and-conquer causal discovery (Fan & Lv, 2008; Bühlmann et al., 2010; Gu & Zhou, 2020; Shah et al., 2025), but these issues become particularly consequential when the candidate space reaches trillions of lagged relationships.

This paper introduces VL-PCMCI, a temporal causal discovery framework that separates dense candidate search from sparse causal reasoning (Figure 1). It first screens all N^2L lagged pairs using GPU-efficient matrix operations, then runs PCMCI independently within sparse blocks, and finally reconciles candidate relationships across blocks using the discovered parents. Expensive CI testing is therefore concentrated on a sparse set of candidates rather than the full lagged search space.

This decomposition is reliable only when it preserves the information required for causal decisions. The paper identifies three requirements: (i) true causes survive screening, (ii) tested non-edges

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.

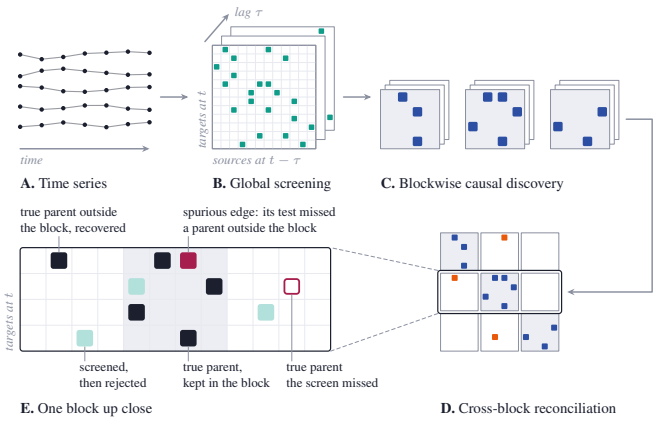

Figure 1: VL-PCMCI in three stages. Global screening retains promising lagged candidates; blockwise causal discovery runs PCMCI independently within blocks; cross-block reconciliation tests candidates across blocks using the discovered parents. Panel E illustrates representative correct and incorrect decisions.

have sufficient separating sets, and (iii) final statistical tests account for the selection induced by screening. Section 4 characterizes these requirements theoretically and constructs counterexamples showing the distinct failure caused by violating each one.

Empirically, VL-PCMCI extends temporal causal discovery from thousands to millions of jointly analyzed series. On same-hardware comparisons, it maintains high accuracy while scaling substantially beyond PCMCI and other baselines; on planted systems with up to two million series and 30 lags, it recovers 99.995% of true edges. At this scale, however, computation ceases to be the only bottleneck: chance survivors of data-dependent screening grow approximately as N^2, making selection-aware testing necessary for statistical scalability.

The contributions are threefold:

1. **Scalable temporal causal discovery.** A screen–discover–reconcile architecture that moves dense all-pairs computation to GPU-efficient screening while restricting expensive CI testing to sparse candidates.
2. **Conditions for reliable decomposition.** Screening recall, separating sets, and selection-aware testing, characterized as the key requirements, with theoretical results and counterexamples identifying their failure modes.
3. **Evaluation from hundreds to millions of variables.** Computational scaling, statistical failure modes, and real temporal systems, including planted graphs with up to two million series.

2 Temporal Causal Discovery at Scale

The data are N time series observed at T time points; X_t^j is series j at time t. A lagged edge $X_{t-\tau}^i \to X_t^j$ states that series i at lag τ directly influences series j, given the rest of the past. The target is the set of lagged edges up to a horizon L no shorter than the longest true lag,

$$E^\star = \{(i, j, \tau) : X_{t-\tau}^i \to X_t^j,\ 1 \le \tau \le L\}, \tag{1}$$

the lagged part of the time-series graph of Runge et al. (2019); a series' own past counts as a parent (an autodependency).

Identifiability setting. The analysis works in the standard PCMCI setting: the process is stationary, causal relations occur at positive lags, relevant common causes are observed, and the distribution is faithful to the time-series graph. Section 4 states these conditions formally (Assumption 1) and distinguishes them from the additional requirements introduced by the decomposition of VL-PCMCI.

3 VL-PCMCI: SCREEN, DISCOVER, RECONCILE

VL-PCMCI keeps PCMCI's MCI test (Runge et al., 2019), a partial-correlation test of a lagged pair given estimated parents of both ends, and changes how the pairs worth testing and their conditions are found. PCMCI treats all NL lagged sources as potential parents of each target, so its work grows with the N^2L candidate edges (Section 2); VL-PCMCI replaces this search with three stages (Algorithm 1; panels B–D of Figure 1).

3.1 GLOBAL SCREENING

In a sparse system almost every lagged pair is unrelated, and a true parent usually leaves a trace in its lagged correlation with the target. VL-PCMCI therefore screens first. It scores all N^2L candidate edges exactly, as tiled dense matrix products of the standardized series on GPUs, and stores only the survivors, as sparse rows (6.2 million of 3×10^{13} at 10^6 series). Exactness matters: an approximate nearest-neighbor screen lost almost all true parents in an earlier large-scale run (Section C.7), and no later stage can add an edge the screen dropped (screening recall, Section 4).

3.2 BLOCKWISE CAUSAL DISCOVERY

After screening, VL-PCMCI partitions the variables into blocks and runs PCMCI independently within each block, with a Benjamini–Hochberg (BH) decision over the block (Benjamini & Hochberg, 1995). Only screened candidates whose source and target belong to the same block are considered at this stage; cross-block candidates are deferred to Stage 3.

In PCMCI every target's PC step draws its conditions from all its lagged candidates, in one global search. Blocking limits both the candidate set and the conditioning pool seen by each PCMCI instance. For a fixed block size, the expensive local discovery step therefore depends on the block size rather than on the total number of variables, and blocks can be processed independently, in parallel and with per-block memory.

The partition schedules computation rather than imposing a causal boundary: cross-block candidates are retained for global reconciliation. Partitioning strategies differ by scale, with random blocks of 5,000 series in the scaling study; implementation details are given in Section A.

3.3 CROSS-BLOCK RECONCILIATION

A candidate edge between two blocks needs the target's parents as conditions, and the within-block runs have already estimated most of them. The *cross-block refinement* therefore tests each such candidate given the parents already estimated for its target and its source (Equation (2) in Section A), with one BH family per pass. The first pass knows only the within-block parents; the second adds the cross-block parents accepted in the first, so, when parents out-rank competing conditions under the conditioning cap, its conditioning set holds every parent of the target. No BH family, per block or per pass, counts the N^2L candidate edges the screen examined; Section 4 shows why that matters. The refinement does not re-test within-block decisions, which leaves the one gap that Section 4 analyzes (Figure 2(b)).

Alternative reconciliation. For smaller systems, block merging, which is more expensive but can revisit within-block decisions, is also considered; Section A gives the procedure.

3.4 SCALING ANALYSIS

Why does this decomposition scale? The key distinction is between operations that are cheap but dense and operations that are expensive but sparse. Screening still examines all N^2L lagged pairs, but it does so through dense matrix multiplication, which maps efficiently to GPUs. Conditional-independence testing is much more expensive per candidate, so VL-PCMCI restricts it to the sparse set that survives screening.

PCMCI's MCI step alone runs $N \cdot NL$ tests (Proposition 2 in Section B), and its PC step's work depends on the settings that bound the search. In VL-PCMCI the screen costs $\Theta(N^2LT)$ arithmetic

Algorithm 1 VL-PCMCI (controls and stage details in Section A)

Inputs: series X; maximum lag L; screen threshold θ; level α
Output: lagged edge set $\hat{E}$

```
Keep every (i, j, τ) with |ρ̂(X^i_{t−τ}, X^j_t)| ≥ θ, scoring all N²L exactly        ▷ Stage 1: global screening
Partition the series into blocks                                                    ▷ Stage 2: blockwise causal discovery
for each block, in parallel do
    Run PCMCI on the screened candidates inside the block; BH over the block at α
for pass 1, 2 do                                                                    ▷ Stage 3: cross-block reconciliation
    Test each screened cross-block candidate given the estimated parents of both ends
    Accept by one BH at α over all cross-block candidates (replaces the last pass)
return every candidate edge accepted by its final test
```

but no CI tests, the within-block work is $O(N)$ CI tests for a fixed block size, and each cross-block pass tests the true parents and related non-edges, $O(N)$ of them, plus about N^2Lq chance survivors, where q is the chance that the screen keeps a candidate edge between two unrelated series. Memory follows the candidates rather than N^2L. At 10^6 series, 200 blocks ran 141 million tests and the two cross-block passes 9.2 million.

VL-PCMCI does not eliminate quadratic work; it moves the unavoidable all-pairs computation into a GPU-efficient screening operation while keeping expensive CI testing approximately linear throughout the practically relevant sparse regime. Asymptotically the CI work also turns quadratic once the chance survivors per target, NLq, rival the true and related candidates; the observed near-linear runtime is a property of the evaluated range (Proposition 2). Section 5.2 measures these costs on the same hardware.

4 When is decomposition reliable?

Scaling causal discovery by decomposition is useful only if decomposition preserves the causal decisions of the original problem. Each stage of VL-PCMCI removes information: screening removes candidate edges, blocking removes potential conditioning variables, and refinement tests hypotheses selected using the same data. This section therefore asks what information each stage must preserve. Proofs are in Section B.

Assumption 1 (Structural time-series model)**.** The series follow a stationary structural model $X_t^j = f_j(\mathcal{P}_j, \varepsilon_t^j)$ with finite second moments, where the parents $\mathcal{P}_j$ are lagged values $X_{t-\tau}^i$ with $1 \le \tau \le L$ (own lags included), the functions f_j are arbitrary, and the noise terms are mutually independent and i.i.d. over time. There are no hidden series and no contemporaneous effects. The distribution satisfies the causal Markov condition and faithfulness with respect to the time-series graph, whose lagged edges form $E^\star$.

This is the setting of PCMCI (Runge et al., 2019); linearity enters only through the screen and the partial-correlation tests, for which a stable linear VAR suffices (Lemma 1).

4.1 Three requirements

In the cross-block refinement, every candidate edge that passes the screen gets one final test, which alone decides whether the edge is kept; block merging tests only candidates whose two blocks it merges. A correct decomposition must then do three things. *Preserve causes*: every true edge passes the screen and reaches a final test. *Recover separators*: every non-edge that reaches a final test is tested given a set that d-separates it from its target in the time-series graph; the target's parents always do (Lemma 2). *Correct for selection*: the final level accounts for all N^2L candidate edges the screen examined, not only those it kept. Formally, these correspond to **screening recall**, valid **separating sets**, and **a testing level that accounts for screening**.

Failure mode analysis. Sparsity alone does not make screen-then-test correct. Figure 2 shows stable linear VARs of at most five series in which exactly one requirement fails, from cancelling paths that hide a parent from the screen to a persistent driver outside a block that leaves a within-block non-edge unseparated; Section B.3 gives all cases with exact values (Table 2).

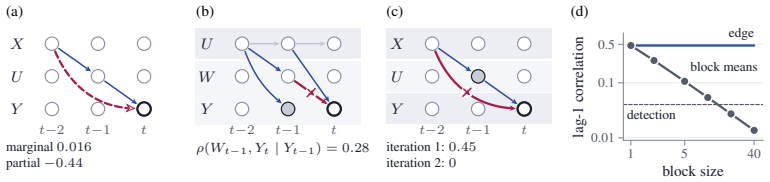

Figure 2: Minimal failure modes of decomposition. (a) Marginal screening misses a true parent because direct and indirect effects cancel. (b) Blocking removes a variable needed to separate a non-edge. (c) One reconciliation pass leaves an indirect relationship unresolved. (d) Block averaging can hide a cross-block signal.

4.2 Correctness on screenable systems

Theorem 1 (Oracle characterization of decomposition errors; proof in Section B.1)**.** *Under Assumption 1 and with a perfect conditional-independence test, every false negative of VL-PCMCI is a true edge that never reached a final test (screening recall failed or, in block merging, its blocks were never merged), and every false positive is a tested non-edge whose final conditioning set does not d-separate it. Screening recall and separating sets thus suffice for* $\hat{E} = E^\star$.

The theorem holds for both reconciliation options and makes errors diagnosable. It reduces correctness to two structural questions: will screening retain every true edge, and will reconciliation recover the conditioning information needed to reject non-edges? The next definition gives sufficient conditions under which both hold.

Definition 1 (Screenability)**.** Let $\rho(e)$ be the population lagged correlation of a candidate edge e and $\rho_{ii}(h)$ the autocorrelation of series i at lag h. A lagged graph taken together with its edge coefficients and noise variances (a weighted lagged graph; more generally, the model of Assumption 1, which fixes these population correlations) is *screenable* at conditioning cap c and threshold θ if (a) *sparse in-degree*: $|\mathcal{P}_j| \le c$ for every j, own lags included; (b) *screening margin*: $|\rho(e)| \ge \theta + \gamma$ for every $e \in E^\star$ and some $\gamma > 0$; and (c) *rank margin*: for every target j, each parent's $|\rho|$ exceeds that of every non-parent of j that passes the screen by some $\omega > 0$.

The screenability conditions (a)–(c) concern the population and depend on the weights as well as the arrows: the arrows of Hidden stitch are screenable unless the paths cancel. A separate, finite-sample condition is *bounded persistence*: $\kappa_{ij} = \sum_h \rho_{ii}(h)\rho_{jj}(h) \le \bar{\kappa}$ for every pair i, j. It plays no part in the population result below; persistence matters only with finite data, because it inflates the screen's false-positive rate q.

Corollary 1 (Exactness on screenable graphs; proof in Section B.2)**.** *Under Assumption 1, with a perfect conditional-independence test and the screen applied to population correlations, let the weighted lagged graph be screenable (Definition 1). Then the cross-block refinement applied twice meets screening recall and gives every cross-block candidate a separating set. Followed by a confirmation re-test of each accepted edge given the target's other estimated parents, it returns* $\hat{E} = E^\star$. *Condition (b) cannot be dropped; (a) and (c) are needed for the two passes without the re-test.*

On screenable sparse systems, the refinement thus meets the first two requirements. The confirmation re-test is proposed, not run; without it, the refinement can err only in within-block decisions of targets with a parent outside their block (Corollary 2). Of the stress-test families (Figure 5), Patchwork is screenable and weakly persistent, Hidden stitch breaks (b) as in Figure 2(a), Through-stitch keeps (a)–(c) but places a persistent driver outside its children's block as in (b), and Bleed breaks bounded persistence.

4.3 Selection error from screening

Even when decomposition supplies the correct separating set, statistical validity can fail because the same data are used to select unusually correlated pairs and test them. At a fixed screening threshold, approximately N^2Lq independent pairs survive screening by chance, where q is the chance that the screen keeps a candidate edge between two unrelated series. Final testing does not remove them: a final test on the same data sees largely the same chance correlation, and conditioning on the parents

does not help, because these pairs are selected, not confounded. False edges therefore number about N^2Lq plus a term linear in N (Proposition 5 in Section B.5). The law's two inputs, q and the pass rate of dependent non-edges, come from independent simulations, with nothing fitted to the scaling study, which tests it (Table 3; Section 5.3).

A finite-sample bound (Theorem 2 in Section B.4), for linear VARs with uniformly concentrating partial correlations, shows that accounting for the adaptive choice of tests raises the signal a true edge needs only logarithmically in NL. This dependence is qualitative rather than numerically tight at the sizes studied here, and the bound covers the refinement followed by a confirmation re-test, not the refinement alone as evaluated in Section 5. At a fixed level this calibration fails as N grows, which is what the false-edge law quantifies. Final tests on data the screen did not see, or a same-data level valid over every conditioning set the algorithm could choose, would restore control (Section B.4); neither is part of VL-PCMCI as evaluated here. In a diagnostic at 10^4 to 10^5 series, confirmation on an independent trajectory at a Bonferroni level removed every false edge (a mean of 15.5, 66 and 429 with reused data) with recall above 0.9999, at the price of twice the data (Section C.7).

4.4 When more variables help

Size also brings a benefit: every variable left out of an analysis is a potential hidden common cause whose children show a false edge that no test on the subset can remove (Section B.6). Consider an idealized rule that tests each candidate edge among fixed targets given all of the target's observed parents.

Proposition 1 (Including more variables; proof sketch in Section B.6)**.** *Let a stable linear VAR over a universe of series satisfy Assumption 1, fix a set S of target series, and observe nested sets $S \subseteq V \subseteq V'$. Under this idealized rule, with population partial correlations, the false edges among S when observing V' are a subset of those when observing V; they vanish once every parent of every target in S is observed, and every true edge among S is accepted.*

This is not the general statement that conditioning on more variables monotonically improves causal discovery; that statement is false. The result relies on adding observed parents under the idealized rule. The implemented rule is not covered, but a controlled study finds the predicted direction for VL-PCMCI: accuracy on 200 fixed targets rose from 0.46 to 0.78 as the analysis grew to 2,040 series (Section 5.5). At a fixed level the screen's chance survivors still grow as N^2 (Section 5).

5 Scaling and reliability

5.1 Experimental setup

Synthetic studies plant a sparse lagged graph and score F1 over non-self edges (Section 2), recall, and the false-discovery proportion (FDP); scale is N (one variable per observed series), T, L, and the N^2L candidate edges, and fit cost is the wall time of screening, blocking, and discovery, not tuning. The baselines are PCMCI (Runge et al., 2019), screened PCMCI (PCMCI on the screened candidates with one Benjamini–Hochberg (BH) family; Benjamini & Hochberg, 1995), and sparse vector autoregression (sparse VAR, node-wise lasso; Basu & Michailidis, 2015), all tuned on development data only (Section C.1).

5.2 Scaling against baselines

The same-hardware scaling study runs every method on one CPU thread with identical memory and time limits, from 100 to 30,000 series of a linear generator (Figure 3). The cross-block refinement finished every size with F1 0.998, its time growing as $N^{1.21}$ (steepening above 10,000 series, where the screen's N^2 arithmetic dominates; Section 3.4). The baselines grew as $N^{1.9}$ to $N^{2.1}$ and stopped at 300 (PCMCI), 1,000 (screened PCMCI), and 3,000 series (sparse VAR), so the refinement was the only method on the CPU accuracy–time frontier at every size; on one GPU it reached 100,000 series in 26 minutes (F1 0.996). At 30 lags the scaling-limit study shows the same gap: VL-PCMCI returned all 80 fits up to 20,000 series (F1 0.9987), while tuned PCMCI and screened PCMCI returned none at 2,000 series within 24 hours.

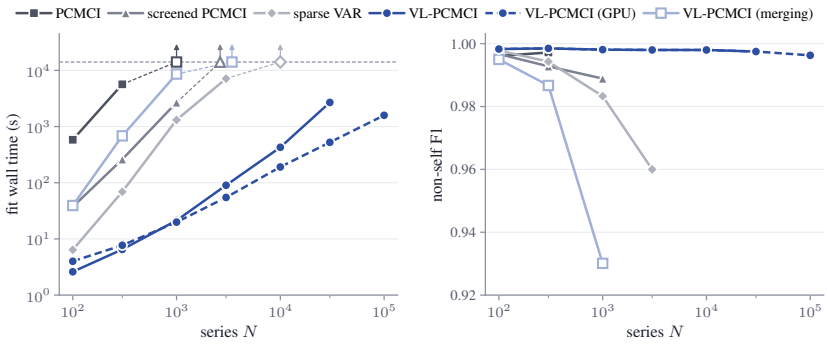

Figure 3: Under identical CPU resources, VL-PCMCI keeps F1 nearly constant while its runtime grows much more slowly than the baselines' over their completed ranges; dashed: VL-PCMCI on one GPU.

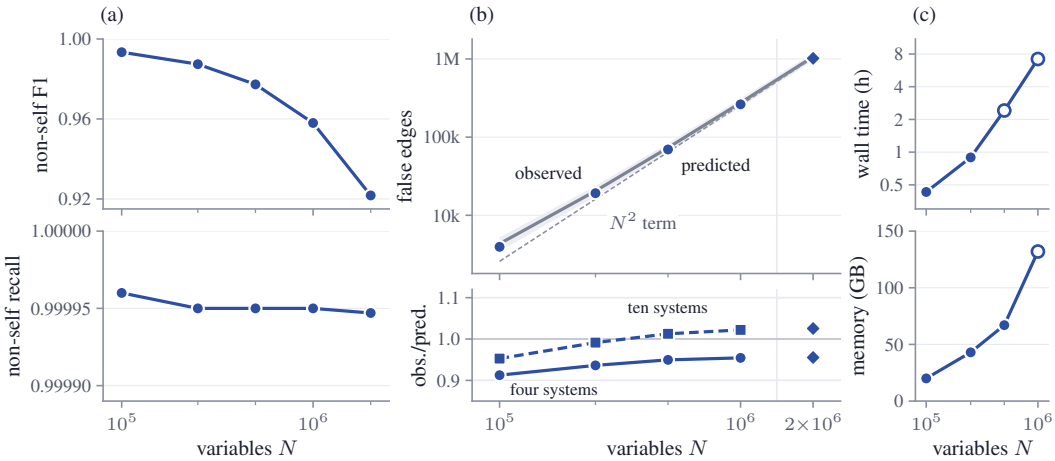

Figure 4: Scaling VL-PCMCI to two million variables. (a) Recall stays nearly constant while F1 falls. (b) False edges (diamond: 2×10^6) against the false-edge law (gray, 95% band) and its N^2 term (dashed); below, observed over predicted for two null-rate estimates. (c) Wall time and memory, secondary (hollow: resumed runs or approximate values).

These results concern linear systems whose true edges clear the screen, with baselines tuned on small systems; the 30-lag study also mixes GPU and CPU, and there sparse VAR, correctly specified, finished 5,000 series with F1 1.0000 (Sections C.5 and C.6).

5.3 The million-variable statistical bottleneck

Section 4 predicts that at a fixed level the screen's chance survivors grow as N^2 while true edges grow as N. The scaling study runs the full pipeline, with no access to the truth, on planted graphs of 10^5 to 2×10^6 series with 30 lags (Section C.2). At one million series recall was 0.99995 and F1 0.958, in 7.18 wall-hours; at two million, recall held at 0.99995 while F1 fell to 0.922 (Figure 4a).

The remaining errors are almost all false edges: their count grew nearly as N^2, to 262,672 at 10^6 and 1,018,316 at 2×10^6, and FDP rose to 0.145. The empirical law of Proposition 5 predicts every count within 10%, and at two million series it predicted 1.07 million false edges where the run produced 1.02 million (Figure 4b; Table 3). The run's own screen confirms the mechanism: pairs with no short connection in the true graph passed it at a constant rate and make up 93.9% of the false edges at 10^6. Recall holds while FDP rises at the rate the law sets, so the test level must account for the screen; a level a hundred times stricter removed only 7.8% of the false edges. The prediction uses independently estimated rates rather than fitting the scaling curve (Section C.7).

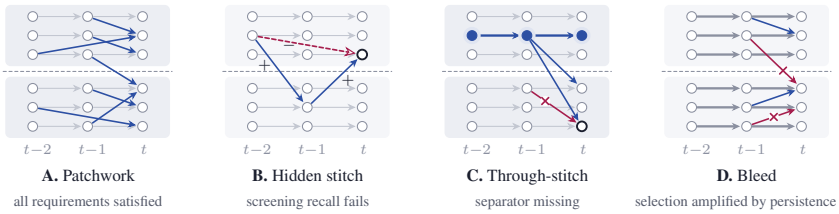

Figure 5: Stress tests of the three requirements: Hidden stitch violates screening recall, Through-stitch removes a separator across blocks, and Bleed amplifies chance screening through persistence; Patchwork is the control.

These runs use one planted linear graph per size whose true parents clear the screen threshold, and no baseline runs at these sizes.

5.4 Stress tests of the three requirements

The previous experiments show that VL-PCMCI can scale; the next question is whether the failure modes predicted by Section 4 explain when it does not. The stress tests construct graph families that selectively violate the requirements (Figure 5): Patchwork is the control, Hidden stitch the screening failure, Through-stitch the separator failure, and Bleed persistence and selection. On Patchwork all methods scored within 0.005 F1; on Hidden stitch no screened method recovered the cancelled parent, while PCMCI and sparse VAR did; on Through-stitch the cross-block refinement kept the driver's spurious edges, while block merging, which re-tests within-block decisions, recovered; on Bleed the screen grew 5.6-fold and the refinement lost 0.02 F1, each as Section 4 predicts (Section C.9).

These small designed systems (64 to 68 series) show mechanisms, not failure rates in practice.

5.5 Benefits of observing more variables

Proposition 1 argues that observing more of a system can remove false edges from hidden common causes. A controlled study places 200 fixed targets among 2,040 series with 40 common drivers, and runs VL-PCMCI on nested subsets of 200 to 2,040 series that reveal 0 to 40 drivers. Target F1 rose from 0.46 to 0.78, monotonically in every seed, at cost linear in N. Recall stayed at 0.998, so the gain is all precision: false edges between children of one driver fell from 261 to 56 per seed, and the remaining 56 are pairs in one block whose driver sits in another, which the cross-block refinement never re-tests.

The drivers are designed, so the gain's size is a design choice, and the benefit is for fixed targets: whole-graph chance false edges still grow with N^2 (Section C.10).

5.6 Real-world systems

Finally, VL-PCMCI is evaluated in two complementary real-world settings. CausalRivers provides partial ground-truth structure and tests recovery accuracy at moderate scale; the financial panel lacks causal ground truth but tests operation on a natural system with over 200,000 time series.

Recovery against known structure. On CausalRivers (Stein et al., 2025), scored against the river network over 30 held-out windows of 32 to 128 stations, tuned VL-PCMCI reached mean F1 0.285 against 0.243 for tuned PCMCI at a tenth of its fit time; sparse VAR tied it (0.286). Absolute accuracy is low for every method, so the comparison is descriptive; Section C.3 reports every evaluation of these windows.

Operational scale. On 204,879 daily series of a global financial market over 4,198 days, VL-PCMCI screened all 1.26×10^{12} lagged pairs exactly and ran 22.3×10^{9} CI tests in about 5.5 days on four to eight GPUs (Section C.4). Its recall-favoring configuration matches no evaluated one and, by Section 4, admits many chance links, so the run shows operational scale only.

6 Related work

Temporal causal discovery. PCMCI and its successors (Runge et al., 2019; Runge, 2020; Gerhardus & Runge, 2020) reduce conditioning sets by selecting candidate parents before final MCI tests. Other approaches use score-based optimization (Pamfil et al., 2020), neural Granger models (Tank et al., 2022; Cheng et al., 2024), or sparse VAR estimation (Basu & Michailidis, 2015; Shojaie & Michailidis, 2010). These methods establish the temporal causal discovery setting in which VL-PCMCI operates, but reported joint analyses remain far below million-variable scale.

Screening and decomposed causal discovery. High-dimensional structure learning has long used screening (Fan & Lv, 2008; Yousuf, 2018; Bühlmann et al., 2010), restricted superstructures (Ha et al., 2016; Nandy et al., 2018), local separation (Sondhi & Shojaie, 2019), and divide-and-conquer strategies (Gu & Zhou, 2020; Shah et al., 2025; Wang & Wan, 2026) to reduce search. These approaches expose two issues central to VL-PCMCI: marginal screening requires true causes to retain detectable marginal signal, while partitioned discovery requires sufficient separating information to remain available across partitions. Testing on the data that selected the candidates also voids cheap multiplicity corrections unless the data are split (Wasserman & Roeder, 2009), the problem behind the third requirement.

Very large-scale causal discovery. Existing methods leave a large gap between temporal causal discovery and causal discovery at very large scale. As summarized in Table 14 (Section D), temporal methods are typically demonstrated on tens to a few thousand variables; larger results either make different assumptions or address i.i.d. structure learning. FGES has been demonstrated at 10^6 i.i.d. variables (Ramsey et al., 2017), GPU implementations accelerate PC-style CI testing (Zarebavani et al., 2020; Hagedorn et al., 2022), and voxel-level analyses reach tens of thousands of temporal variables (Wu et al., 2013). VL-PCMCI targets this gap: to the authors' knowledge, it is the first PCMCI-family lagged discovery pipeline demonstrated at millions of jointly analyzed time series.

7 Discussion and limitations

Once the CI-test workload is controlled, chance candidates introduced by screening become the dominant source of error: computational scalability does not by itself imply statistical scalability. Decomposition trades computation for information, and the three requirements state what must survive the trade; each can fail even in small sparse systems (Section 5.4), and the perspective may extend to other screen-and-decompose approaches to causal discovery.

Increasing dimension has competing effects. More observed variables enlarge the candidate space and therefore increase the opportunity for chance screening errors. At the same time, observing more of the underlying system can reveal common causes that would otherwise remain hidden. In the controlled experiment of Section 5.5, the latter effect substantially improves precision on a fixed set of targets. Thus, for sparse causal systems, dimension is not uniformly detrimental: its effect depends on whether additional variables primarily add irrelevant candidates or reveal missing causal context. Dimension is therefore neither purely a curse nor purely a blessing.

Limitations and open directions. The current implementation uses marginal correlation for screening and partial correlation for CI testing, making nonlinear relationships an important extension. The theoretical guarantees assume the structural conditions of Section 4, including causal sufficiency over the analyzed system and no contemporaneous effects, and the error attribution assumes a perfect CI test. The largest-scale experiments use planted linear systems, with one run each at 10^6 and 2×10^6 series, and beyond 30,000 series their speed comparisons set GPUs against CPUs; the financial experiment demonstrates operational scale rather than causal validity. Finally, the pipeline as evaluated reuses data for screening and final testing; the resulting selection effect explains the observed N^2 growth in false positives. Split-sample or otherwise selection-aware confirmation, which removed every false edge in a diagnostic at 10^4 to 10^5 series (Section 4), is therefore a particularly important next step toward reliable causal discovery at million-variable scale.

AI USE STATEMENT

The research ideas and design of this work originated with the authors. Generative AI tools were used under the authors' direction to discuss and refine ideas, write code, run experiments, help validate the theoretical framework and develop proofs, design and evaluate experiments, create and modify figures, search and summarize relevant literature, format references, and assist with writing and organization.

We did not use generative AI tools to clean or reformat data sets.

The authors reviewed and edited all AI-assisted work, including text and code, and two authors independently checked the proofs. Every number reported in the paper is linked to the record of the run that produced it. Before submission, a separate AI agent also cross-checked the reported claims and numbers against these records.

The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

Section A gives pseudocode for every stage of VL-PCMCI, and Table 1 lists its settings. Section 2 and Section B state every assumption, and Section B contains complete proofs of Theorem 1 and of the other formal results, together with the numerical checks behind Table 2 and Table 3. Section C.1 records the experiment setup of every study (Tables 4 and 5): data and generators, seeds, sample lengths, development and held-out splits, test levels, resource limits and hardware; Table 6 lists the configurations selected on development data. The CausalRivers benchmark is public (Stein et al., 2025). The global financial market panel is proprietary and cannot be released; only the operational-scale statements in Section 5.6 and the panel description in Section C.4 rest on it.

REFERENCES

Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(7):171–234, 2010. URL https://jmlr.org/papers/v11/aliferis10a.html.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015. doi: 10.1214/15-AOS1315.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001. doi: 10.1214/aos/1013699998.

Richard Bourgon, Robert Gentleman, and Wolfgang Huber. Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences*, 107 (21):9546–9551, 2010. doi: 10.1073/pnas.0914005107.

Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer, New York, NY, second edition, 1991. doi: 10.1007/978-1-4419-0320-4.

Peter Bühlmann, Markus Kalisch, and Marloes H. Maathuis. Variable selection in high-dimensional linear models: partially faithful distributions and the PC-simple algorithm. *Biometrika*, 97(2): 261–278, 2010. doi: 10.1093/biomet/asq008.

Ruichu Cai, Zhenjie Zhang, and Zhifeng Hao. SADA: A general framework to support robust causation discovery. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pp. 208–216. PMLR, 2013. URL https://proceedings.mlr.press/v28/cai13.html.

Luca Castri, Sariah Mghames, Marc Hanheide, and Nicola Bellotto. Enhancing causal discovery from robot sensor data in dynamic scenarios. In *Proceedings of the Second Conference on Causal Learning and Reasoning (CLeaR)*, volume 213 of *Proceedings of Machine Learning Research*, pp. 243–258. PMLR, 2023. URL https://proceedings.mlr.press/v213/castri23a.html.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. CUTS+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11525–11533, 2024. doi: 10.1609/aaai.v38i10.29034.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008. doi: 10.1111/j.1467-9868.2008.00674.x.

Mohammad Fesanghary. CEDAR: Causal edge discovery for autoregressive processes. arXiv preprint arXiv:2607.20696, 2026. URL https://arxiv.org/abs/2607.20696.

Mohammad Fesanghary and Abhinav Havaldar. GRACE: Gated refinement for accurate causal edge discovery in high-dimensional time series. arXiv preprint arXiv:2606.23880, 2026. URL https://arxiv.org/abs/2606.23880.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 12615–12625. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/94e70705efae423efda1088614128d0b-Abstract.html.

Jiaying Gu and Qing Zhou. Learning big Gaussian Bayesian networks: Partition, estimation and fusion. *Journal of Machine Learning Research*, 21(158):1–31, 2020. URL https://jmlr.org/papers/v21/19-318.html.

Wiebke Günther, Urmi Ninad, and Jakob Runge. Causal discovery for time series from multiple datasets with latent contexts. In *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 216 of *Proceedings of Machine Learning Research*, pp. 766–776. PMLR, 2023. URL https://proceedings.mlr.press/v216/gunther23a.html.

Min Jin Ha, Wei Sun, and Jichun Xie. PenPC: A two-step approach to estimate the skeletons of high-dimensional directed acyclic graphs. *Biometrics*, 72(1):146–155, 2016. doi: 10.1111/biom.12415.

Christopher Hagedorn, Constantin Lange, Johannes Huegle, and Rainer Schlosser. GPU acceleration for information-theoretic constraint-based causal discovery. In *Proceedings of The KDD'22 Workshop on Causal Discovery*, volume 185 of *Proceedings of Machine Learning Research*, pp. 30–60. PMLR, 2022. URL https://proceedings.mlr.press/v185/hagedorn22a.html.

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(56): 1709–1731, 2010. URL https://jmlr.org/papers/v11/hyvarinen10a.html.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8(22):613–636, 2007. URL https://jmlr.org/papers/v8/kalisch07a.html.

Jonghwan Kim and Sanghack Lee. Don't test what you can deduce: Causal discovery with logical inference. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 337 of *Proceedings of Machine Learning Research*, pp. 2958–2993. PMLR, 2026. URL https://proceedings.mlr.press/v337/kim26a.html.

Steffen L. Lauritzen. *Graphical Models*. Number 17 in Oxford Statistical Science Series. Oxford University Press, Oxford, 1996. doi: 10.1093/oso/9780198522195.001.0001.

Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. Causal information approach to partial conditioning in multivariate data sets. *Computational and Mathematical Methods in Medicine*, 2012:303601, 2012. doi: 10.1155/2012/303601.

Nicolai Meinshausen, Lukas Meier, and Peter Bühlmann. p-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681, 2009. doi: 10.1198/jasa.2009.tm08647.

Ehsan Mokhtarian, Sepehr Elahi, Sina Akbari, and Negar Kiyavash. Recursive causal discovery. *Journal of Machine Learning Research*, 26(61):1–65, 2025. URL https://jmlr.org/papers/v26/24-0384.html.

Preetam Nandy, Alain Hauser, and Marloes H. Maathuis. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183, 2018. doi: 10.1214/17-AOS1654.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pp. 849–856. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html.

J. Jake Nichol, Michael Weylandt, G. Matthew Fricke, Melanie E. Moses, Diana Bull, and Laura P. Swiler. Space-time causal discovery in Earth system science: A local stencil learning approach. *Journal of Geophysical Research: Machine Learning and Computation*, 2(3):e2024JH000546, 2025. doi: 10.1029/2024JH000546.

Hui Ouyang, Cheng Chen, and Ke Tang. Divide-and-conquer strategy for large-scale dynamic Bayesian network structure learning. In Zhongzhi Shi, Jim Torresen, and Shengxiang Yang (eds.), *Intelligent Information Processing XII: 13th IFIP TC 12 International Conference (IIP 2024), Proceedings, Part I*, IFIP Advances in Information and Communication Technology, pp. 63–78, Cham, 2024. Springer. doi: 10.1007/978-3-031-57808-3_5.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. DYNOTEARS: Structure learning from time-series data. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1595–1605. PMLR, 2020. URL https://proceedings.mlr.press/v108/pamfil20a.html.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3(2):121–129, 2017. doi: 10.1007/s41060-016-0032-z.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pp. 1388–1397. PMLR, 2020. URL https://proceedings.mlr.press/v124/runge20a.html.

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.

Ashka Shah, Adela DePavia, Nathaniel Hudson, Ian Foster, and Rick Stevens. Causal discovery over high-dimensional structured hypothesis spaces with causal graph partitioning. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=FecsgPCOHk. arXiv:2406.06348.

Ali Shojaie and George Michailidis. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010. doi: 10.1093/bioinformatics/btq377.

Arjun Sondhi and Ali Shojaie. The reduced PC-algorithm: Improved causal structure learning in large random networks. *Journal of Machine Learning Research*, 20(164):1–31, 2019. URL https://jmlr.org/papers/v20/17-601.html.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, second edition, 2000. doi: 10.7551/mitpress/1754.001.0001.

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

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B. Fox. Neural Granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2022. doi: 10.1109/TPAMI.2021.3065601.

Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9:5233, 2019. doi: 10.1038/s41598-019-41695-z.

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. doi: 10.1007/s10994-006-6889-7.

Wenyu Wang and Yaping Wan. Causal discovery for cross-sectional data based on super-structure and divide-and-conquer. arXiv preprint arXiv:2602.03914, 2026. URL https://arxiv.org/abs/2602.03914.

Larry Wasserman and Kathryn Roeder. High-dimensional variable selection. *The Annals of Statistics*, 37(5A):2178–2201, 2009. doi: 10.1214/08-AOS646.

Sebastian Weichwald, Martin E. Jakobsen, Phillip B. Mogensen, Lasse Petersen, Nikolaj Thams, and Gherardo Varando. Causal structure learning from time series: Large regression coefficients may predict causal links better in practice than small p-values. In *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*, volume 123 of *Proceedings of Machine Learning Research*, pp. 27–36. PMLR, 2020. URL https://proceedings.mlr.press/v123/weichwald20a.html.

Kam Chung Wong, Zifan Li, and Ambuj Tewari. Lasso guarantees for time series estimation under subgaussian tails and β-mixing. arXiv preprint arXiv:1602.04265, 2016. URL https://arxiv.org/abs/1602.04265.

Guo-Rong Wu, Sebastiano Stramaglia, Huafu Chen, Wei Liao, and Daniele Marinazzo. Mapping the voxel-wise effective connectome in resting state fMRI. *PLoS ONE*, 8(9):e73670, 2013. doi: 10.1371/journal.pone.0073670.

Xianchao Xie and Zhi Geng. A recursive method for structural learning of directed acyclic graphs. *Journal of Machine Learning Research*, 9(14):459–483, 2008. URL https://jmlr.org/papers/v9/xie08a.html.

Kashif Yousuf. Variable screening for high dimensional time series. *Electronic Journal of Statistics*, 12(1):667–702, 2018. doi: 10.1214/18-EJS1402.

Behrooz Zarebavani, Foad Jafarinejad, Matin Hashemi, and Saber Salehkaleybar. cuPC: CUDA-based parallel PC algorithm for causal structure learning on GPU. *IEEE Transactions on Parallel and Distributed Systems*, 31(3):530–542, 2020. doi: 10.1109/TPDS.2019.2939126.

APPENDIX

A Algorithm details and configuration map **15**

B Proofs for Section 4 **18**

B.1 Oracle results 18

B.2 Screenability 20

B.3 Failure mode analysis: all cases 20

B.4 Finite-sample properties 21

B.5 False positives after screening 24

B.6 Including more variables 26

B.7 Open problems and scope 26

C Experimental details **27**

C.1 Experiment setup 27

C.2 Configurations 27

C.3 CausalRivers benchmark 28

C.4 Global financial market panel 28

C.5 Scaling-limit study 28

C.6 Same-hardware scaling study: all cells 29

C.7 Scaling study: levels, replicates, screen audit, and resources 29

C.8 Refinement-options comparison 31

C.9 Stress tests 31

C.10 Including more variables 32

D Extended related work **32**

D.1 Comparison with the closest methods 32

D.2 Relation of prior results to the analysis 32

D.3 Search for large lagged analyses 34

A ALGORITHM DETAILS AND CONFIGURATION MAP

Algorithm 2 adds the controls to the compact Algorithm 1 of Section 3, and Algorithm 3 expands each stage, including the three partition rules and both cross-block options. The notes below describe implementation choices that the theory and the experiments depend on. Table 1 lists the configuration each study ran.

Partitions. The CausalRivers benchmark and block merging in the same-hardware study used spectral clustering (Ng et al., 2001) of the weighted candidate graph into a fixed number of blocks. The scaling-limit study used Leiden communities (Traag et al., 2019) of the screened graph. The scaling study used random blocks of 5,000 variables, because Leiden communities reached 16,000 variables at 250,000 variables and exhausted the memory of the within-block step. How the blocks are formed is a design choice that affects cost, not correctness of the analysis in Section 4: clustering the screened candidate graph keeps most candidate edges inside blocks and so reduces the cross-block work, whereas random balanced blocks are truth-blind and cheap. The partition changes only the cost and the exposure of within-block decisions whose separating parent lies in another block (Figure 2(b)); Theorem 1 holds for every partition.

Passes and option choice. VL-PCMCI applies the cross-block refinement twice. In the scaling study a third pass moved the false-discovery proportion by at most 0.0011, with identical recall and misses. Block merging is used for small panels, where re-testing within-block decisions matters (Figure 2(b)).

Conditioning set of the cross-block refinement. In pass r, the decided set $\mathcal{D}_r(v)$ of a variable v holds its within-block parents $E_2(v)$, the cross-block parents $A_{r-1}(v)$ accepted in pass $r-1$, and, if $\iota = 1$, the own lags of v. Entries are ordered by screening score, with unscreened own lags last. The test of $e = (i, j, \tau)$ conditions on

$$Z_e = \big(\mathcal{D}_r(j)_{[c]} \setminus \{(i,\tau)\}\big) \cup \big(\mathcal{D}_r(i)_{[c]} \text{ with every lag increased by } \tau\big), \tag{2}$$

where $\mathcal{D}_{[c]}$ keeps the first c entries. If $|Z_e| > c$, Z_e is re-ranked by each condition's screening score as a parent of j and cut to c. Candidates are processed in fixed-size chunks of targets, so no variable pool accumulates.

Memory and determinism. The screen runs as tiled dense matrix products; the GPU implementation evaluates CI tests of equal conditioning size as one batch and stores within-block edges compactly. At 250,000 variables the screen took 399 of the run's 3,140 seconds, against 2,379 seconds within blocks and 362 seconds between blocks, and the run peaked at 43 GB of host memory; a reproduction without compact storage, on a different host, reached about 300 GB, which is not a controlled comparison. Memory follows the candidates rather than N^2L: the million-variable run peaked at about 132 GB. Block results are deterministic: one-GPU and eight-GPU runs at 100,000 variables returned identical edge sets. No multi-GPU speedup is claimed.

Block merging. The coarse block graph comes from PCMCI on block-mean series (*centroid*) or from the screen itself, which joins two blocks when at least m candidates in $\mathcal{C}$ connect them (*screened*). For each base block B_a, the neighbors of B_a are merged into a growing pool $M \supseteq B_a$, h blocks at a time. Each merge proposes, for every target in B_a, its current parents inside M and every screened candidate whose source lies in the newly merged blocks. PCMCI then decides these links on M, and the result replaces the parent sets of the targets in B_a. Block merging therefore re-tests within-block edges and can remove them, and it conditions on the whole pool rather than on decided structure only. A candidate whose two blocks are not coarse-graph neighbors is never tested, so recall depends on the coarse graph as well as on the screen. In an earlier large-scale run (250,000 variables), which also used the approximate screen, the centroid graph proposed none of the cross-block merges that the planted edges required.

PC level. In every run, the PC step removes candidates at level 0.05: the implementation does not pass the configured PC level to the batched CI test, which uses its default of 0.05. The configured values α_{pc} in Table 1 were therefore not applied, within blocks or in the optional PC step of block merging. The MCI level α is applied as configured.

Algorithm 2 VL-PCMCI with its controls (overview of Algorithm 1)

Inputs: series $X \in \mathbb{R}^{T \times N}$; maximum lag L; screen threshold θ; partition rule; MCI level α; conditioning cap c; combination budget q; own-lag switch ι; cross-block option with its controls (refinement: passes R; merging: coarse graph, merge width h, BH family)

Output: lagged edge set $\hat{E}$

$\mathcal{S}_{\tau,j} \leftarrow \{i : |\hat{\rho}(X^i_{t-\tau}, X^j_t)| \geq \theta\}$ for all $\tau \leq L$, $j \leq N$ ▷ Stage 1: exact, sparse
Partition the variables into blocks $B_1, \ldots, B_K$
for each block B_k, in parallel **do** ▷ Stage 2
    Run PCMCI on links $(i, \tau) \to j$, $j \in B_k$, $i \in \mathcal{S}_{\tau,j} \cap B_k$ (+ own lags if $\iota = 1$); BH over the block
    $E_2(j) \leftarrow$ accepted parents of j, for $j \in B_k$
$\mathcal{C} \leftarrow \{(i, j, \tau) : i \in \mathcal{S}_{\tau,j},\ b(i) \neq b(j)\}$ ▷ Stage 3
if option is the cross-block refinement **then**
    for $r = 1, \ldots, R$ **do**
        MCI-test every $e \in \mathcal{C}$ given Z_e from Equation (2)
        $A_r \leftarrow$ links accepted by one BH over $\mathcal{C}$ ▷ replaces A_{r-1}
    return $\hat{E} = E_2 \cup A_R$
else ▷ merge
    for each base block B_a, starting from $M \leftarrow B_a$ **do**
        For each group G of h coarse-graph neighbors: $M \leftarrow M \cup G$; rerun PCMCI on M for links into B_a
        Replace the parent sets of targets in B_a with the latest decisions
    return union of base-target parent sets (BH per merge, or one pooled BH)

Within-block decisions. The cross-block refinement never re-tests within-block edges, so a within-block edge is decided by its within-block MCI test alone. For a target with a cross-block parent, that test does not condition on the cross-block parent, so a within-block non-edge of such a target can be tested without a separating set. Dependent within-block non-edges fall in this gap (Corollary 2(b)) and can carry residual partial correlations up to about 0.09, but they are few because, under random blocks, few dependent pairs share a block (Section 5). A confirmation re-test of each within-block edge in the last pass would close the gap; it was not run.

Conditioning cap. VL-PCMCI passes the cap c as the bound on the PC conditioning dimension and on the MCI conditioning set. The reference PCMCI used for comparison instead splits the MCI budget between target and source parents, $\lceil c/2 \rceil$ and $\lfloor c/2 \rfloor$. The two methods therefore share c but not the exact conditioning rule.

Source cap. The joint re-rank in Equation (2) scores every condition as a parent of the target j. A source-side condition with a high score can therefore evict a true parent of j, and the source's own lags, which are unscored as parents of j, are evicted first. A separate source cap c_x keeps the target's first c conditions and adds at most c_x source conditions without a joint cut, which avoids both effects. Tested as a correction, it had no measurable effect: 19,055 against 19,177 false positives at 250,000 variables, and F1 0.286 against 0.294 on the development windows of the CausalRivers benchmark. The reported runs use the joint cap.

Own lags. The GPU implementation (the scaling-limit and scaling studies) runs with $\iota = 1$: each target's own lags $1, \ldots, L$ are tested within blocks and enter its decided set in the cross-block refinement. The CPU implementation (the small-panel studies) runs with $\iota = 0$, so an own lag enters only if the screen keeps it.

Pooled family. Before the pooled family re-decides any link, the implementation verifies that the per-family adjusted p-values reproduce its edge set, so that only the family changes. At equal α, many small families tend to admit more false positives than one pooled family.

Algorithm 3 VL-PCMCI, stage by stage

procedure SCREEN(X, L, θ) ▷ Stage 1
  for $\tau = 1, \ldots, L$ **do**
    for each tile of sources $\times$ tile of targets **do**
      Compute $|\hat{\rho}(X^i_{t-\tau}, X^j_t)|$ for every pair in the tile
      Append each i with score $\geq \theta$ to $\mathcal{S}_{\tau,j}$ and its score to $\mathcal{W}_{\tau,j}$
  return $(\mathcal{S}, \mathcal{W})$ as per-lag compressed sparse rows

procedure PARTITION($\mathcal{S}, \mathcal{W}$, rule)
  $A_{ij} \leftarrow$ largest retained score of $i \to j$ over lags ▷ candidate graph
  if rule is spectral(K) **then**
    Spectral clustering of A into K blocks ▷ dense; small N only
  else if rule is screened Leiden **then**
    Leiden communities of A, resolution 1
  else ▷ random(s)
    $K \leftarrow N/s$; label $\pi(v) \bmod K$ for a random permutation π

procedure LOCALPCMCI($X, B_k, \mathcal{S}, \iota$) ▷ Stage 2, one block
  Tested links: $(i, \tau) \to j$ for $j \in B_k$, $i \in \mathcal{S}_{\tau,j} \cap B_k$ (+ $(j, 1..L)$ if $\iota = 1$)
  PC step at level 0.05; conditioning dimension $\leq c$, at most q combinations per dimension
  MCI step on *every* tested link; at most c conditions
  BH over the block's MCI p-values; $E_2(j) \leftarrow$ links into j accepted at α

procedure REFINE($\mathcal{C}, E_2, R, \iota$) ▷ Stage 3, cross-block refinement
  $A_0 \leftarrow \emptyset$
  for $r = 1, \ldots, R$ **do**
    $\mathcal{D}_r(v) \leftarrow E_2(v) \cup A_{r-1}(v)$ (+ own lags if $\iota = 1$), ordered by Stage-1 score
    for each chunk of targets **do**
      MCI-test each $e \in \mathcal{C}$ in the chunk given Z_e in Equation (2)
    $A_r \leftarrow$ links whose BH-adjusted p-value over all of $\mathcal{C}$ is $\leq \alpha$
  return A_R ▷ last pass only

procedure MERGE($\{B_k\}, E_2$, coarse rule, h, family) ▷ Stage 3, block merging
  Coarse graph: centroid (PCMCI on $\bar{X}^{B_k}_t = |B_k|^{-1} \sum_{v \in B_k} X^v_t$; neighbors share a coarse edge) or screened (neighbors share $\geq m$ candidates)
  for each base block B_a **do**
    $M \leftarrow B_a$; $P(j) \leftarrow E_2(j)$ for $j \in B_a$
    for each group G of h neighbors of B_a, most connected first **do**
      $M \leftarrow M \cup G$
      Propose for $j \in B_a$: $P(j)$ and $\{(i, \tau) : i \in \mathcal{S}_{\tau,j} \cap G\}$
      Run PCMCI on M for the proposed links (PC step optional, at 0.05); BH per merge
      $P(j) \leftarrow$ accepted links, for $j \in B_a$
  if family is pooled **then**
    Re-decide every tested link by one BH over the raw p-values of its final MCI test
  return $\bigcup_a \{(i, j, \tau) : j \in B_a,\ (i, \tau) \in P(j)\}$

Table 1: Configuration of VL-PCMCI in each study (CausalRivers: the CausalRivers benchmark). "Default" is the default configuration of every tuning study; other rows are the selected configurations. θ: screen threshold; α: MCI level; α_{pc}: configured PC level, not applied (every run removes at 0.05; see the PC-level note); c: conditioning cap; q: combinations per conditioning dimension; h: blocks merged per step; R: passes of the cross-block refinement.

Study	Hardware	Blocks	Cross-block	BH family	Controls
Default	CPU	spectral, K=4	merging, centroid, no PC	per block, merge	θ=0.1, α=0.01, α_{pc}=0.05, c=6, q=50
Same hardware, merging	CPU	spectral, K=4	merging, centroid, PC	per block, merge	θ=0.05, α=10^{-4}, α_{pc}=0.01, c=6, h=2
CausalRivers	CPU	spectral	merging, centroid, PC	per block, merge	q=1; other selected values not available
Scaling-limit	GPU	screened Leiden	refinement, R=2	per block, per pass	θ=0.1, α=0.01, α_{pc}=0.05, c=6, q=50
Scaling	GPU	random, 5,000	refinement, R=2	per block, per pass	θ=0.1, α=0.01, α_{pc}=0.05, c=6, q=50

B Proofs for Section 4

Notation. $\mathcal{G}$ denotes the time-series graph with nodes X_t^i ($i \le N, t \in \mathbb{Z}$) and an edge $X_{t-\tau}^i \to X_t^j$ whenever $X_{t-\tau}^i \in \mathcal{P}_j$, and $\mathrm{Past}_t = \{X_{t-u}^i : u \ge 1\}$. A test *reports dependence* when its p-value is at or below the decision threshold of its family. For a candidate edge $e = (i, j, \tau)$, Z_e is its final conditioning set, and C^{rt} is the set of candidate edges that reach a final test (Section 4.1). The algorithmic symbols follow Section A: $\mathcal{S}_j$ is the set of screened candidates of target j, $b(j)$ its block, c the conditioning cap, R the number of passes of the cross-block refinement, $\mathcal{D}_r(j)$ the decided set of j in pass r, and ι the own-lag switch. *Protection* holds for j if every true parent of j in $\mathcal{D}_r(j)$ survives the cap c and the re-ranking of Equation (2).

B.1 Oracle results

Lemma 1 (Markov property of a stable linear VAR for partial correlations)**.** *Let the process be a stable linear VAR with independent innovations of positive variance, and let Z d-separate nodes x, y in $\mathcal{G}$, with $\{x, y\} \cup Z$ finite. Then $\rho(x, y \mid Z) = 0$. With Gaussian innovations, x and y are conditionally independent given Z.*

Proof. Let $V = \{x, y\} \cup Z$, and let t_0 precede every time index in V. The process $X^{(t_0)}$ started at zero at time t_0 and driven by the same innovations is a finite linear structural model with independent errors on the subgraph $\mathcal{G}_{t_0}$ of nodes at times $\ge t_0$. Every path of $\mathcal{G}_{t_0}$ is a path of $\mathcal{G}$, and descendant sets coincide in the two graphs because descendants lie in the future; hence d-separation in $\mathcal{G}$ implies d-separation in $\mathcal{G}_{t_0}$. The covariance of a linear structural model with independent errors equals that of the Gaussian model with the same coefficients and variances, which is globally Markov with respect to its DAG (Lauritzen 1996; Spirtes et al. 2000). Therefore $\rho^{(t_0)}(x, y \mid Z) = 0$. By stability, $\mathrm{Cov}(X_V^{(t_0)}) \to \mathrm{Cov}(X_V)$ as $t_0 \to -\infty$, and the limit is positive definite because the spectral density of X is bounded below when all innovation variances are positive (Basu & Michailidis, 2015, Prop. 2.3). Partial correlations are continuous on positive-definite matrices, so $\rho(x, y \mid Z) = 0$. □

Lemma 2 (Conditioning on the parents; Runge et al., 2019, Supp. Lemma 2; Fesanghary, 2026, Prop. 3)**.** *Under Assumption 1, let $Z \subseteq \mathrm{Past}_t$ with $\mathcal{P}_j \subseteq Z$, and let $X_{t-\tau}^i \notin \mathcal{P}_j \cup Z$. Then Z d-separates $X_{t-\tau}^i$ and X_t^j in $\mathcal{G}$; hence $X_{t-\tau}^i$ and X_t^j are conditionally independent given Z, and, for a stable linear VAR, $\rho(X_{t-\tau}^i, X_t^j \mid Z) = 0$.*

Proof. Let $x = X_{t-\tau}^i$ and consider a path between X_t^j and x. If its first edge points into X_t^j, the adjacent node is a parent $p \in Z$ and a non-collider on the path, so the path is blocked. Otherwise the first edge is $X_t^j \to w$ with w later than t; since x precedes t, the path contains a collider d that is a descendant of X_t^j. All descendants of d lie after t, so none is in Z, and the path is blocked at d. Hence Z d-separates x and X_t^j. The Markov condition of Assumption 1 gives conditional independence, and Lemma 1 gives $\rho = 0$ for a stable linear VAR. □

The condition $\mathcal{P}_j \subseteq Z$ cannot be replaced by time order alone. In the past-collider system of Table 2, row 5, $\rho(X_{t-2}, Y_t \mid C_{t-1}) = -1/\sqrt{10}$, whereas $\rho(X_{t-2}, Y_t \mid C_{t-1}, U_{t-2}) = 0$.

Proof of Theorem 1. An edge enters $\hat{E}$ only if its final test ran and reported dependence, so $\hat{E} \subseteq C^{\mathrm{rt}}$. Let $e = (i, j, \tau)$ be a tested true edge. Its final conditioning set excludes $X^i_{t-\tau}$, since the target conditions exclude e and the shifted source conditions lie at lags beyond τ. The two nodes are adjacent in $\mathcal{G}$, so no set d-separates them, and by faithfulness the oracle reports dependence. Hence $e \in \hat{E}$ and $\mathrm{FN} = E^\star \setminus C^{\mathrm{rt}}$. Let e be a tested non-edge whose final conditioning set d-separates it from its target. By the Markov condition the pair is conditionally independent given that set, and the oracle reports independence; hence every false positive has a non-separating final conditioning set. Conversely, by faithfulness a non-separating set gives conditional dependence, and the non-edge is accepted. Each decision depends only on the candidate's own final conditioning set, irrespective of how earlier decisions shaped it. □

Remark 1 (Faithfulness and discrete coefficients). Faithfulness is assumed, not derived. The scaling study draws its coefficients from a discrete set, so generic-coefficient arguments do not apply and exact path cancellations can occur; they affect only the converse in the proof of Theorem 1.

Remark 2 (Protection and the own-lag switch). In the cross-block refinement, $\mathcal{D}_r(j)$ is ordered by screening score, and unscored own lags are appended last when $\iota = 1$. Protection holds if $|\mathcal{D}_r(j)| \leq c$ and the joint re-ranking of Equation (2) does not bind (or a separate source cap is used), or if every true parent out-scores every non-parent in $\mathcal{D}_r(j)$ and every shifted source condition, and $|\mathcal{P}_j| \leq c$. Whether the scaling study meets these conditions is an empirical question and not part of Corollary 2. There the screen keeps autodependencies at lags ≥ 1, and the true own lag 1 ($|\rho| \approx 0.3$) ranks first. For cross-block parents, protection fails when at least three scored non-parents outrank the weakest true parent (population $|\rho| \approx 0.15$, sampling standard deviation 0.017, against spurious scores of 0.10–0.12), which is rare but possible. With $\iota = 1$, own lags are within-block candidates of every target; a true autodependency is accepted within the block by faithfulness and competes by score, while the unscored own lags cut at c are non-parents. With $\iota = 0$, an autodependency enters only if the screen keeps it; if it is missed, it is a false negative, every other candidate of j loses its separating set through the parents, and the target residual stays autocorrelated.

Corollary 2 (The two cross-block options)**.** *Assume Assumption 1, an oracle test, screening recall, and protection. (a) Applied twice ($R \geq 2$), the cross-block refinement gives every cross-block non-edge a separating set and accepts exactly the true cross-block edges, and every later pass returns the same set. With $R = 1$ this can fail. (b) The refinement never re-tests within-block decisions, so a within-block non-edge whose target has a parent outside its block may lack a separating set. One further MCI test of each decided edge e given $\mathcal{D}_{R+1}(j) \setminus \{e\}$ (the confirmation re-test) makes the output exact. (c) In block merging, screening recall additionally requires the block of every cross-block parent to be a coarse-graph neighbor of its target's block; under this routing condition, the last merge gives every tested non-edge a separating set.*

On screenable graphs, protection follows from the rank margin (Section B.2).

Proof. (a) After the within-block step and pass 1, $\mathcal{D}_2(j) \supseteq \mathcal{P}_j$: a true within-block parent is a within-block candidate (screened, or an own lag with $\iota = 1$) whose within-block MCI test reports dependence by faithfulness, and a true cross-block parent is tested in pass 1 and accepted by faithfulness. True cross-block parents are accepted in every pass, so the inclusion persists for all $r \geq 2$. Protection then puts $\mathcal{P}_j \setminus \{e\}$ into every final conditioning set of pass $r \geq 2$; Lemma 2 rejects every cross-block non-edge, and faithfulness accepts every true cross-block edge. For $R = 1$, see Table 2, row 3.

(b) The within-block MCI test conditions on PC-selected parents from the target's block. If $\mathcal{P}_j \subseteq b(j)$, the PC step never removes a true parent (faithfulness), and protection places all of them in Z_e. Otherwise $Z_e \not\supseteq \mathcal{P}_j$, and Table 2, row 2, gives a failure. By (a), $\mathcal{D}_{R+1}(j) \supseteq \mathcal{P}_j$, so Lemma 2 and faithfulness make the confirmed output exactly $\mathcal{P}_j \cap \mathcal{S}_j$.

(c) Every proposal is re-tested at every merge, and faithfulness ensures that true parents are never rejected. A base-block parent enters through the within-block step and is re-proposed at every merge; a cross-block parent is proposed when its block is merged and remains in the pool. Under the routing condition the last merge therefore proposes all of $\mathcal{P}_j$, the PC step keeps them, and protection and

Lemma 2 cover every proposal. The output is the last merge's accepted set. A base block without neighbors has only targets whose parents lie in the block, and the argument of (b) applies. □

Proposition 2 (Cost)**.** *Let every target have at most m candidates, own lags included, and let the PC step try at most $q_{\max}$ subsets per conditioning size. Then VL-PCMCI with the cross-block refinement runs at most $Nm(2 + c\,q_{\max} + R)$ CI tests, each conditioning on at most c variables, whereas the MCI step of PCMCI alone runs $N \cdot NL$ tests.*

Proof. Per target, the within-block PC step tests each within-block candidate at most $1 + c\,q_{\max}$ times (once at size zero and at most $q_{\max}$ times at each size up to c), the within-block MCI step tests it once, and the cross-block refinement tests each cross-block candidate once per pass. Summing over at most m candidates per target and N targets gives the bound. The MCI step of PCMCI tests all $N \cdot NL$ lagged pairs. □

Remark 3 (When the cost is linear in N). At a fixed screen threshold m is not constant: $m \approx k + m_{\mathrm{rel}} + \iota L + NLq$, where k is the number of true parents per target, m_{rel} the number of dependent non-edges the screen keeps per target, and q the screen's false-positive rate (Proposition 5). The CI work is therefore linear in N only while $NLq \ll k + m_{\mathrm{rel}}$. For cross-block candidates in the scaling study ($3 + 1.37$ per target, $q \approx 8 \times 10^{-9}$) this holds below $N^\star \approx 1.8 \times 10^7$; beyond it the selected nulls make the CI work quadratic, as the screen already is.

B.2 Screenability

In Definition 1, $\rho(e)$ is the population value of the screen statistic on a candidate edge e. The statistic may be the lagged sample correlation of the raw series or of each series after filtering by its own fitted autoregression (per-series prewhitening); after prewhitening, the Bartlett factor κ of the bounded-persistence condition is about 1. In the population, the screened non-parents of a target are the dependent ones, carried by autocorrelation. Under the joint re-ranking of Equation (2), the rank margin (c) must also cover the shifted source conditions (Table 2, row 4).

Proof of Corollary 1. With the screen applied to population values, the screening margin (b) gives screening recall, since every true parent scores at least $\theta + \gamma$. Every list ranked for a target j holds screened candidates of j followed by unscored own lags, and the unscored entries are non-parents because by (b) every parent is scored. By the rank margin (c) every parent out-scores every scored non-parent, so the parents occupy the first $|\mathcal{P}_j|$ positions, which lie within the first c by (a). This is protection, and Corollary 2(a) and (b) give the claim: after two passes every cross-block non-edge is tested given a superset of the target's parents, the confirmation re-test does the same for every within-block decision, and Theorem 1 yields $\hat{E} = E^\star$. Condition (b) cannot be dropped: without a screening margin the screen loses the parent (Table 2, row 1a). Conditions (a) and (c) are needed for the two passes without the re-test (row 4: in-degree above the cap at $c = 1$, and a violated rank margin under a binding cap at $c = 2$); an uncapped re-test given all of the target's other estimated parents repairs these cases. □

Remark 4 (Role of bounded persistence). Bounded persistence plays no part in the population statement. It governs the screen's false-positive rate q (Proposition 5), the sampling variability of the rank margin, and the effective sample size of every test.

B.3 Failure mode analysis: all cases

Proposition 3 (Counterexamples)**.** *There exist stable VARs with at most five series and at most two parents per series besides autodependencies with the following properties. (a) Screening recall fails while every tested non-edge has a separating set (near-cancellation: partial correlation -0.44, marginal correlation 0.016). (b) Screening recall holds, and a within-block non-edge lacks a separating set under the cross-block refinement for $R \in \{2, 3\}$ (a driver outside the block). (c) A separating set is missing at $R = 1$ for a chain across three blocks, and at cap $c = 2$ under the joint re-ranking but not under a separate source cap. (d) The screen keeps a parent with correlation 0.48, but a coarse graph on block averages drops it once blocks hold 14 or more series. (e) Every series has one parent, yet the number of candidates per target grows linearly with a hub's out-degree.*

Table 2: Failure modes (Proposition 3; Figure 2 draws four of them): small stable VARs, each isolating one way the pipeline can fail. "Refinement" is the cross-block refinement, "re-test" the confirmation re-test of Corollary 2(b), and "merging" block merging with the screened coarse graph; r is a marginal lagged correlation, $\rho(\cdot \mid \cdot)$ a population partial correlation, and η that of a false edge given its final conditioning set. Numbers in parentheses are edge coefficients.

	What fails	System	Key population values	Outcome
1a	Screening recall (cancelling paths)	$X \to U \to Y$ (0.8, 0.8) and $X_{t-2} \to Y_t$ (-0.62)	$r = 0.016$; $\rho(X_{t-2}, Y_t \mid U_{t-1}) = -0.44$	The screen misses $X_{t-2} \to Y$; no false edge
1b	Same, with persistence	As 1a, own lag 0.3 on every series	$r = 0.016$; partial -0.48	Same miss; false edges $X_{t-3} \to Y$ ($\eta = 0.038$) and $U_{t-2} \to Y$ ($\eta = 0.024$)
2	Separating set inside a block	U (own lag 0.8) in block B drives W and Y in block A (0.8)	$\rho(W_{t-1}, Y_t \mid Y_{t-1}) = 0.279$; 0 given U_{t-1}	Refinement (2 or 3 passes): 8 false edges in block A; with re-test, and merging: exact
3	Separating set across blocks	$X \to U \to Y$ (0.8, 0.8), one series per block	$r = 0.447$; 0 given U_{t-1}	One pass: false edge $X_{t-2} \to Y$; two or three passes: exact
4	Conditioning cap and rank margin	$Y \leftarrow P_1$ (0.9), $Y \leftarrow P_2$ (0.3), $W \to P_2$, $Q \to W$, $Q \to P_1$	Scores for Y: P_1 0.78, Q_{t-3} 0.60, P_2 0.51; $\rho(W_{t-2}, Y \mid P_1, Q_{t-3}) = 0.25$	Cap 1: false edge; cap 2, joint ranking: false edge; cap 2, separate source cap: exact
5	An arbitrary past set	$X_{t-2} \to C_{t-1} \leftarrow U_{t-2} \to Y_t$ (unit)	$r = 0$; $-1/\sqrt{10}$ given C; 0 given C, U	Conditioning on all parents separates (Lemma 2)
6	Merging's coarse graph	Two blocks of n series (own lag 0.3), one edge (0.5) between them	Edge $r = 0.477$; block-mean $r = 0.477$, 0.255, 0.106, 0.054, 0.027, 0.014 for $n = 1, 2, 5, 10, 20, 40$	$n = 5$: the coarse graph omits the pair and merging misses the edge; refinement exact
7	Re-test beyond (a)–(c)	Hub (own lag 0.8) drives n series (0.8)	In-degree 1; 9, 15, 27, 51 candidates per target for $n = 2, 4, 8, 16$	Refinement: not exact; with re-test: exact (cap 2 and 8)
8	Size of the within-block gap	Scaling-study motif: own lag 0.3; parents ± 0.15 at lags 3, 11, 25	Largest spurious own-lag partial given six own lags 4.0×10^{-4} (0.0043 at ± 0.3; 0.016 at ± 0.5); true-parent partial ≥ 0.155	Gap of Corollary 2(b) is 100 times below the detection threshold 0.040

Proof. By exact population computation on a re-implementation of the decision rules (Table 2). All values are population quantities of stationary VARs with unit innovation variances; the lagged covariances $\Gamma(h)$ follow from the companion-form discrete Lyapunov equation and the Yule–Walker recursion. The outcomes apply the decision rules of VL-PCMCI with an oracle test ($|\rho| > 10^{-9}$): the exact screen at $\theta = 0.1$; the stable PC step over all subsets up to size c, then the MCI step over every candidate with the joint re-ranking at cap c; and passes of the cross-block refinement, or block merging with re-testing at every merge and replacement of the parent sets. □

B.4 Finite-sample properties

Throughout this subsection the process is a stable linear VAR and every test is a partial-correlation test. A *query* $\omega = (x, y, Z)$ is a pair of nodes with a conditioning set; ρ_ω and $\hat{\rho}_\omega$ are its population and sample partial correlations.

Assumption 2 (Uniform concentration of fixed partial correlations)**.** There are constants $a_s, b, u_0 > 0$ and an effective sample size T_{eff} such that every fixed query $\omega = (x, y, Z)$ with nodes in a window

of $2L$ lags and $|Z| \le s$ satisfies

$$P\big(|\hat{\rho}_\omega - \rho_\omega| \ge u\big) \le a_s\, e^{-bT_{\text{eff}}u^2} \qquad (0 < u \le u_0).$$

Remark 5 (Sufficient conditions for Assumption 2). Let $W_t = (X_t, \dots, X_{t-2L})$, a stable Gaussian process, and $n = T - 2L$. Write f_W for its spectral density, $\mathfrak{m}(f_W)$ and $\mathfrak{M}(f_W)$ for the extreme eigenvalues of f_W, and $\mathcal{M}(f_W, 2)$ for the corresponding quantity of Basu & Michailidis (2015). Their Prop. 2.4(a) bounds each covariance entry, $P[|\hat{S}_{ab} - \Gamma_{ab}| > 6\pi\mathcal{M}(f_W,2)\xi] \le 6e^{-c_0 n \min\{\xi,\xi^2\}}$; a union bound over the $(s+2)^2$ entries of a query block controls its operator norm; and Lemma 3 transfers the bound to partial correlations, the eigenvalues of the block lying in $[2\pi\mathfrak{m}(f_W), 2\pi\mathfrak{M}(f_W)]$ (Basu & Michailidis, 2015, Prop. 2.3). This yields $a_s = 6(s+2)^2$ and $bT_{\text{eff}} \asymp n\,(s+2)^{-2}\big(\mathfrak{m}(f_W)^2/(\mathfrak{M}(f_W)\mathcal{M}(f_W,2))\big)^2$. The effective sample size shrinks with the spectral condition number and collapses near a unit root; for an AR(1) with coefficient a, $2\pi\mathfrak{M} = (1+a)/(1-a)$. Two steps are not written out: the centering term, and the bound of $\mathcal{M}(f_W,2)$ by $\mathcal{M}(f_X,2)$. The crude bound $\mathfrak{M}(f_W) \le (2L+1)\mathfrak{M}(f_X)$ always holds.

Lemma 3 (Perturbation of partial correlations). *Let Σ be an $(s+2)\times(s+2)$ matrix with $\mu I \preceq \Sigma \preceq MI$, and let $\|\hat{\Sigma} - \Sigma\|_{\text{op}} \le \delta \le \mu^2/(4M)$. Then every partial correlation of two coordinates given the others satisfies $|\hat{\rho} - \rho| \le 8M\delta/\mu^2$.*

Proof. Let $\Omega = \Sigma^{-1}$ and $e = 2\delta/\mu^2$. Since $\delta \le \mu/4$, $\|\hat{\Omega} - \Omega\| \le \|\hat{\Omega}\|\|\hat{\Sigma} - \Sigma\|\|\Omega\| \le \delta/((\mu - \delta)\mu) \le e$, so every entry of Ω moves by at most e; moreover $\Omega_{xx}, \Omega_{yy} \ge 1/M$ and $eM \le 1/2$. With $\rho = -\Omega_{xy}/\sqrt{\Omega_{xx}\Omega_{yy}}$,

$$|\hat{\rho} - \rho| \le \frac{|\hat{\Omega}_{xy} - \Omega_{xy}|}{\sqrt{\hat{\Omega}_{xx}\hat{\Omega}_{yy}}} + |\rho|\left|\sqrt{\tfrac{\Omega_{xx}\Omega_{yy}}{\hat{\Omega}_{xx}\hat{\Omega}_{yy}}} - 1\right|.$$

The first term is at most $e/(1/M - e) \le 2eM$. Since $\Omega_{xx}/\hat{\Omega}_{xx} \in [1 - eM, 1 + 2eM]$, and likewise for y, the second term is at most $2eM$. The sum is $4eM = 8M\delta/\mu^2$. □

Lemma 4 (BH sandwich). *For any M_f p-values and level α, the Benjamini–Hochberg procedure rejects every $p \le \alpha/M_f$ and no $p > \alpha$.*

Proof. Let $k_0 = \#\{p \le \alpha/M_f\}$. Then $p_{(k_0)} \le k_0\alpha/M_f$, so the BH index $\hat{k}$ is at least k_0, and every rejected $p_{(i)}$ satisfies $p_{(i)} \le \hat{k}\alpha/M_f \le \alpha$. □

For partial-correlation tests, $p \le \alpha'$ exactly when $|\hat{\rho}| \ge \zeta(\alpha')$ for a decreasing function ζ. Let $M_{\max}$ be the size of the largest BH family. By Lemma 4, under each family used by VL-PCMCI (per block and per merge, pooled, or per block and per pass), a final test rejects whenever $|\hat{\rho}| \ge \zeta_{\text{hi}} = \zeta(\alpha/M_{\max})$ and never when $|\hat{\rho}| < \zeta_{\text{lo}} = \zeta(\alpha)$. The exact-recovery statement below therefore covers all three families; false-discovery-rate control is a different question (Section B.7).

Theorem 2 (Same-data error decomposition). *Assume Assumption 2 for a stable linear VAR with partial-correlation tests (so that Lemma 1 applies), and consider the cross-block refinement with $R \ge 2$ and the confirmation re-test, or block merging whose coarse graph satisfies the routing condition of Corollary 2(c). Let $\mathcal{Q}$ be the deterministic family of all queries the algorithm could run: each of the N^2L candidate edges with each conditioning set of at most s variables from the lag window, so that $\log|\mathcal{Q}| \le (s+2)\log(2NL)$. Let $\gamma = \min_{e\in E^\star}|\rho(e)| - \theta$ be the screening margin; let λ be the smallest $|\rho(e \mid Z)|$ over true edges e and sets Z with $|Z| \le s$; and let η be the largest population $|\rho|$ of a tested non-edge given its final conditioning set ($\eta = 0$ when every tested non-edge has a separating set). Let $u > 0$ satisfy $\eta + u \le \zeta_{\text{lo}} \le \zeta_{\text{hi}} \le \lambda - u$, and let the PC removal level satisfy $\zeta(0.05) \le \zeta_{\text{hi}}$. Then*

$$P(\hat{E} \ne E^\star) \le \underbrace{|E^\star| a_0 e^{-bT_{\text{eff}}\gamma^2}}_{\delta_{\text{screen}}} + \underbrace{\mathbf{1}\{\eta > \zeta_{\text{lo}} - u\} + P(\textit{protection fails})}_{\delta_{\text{sep}}} + |\mathcal{Q}|\, a_s\, e^{-bT_{\text{eff}}u^2},$$

where a_0 is the constant of Assumption 2 for $s = 0$. Under the stated threshold condition the indicator vanishes, so δ_{sep} is the probability that protection fails.

Proof. Let $\mathcal{E}$ be the event that $|\hat{\rho}_\omega - \rho_\omega| < u$ for all $\omega \in \mathcal{Q}$ and that every true edge passes the screen. By Assumption 2 and a union bound over $\mathcal{Q}$ and over the $|E^\star|$ true edges, $\mathcal{E}$ fails with probability at most the first and last terms. On $\mathcal{E}$, with protection and $\eta \le \zeta_{\text{lo}} - u$, every executed test is some $\omega \in \mathcal{Q}$. A true edge has $|\hat{\rho}| > \lambda - u \ge \zeta_{\text{hi}}$; it is therefore kept by the within-block PC step, whose level 0.05 gives a threshold at most ζ_{hi}, and accepted by every BH family (Lemma 4), so the inclusion step of Corollary 2(a) holds on the sample path. At the final test of a non-edge, $|\rho_\omega| \le \eta$, so $|\hat{\rho}| < \eta + u \le \zeta_{\text{lo}}$ and the non-edge is rejected. Decisions on non-edges made before the final test only add members to decided sets, which protection makes harmless. Hence $\hat{E} = E^\star$. □

Remark 6 (Interpretation and scope). The first two terms are the population requirements seen through finite data; the last is the price of selection, since the level must control every query the algorithm could have run. The bound is small once $u \gtrsim \sqrt{(s+2)\log(2NL)/(bT_{\text{eff}})}$, so a level that accounts for the screen requires true-edge signals of order only $\sqrt{\log(NL)/T}$. The constants are loose, and the bound is numerically vacuous at $T = 4{,}096$ because of the $(2NL)^s$ union; its content is the $\log N$ dependence and the calibration requirement $\zeta_{\text{lo}} \ge u$, which a level held fixed while $\log|\mathcal{Q}|$ grows eventually violates (Proposition 5 describes the consequence). The theorem does not cover the cross-block refinement without the confirmation re-test, where η is a maximum over tested non-parents: under random blocks of 5,000 variables almost every target has a parent outside its block, within-block dependent non-edges have defects up to about 0.09 (a two-block check with four autocorrelated series), and hence $\eta > \zeta_{\text{lo}} - u$. Such pairs are rare, about $1.37/200$ per target, or a few thousand at 10^6 variables.

Proposition 4 (Split-sample confirmation)**.** *Assume Assumption 2 for a stable linear VAR, and let the process be β-mixing with coefficient $\beta(g)$ at gap g. Run the pipeline on $D_A = \{t \le T_A\}$, freeze the family $\mathcal{F} = \{(e, Z_e)\}$ of final queries, with $|\mathcal{F}| = M \le Nm$, and test it on $D_B = \{t \ge T_A + g\}$. Then*

$$P(\hat{E} \ne E^\star) \le \delta_A + \beta(g) + M\, a_s\, e^{-bT_{B,\text{eff}}u^2},$$

where the discovery error on D_A, bounded by Theorem 2 restricted to true-edge queries, satisfies $\delta_A \le \delta_{\text{screen}}(T_A) + P(\textit{protection fails}) + |E^\star|(2NL)^s a_s e^{-bT_{A,\text{eff}}(\lambda-\zeta_{\text{hi}})^2}$.

Proof. By Berbee's coupling, D_B is independent of D_A except on an event of probability at most $\beta(g)$. Conditionally on D_A the M queries are fixed, and Assumption 2 with a union bound over them concludes, since on the discovery event every final query of a non-edge has $\rho = 0$ and every true edge has $|\rho| \ge \lambda$. □

Remark 7 (Cost of splitting). The multiplicity drops from $(s+2)\log(2NL)$ to $\log(Nm)$. For stable Gaussian VARs $\beta(g)$ decays geometrically (Wong et al., 2016); its dependence on the dimension at $N = 10^6$ is an assumption. Splitting costs power: on half of the sample a true edge's MCI z-score is about $0.155\sqrt{2048} \approx 7.0$, against a Bonferroni threshold of about 6.3 for $\alpha/(Nm)$.

Remark 8 (Fresh-data confirmation). On fresh data, for example a later segment after a gap, the frozen final queries form the null family, and a per-target Bonferroni level is valid up to screening, separating-set and mixing terms (Wasserman & Roeder, 2009; Wong et al., 2016). On the same data it is not. This remedy is proposed, not part of the evaluated pipeline.

Remark 9 (Per-target Bonferroni). No reported run used a per-target level. With split data, under Proposition 4, the per-target level $\alpha/|\mathcal{F}_j|$ bounds the per-target family-wise error by $\alpha + \beta(g) + P(\text{a separating set is missing})$ plus the calibration error of the p-values on D_B. When the parents are conditioned on, the target residual is the innovation, a martingale difference, so the calibration is asymptotically exact (the extreme tail is assumed). Consequently $\mathbb{E}[\text{FP}] \lesssim \alpha N$, and the false-discovery proportion is flat in N (Wasserman & Roeder, 2009, Thm. 4.1). On the same data, validity would require a union over the conditioning sets the algorithm might choose, as in Kalisch & Bühlmann (2007), at level $\alpha/(NL(2NL)^{s-k})$; the cheaper $\alpha/(NL)$ is not covered (Wasserman & Roeder, 2009, Thm. 4.7). Bonferroni cannot repair a missing separating set, because the corresponding null is false.

Remark 10 (Three kinds of false edges). A false edge e of target j with final conditioning set Z_e is of one of three kinds: *no separating set*, if Z_e does not separate (the conditional-independence null is false); *miscalibrated*, if Z_e separates but omits a parent of j, so the target residual stays autocorrelated; or *selection-induced*, if $\mathcal{P}_j \subseteq Z_e$. If $\mathcal{P}_j \subseteq Z_e \subseteq \text{Past}_t$, the population residual of X_t^j given Z_e is exactly ε_t^j, so conditioning on the parents separates the first two kinds from the

third. Four checks distinguish them: reconstructing Z_e for every tested non-edge gives the fraction of false edges with $\mathcal{P}_j \subseteq Z_e$; re-running the cross-block step with $Z_e \cup \mathcal{P}_j$ removes the first two kinds; re-testing the frozen (e, Z_e) on fresh data removes the third; and d-separation in the true graph splits the first from the second. The analysis predicts a fraction near one for the first check, almost no change for the second, and a drop to the nominal rate for the third.

B.5 False positives after screening

A candidate edge (i, j, τ) is *independent* if its population lagged correlation is zero at every lag. In a linear VAR with independent innovations this generically means that no series is an ancestor of both i and j in the summary graph, counting a series as its own ancestor. The null checks below realize independence exactly by pairing series from two independently generated systems.

Proposition 5 (Empirical scaling law for false positives)**.** *Let q be the probability that the screen keeps a candidate edge between two independent series ($|\hat{\rho}| \geq \theta$), so that it keeps about N^2Lq such candidates. When the screen and the final test share data, nearly all of them pass the final test, while conditioning on the parents removes the dependent non-edges up to a pass rate ℓ. Hence, with m_{rel} dependent non-edges kept per target,*

$$\mathrm{FP} \approx N^2Lq + m_{\mathrm{rel}}\ell N.$$

Remark 11 (Status of the law). The count N^2Lq is exact given q; the rest is an empirical law. The rate q is counted, and the pass probability, m_{rel} and ℓ are simulated; nothing is fitted to the scaling study. With either of two estimates of q, the law predicts the scaling study's non-self false positives within 10% at every size (Table 3).

Screen false-positive rate. For independent linear processes, $\sqrt{T'}\hat{\rho}_{ij}(\tau) \Rightarrow \mathcal{N}(0, \kappa_{ij})$, where T' is the number of lagged pairs and $\kappa_{ij} = \sum_h \rho_{ii}(h)\rho_{jj}(h)$ (Bartlett's formula; Brockwell & Davis, 1991, Sec. 11.2). On the scaling study's generator, the formula and $T \cdot \mathrm{mean}(\hat{\rho}^2)$ both give $\kappa = 1.235$. The i.i.d. formula ($\kappa = 1$) undercounts null exceedances by a factor of 3–13 at $\theta = 0.05$–0.08. Because the threshold lies about 5.75 standard deviations out, q is counted directly rather than extrapolated from a tail approximation. A first count on two systems gave $q = 6.7 \times 10^{-9}$, and an extension to seven independent pairs, specified in advance, gave 7.86×10^{-9} (95% interval $[7.05, 8.74] \times 10^{-9}$ after widening for dispersion $\varphi = 1.57$). The one pair among these seven drawn from the generator used in the runs gave 93 exceedances, against 73.7 for the other six, which came from an independent re-implementation. Three further pairs from the run generator gave $q_{\mathrm{run}} = 8.59 \times 10^{-9}$ (interval $[7.69, 9.56] \times 10^{-9}$) over four pairs, 1.13 times the re-implementation's 7.58×10^{-9}, a difference that is not significant ($p = 0.085$).

The primary rate is q_{run}, because the scaling study used that generator; it is also the rate of the 2×10^6 prediction below, and with it the observed-to-predicted ratio at the five sizes is 0.91–0.96, every size lying inside its interval. As a sensitivity analysis, chosen after q_{run} was seen (the three extra pairs were specified in advance, this estimator was not), pooling all ten pairs is justified because the two generators are identical in law and their rates do not differ significantly; it gives 776 exceedances in 9.72×10^{10} candidate edges, $q = 7.98 \times 10^{-9}$ (interval $[7.37, 8.63] \times 10^{-9}$, $\varphi = 1.22$), within 3% of the Fisher-z value, and ratios of 0.95–1.02. Adding the reverse lag directions gives $q = 7.73 \times 10^{-9}$ (ratios 0.97–1.05). The excess of the run generator is smooth in θ: the cumulative ratio of the two generators is 1.006, 1.019, 1.067, 1.099 and 1.133 at $\theta = 0.08$, 0.09, 0.094, 0.096 and 0.10, which is suggestive but not established. On the first seven pairs, the cumulative ratio of observed to Fisher-z exceedances is 0.99, 0.99, 0.98, 0.99 and 0.96 at $\theta = 0.09$, 0.094, 0.096, 0.098 and 0.10, and per-bin ratios stay within 0.97–1.03 up to $\theta = 0.102$. The sizes of the scaling study share q and the pass rate of dependent non-edges, so together they form *one* test of a two-parameter law; the four sizes up to 10^6 were observed before the law was written, and only the 2×10^6 size is out of sample.

Other inputs. A re-implementation of the scaling study's generator (in-degree 3, coefficients ± 0.15, autoregressive coefficient 0.3, half of the parents in other blocks, $T = 4{,}096$, $L = 30$) supplies the remaining inputs. With the scaling study's conditioning (target parents, no source conditions), the screening and MCI z-scores of null pairs correlate at 0.90, and for selected pairs the median ratio of the MCI to the screening correlation is 0.81. Unselected nulls are calibrated (0.9% rejected at level 0.01, z-variance 1.03), whereas 94.8% of nulls screened at $|\hat{\rho}| \geq 0.06$ pass level

0.01, and 75.9% with full MCI conditioning. In one system with $\theta = 0.1$, the screen keeps all true non-self edges and 1.37 dependent non-edges per target, at both 2,000 and 18,000 series; their median partial correlation given $\mathcal{P}_j$ is 0.011, and at the scaling study's effective per-pass BH threshold of about 0.0071, 1.27% pass (35 events, relative standard error 17%). Inside one 18,000-series system, the bulk variance $T' \cdot \mathrm{mean}(\hat{\rho}^2)$ over pairs with $|\hat{\rho}| < 0.06$ equals the exact-null value to 10^{-4}, which rules out a diffuse background of weak correlations. The structural reason no dependence mechanism produces the N^2 term is separate: dependent pairs, including distant common ancestors reached by treks of bounded length, are bounded in number per target in a sparse graph and contribute $O(N)$ false positives in total, so only independent pairs, whose number grows as N^2L, feed the N^2 term, and q counts them.

Table 3: Proposition 5 against the scaling study's non-self false positives at level 0.01 (one run per size; nothing fitted to these runs). Predictions use q_{run}, the rate counted on the four pairs from the run generator. "Independent" is N^2Lq and "dependent" is $m_{\mathrm{rel}}\ell N$. The 95% intervals combine the null count with ± 2 standard errors of the pass rate of dependent non-edges (relative standard error 17%). The last column gives the observed-to-predicted ratio with the rate pooled over all ten pairs, a post hoc sensitivity analysis. The 2×10^6 row is out of sample: its prediction was fixed before that run was scored.

N	independent	dependent	predicted [95%]	observed	obs./pred.	pooled
100,000	2,577	1,750	4,327 [3,467, 5,210]	3,949	0.91	0.95
250,000	16,107	4,375	20,482 [17,322, 23,784]	19,177	0.94	0.99
500,000	64,429	8,750	73,179 [63,496, 83,430]	69,494	0.95	1.01
1,000,000	257,716	17,500	275,216 [242,400, 310,306]	262,672	0.95	1.02
2,000,000	1,030,864	35,000	1,065,864 [946,431, 1,194,394]	1,018,316	0.96	1.03

Remark 12 (Scope of the law). The law reproduces the mechanism and the N^2 growth. Every observed count lies inside the prediction's 95% interval (Table 3). The fitted N^2 component at 10^6 is 248k, against 258k predicted with q_{run} (240k pooled), and it carries about 94% of the false positives there; the Fisher-z extrapolation predicts 263k. The surviving false edges have screening correlations of 0.100–0.106, just above θ. An audit of the scaling study's screening output against the true graph confirms the simulated inputs on the runs' own data. The screen's false-positive rate at $\theta = 0.1$ is 8.08, 8.20, 8.22 and 8.23×10^{-9} at 10^5, 2.5×10^5, 5×10^5 and 10^6 variables, inside both intervals set before the audit ($[7.37, 8.63] \times 10^{-9}$ pooled and $[7.69, 9.56] \times 10^{-9}$ for q_{run}). The screen admits 2,411, 15,334, 61,611 and 246,846 independent candidate edges, of which 99.9% survive the final test; at 10^6 the cross-block pass rate is 0.9994, and these survivors are 93.9% of the 262,672 non-self false edges. Dependent non-edges at a wrong lag of a true pair pass at 1.15%, against 1.27% simulated, while the few dependent pairs with longer treks (335 at 10^6) pass almost always. The generator configuration matches, and recomputation in double precision moves $|\hat{\rho}|$ by at most 2.1×10^{-5}. The law is not calibrated for much stricter levels, where survival depends on the joint tail of the screening and test statistics, which was not measured.

Prediction at 2×10^6 variables. Before the 2×10^6-variable run of the same design was scored, the law with q_{run} predicted 1.07×10^6 non-self false positives at BH level 0.01 (interval $[0.95, 1.19] \times 10^6$); an extrapolation of the observed counts with $\mathrm{FP} = aN^2 + bN$ gave 1.02×10^6 (interval $[0.96, 1.08] \times 10^6$), and the pooled rate gave 0.99×10^6. The run produced 1,018,316 non-self false positives, with recall 0.99995 and false-discovery proportion 0.145: the observed-to-predicted ratio is 0.96 for the law, 1.00 for the extrapolation and 1.03 for the pooled rate, and the count lies inside every interval (Table 3). It is 3.88 times the count at 10^6, against 4 for pure N^2 growth.

Recall cost of a stricter screen. The population correlation of a ± 0.15 edge is about 0.15. A screen admitting one independent candidate edge in expectation at 10^6 needs $\theta \approx 0.132$ (exact Gaussian quantile; the sub-Gaussian bound gives 0.137), and would keep only 77–85% of true edges. The screening-family level, BH at level $0.01/(N^2L)$ applied to the final MCI p-values (Section C), predicts, with the conditional signal $\lambda = 0.155$ of Table 2, row 8, and the Fisher-z noncentrality, recall losses of 1.8, 2.6 and 3.9% at 2.5×10^5, 5×10^5 and 10^6 variables; the observed losses are 1.9, 2.9 and 4.3%. The calculation is steep in λ: 0.150 doubles the loss, and 0.160 halves it.

B.6 INCLUDING MORE VARIABLES

Remark 13 (Motivation). Discovery is often restricted to a hand-picked subset of variables because the full set is too costly to search. Every variable left out is a potential hidden common cause, and a hidden common cause of two included series creates a false edge between them that no test on the subset can remove. Scaling discovery up to all available variables therefore moves the analysis toward the causal sufficiency that Assumption 1 requires: including more variables can only remove false edges among the variables of interest, and including all of their parents removes every one of them (Proposition 1).

Setting. The rule considered here is idealized, not the implemented one. The full system is a stable linear VAR over a universe U of series, and Assumption 1 holds for U but not necessarily for the observed series. Fix a target set $S \subseteq U$ and nested observed sets $S \subseteq V_d \subseteq V_{d'}$, $d < d'$. The estimand is the set of true lagged edges with both ends in S, which does not depend on d. Under the *ancestral* rule, the final test of $X^i_{t-\tau}$ for X^j_t $(i, j \in S)$ conditions on the target's observed parents, possibly augmented by further observed ancestors chosen so that the conditioning sets are nested in d. Let F_d be the set of population false edges among S under this rule. In this notation Proposition 1 states that (i) every edge in F_d has a path to its target that enters through an unobserved parent, so $F_d = \emptyset$ once every parent of every target in S is observed; (ii) $F_{d'} \subseteq F_d$; and (iii) true edges among S are always accepted. Population accuracy on S is therefore non-decreasing in d, and exact once the targets' parents are observed.

Proof sketch of Proposition 1. (i) By the path argument of Lemma 2, a path that leaves the target toward the future is blocked at a future collider, and a path that enters through an observed parent is blocked at that parent; only paths through unobserved parents remain. (ii) Write $Z_{d'} = Z_d \cup W$, where W is a set of observed parents of the target y, so $W \subseteq \mathrm{An}(y)$. A set Z d-separates x and y if and only if Z separates them in the moral graph of the subgraph induced by $\mathrm{An}(\{x, y\} \cup Z)$ (Lauritzen, 1996). Since $W \subseteq \mathrm{An}(y)$, we have $\mathrm{An}(\{x, y\} \cup Z_d \cup W) = \mathrm{An}(\{x, y\} \cup Z_d)$, so adding W leaves this moral graph unchanged, and in an undirected graph every superset of a separator is again a separator. Hence $Z_{d'}$ d-separates x and y whenever Z_d does. The restriction to ancestors is essential: adding a collider or one of its descendants can open a path. Lemma 1 turns separation into zero partial correlation in the stable linear VAR over U. (iii) This is faithfulness (Assumption 1) applied to U. □

Remark 14 (Limits). Only the *set* of false edges shrinks, not their effect sizes: observing one parent can unmask another confounded path. In 70 of 200 random systems of eight series, one step made a confounded edge newly detectable at level 0.01, although the average number of detectable false edges fell from 6.8 to 0. The implemented rule is not covered: it conditions on every accepted edge, which can include a confounded non-parent. Screened PCMCI was non-monotone in 44 of the 200 systems, and the cross-block refinement without the confirmation re-test keeps within-block false edges whose driver lies in another block, even when every series is observed.

B.7 OPEN PROBLEMS AND SCOPE

Finite-sample false-discovery control. No finite-sample FDR guarantee is claimed for the pipeline as run, for five reasons: selected null p-values are not uniform under same-data screening (Proposition 5); nulls without a separating set are false hypotheses for the test actually run; the Student-t reference ignores serial dependence for targets whose parents are not all conditioned on; three different BH families are used, and later tests supersede earlier ones; and a union of per-family BH decisions does not control the graph-wide FDR (Benjamini & Yekutieli, 2001). A feasible route combines Proposition 4 with the Benjamini–Yekutieli procedure over the frozen family on D_B; it would give $\mathrm{FDR} \leq \alpha + \delta_A + \beta(g)$ plus the calibration error, and requires an implemented split.

Open directions. Rates as $N, T \to \infty$ follow from Theorem 2 once the constants of Assumption 2 are written out. Approximate nearest-neighbor screening has no recall guarantee. Local-separation conditions (Sondhi & Shojaie, 2019) are a natural route to graphs with hubs.

Scope. The paper claims no FDR or family-wise error guarantee for the pipeline as run, under any of its BH families; no accuracy advantage over PCMCI or screened PCMCI in general; and

no general gain from adding variables, since the gain in Section 5.5 is on a designed system with hidden common drivers and, at a fixed level, the screen's chance survivors grow as N^2. It gives no theorem for approximate screening, for block merging with a coarse graph of block means, or for graphs of real data; no causal validity beyond Assumption 1; nothing about nonlinear CI tests; and no prediction of false positives at stringent levels. Lemma 2 is known (Runge et al., 2019; Fesanghary, 2026).

C Experimental details

C.1 Experiment setup

Each study fixed its cases, search space, objective, comparators, and per-fit time limit before any data existed. Tuned methods received the same number of trials and the same promotion rule on development data and were then scored once on held-out data; a held-out evaluation is scored only when every fit has finished. A changed setup is a new held-out evaluation of the same data, and Table 4 counts them. Table 5 summarizes the data of each study.

Table 4: Study design. Block merging ran in the sequential CPU implementation and the cross-block refinement in the GPU implementation (on one CPU thread in the same-hardware study). "Eval." counts the held-out evaluations of the same data up to the reported one.

Study	Option	Hardware*	N	Held-out units	Eval.
CausalRivers	merging	CPU / CPU	32–128 stations	30 river windows	3rd
Scaling limit	refinement	GPU / CPU	500–20,000	80 fits (8 cells $\times$ 10 seeds)	1st
Same hardware	refinement	CPU (GPU) / CPU	100–30,000	three seeds per cell	1st
Scaling	refinement	GPU / none	10^5–2×10^6	one planted graph per size	1st‡

*VL-PCMCI/ baselines. ‡Second design: the earlier large-scale design failed at 250,000 variables (Section C.7).

Hardware differs across studies, and Table 4 states it per study. Wall times from different hardware are never divided in the main text.

Table 5: Data and held-out units of each study.

Study	Data	Held-out
CausalRivers	River discharge, 15 panels of 32/64/128 stations; 2 pilot, 15 development windows	30 windows, third evaluation
Scaling limit	Linear sparse VAR, $T = 4{,}096$; $N = 500$–$20{,}000$ at $L = 30$; $L \in \{1, 5, 10\}$ and a long-delay condition at $N = 2{,}000$	80 fits (10 seeds $\times$ 8 cells)
Scaling	Degree 3, ± 0.15, cross-community fraction 0.5, $T = 4{,}096$, $L = 30$, seed 1001	one graph per size, 10^5–2×10^6

C.2 Configurations

Table 1 gives the controls of each study, and Table 6 gives the configuration and the BH family behind each reported VL-PCMCI result. Four points matter for the theory. First, the scaling study's candidates come only from the exact threshold screen on the data: at 10^6 it kept 6,213,320 candidates against 4,000,000 true edges, with screen recall of at least 0.9997, and its blocks are a random permutation of the variables into groups of 5,000, independent of the truth. Second, the GPU implementation adds the target's own lags to the conditioning sets of both testing steps, after the scored parents and up to the cap of 6, so the second pass conditions on the autodependencies found within blocks; the CPU implementation (small panels) does not. Third, the scaling-limit study ran the cross-block refinement twice without the PC step on a screened Leiden partition (resolution 1.0), with the scaling study's settings ($\theta = 0.1$, $\alpha = 0.01$, cap 6, 50 combinations). Fourth, the cross-block refinement recomputes the accepted cross-block edges in every pass, and its output is the within-block edges plus the *last* pass's cross-block edges, so an edge accepted in the first pass can be dropped in the second.

Table 6: VL-PCMCI configuration and BH family per reported result. The trial selected for the third CausalRivers evaluation was not recorded; its option and family are known.

Result	Configuration	Option	BH family
CausalRivers, third evaluation	not recorded	merging, centroid, full PC	per block and per merge
Scaling limit	fixed declared configuration	refinement, 2 iterations, no PC	per block, per iteration
Scaling	fixed declared configuration	refinement, 2 iterations	per block, per iteration

Table 7: The scaling-limit study: fits returned out of ten at lag 30 under the declared limits, with the median wall time of completed fits. Comparators ran their tuned policies on CPU nodes; VL-PCMCI ran the cross-block refinement with the scaling study's settings on shared GPUs, so wall times compare method plus hardware. A timeout or refusal is a measurement under the 24 h and memory limits, not a proof that a method cannot finish.

		Variables N (lag 30)			
Method	Hardware	500	2,000	5,000	20,000
VL-PCMCI	GPU	10/10, 44 s	10/10, 104 s	10/10, 287 s	10/10, 1,132 s
PCMCI	CPU	10/10, 10.2 h	0/10 at 24 h	not submitted[§]	refused[§]
Screened PCMCI	CPU	10/10, 3.71 h	0/10 at 24 h	0/10 at 24 h	refused[§]
Sparse VAR	CPU	10/10, 0.044 h	10/10, 0.63 h	10/10, 5.00 h	0/10[‡]

[§] Withheld by the experiment setup's declared memory rule for dense methods. [‡] Eight fits failed at a 256 GB allocation; two did not finish before the study closed.

C.3 CausalRivers benchmark

The panels come from the CausalRivers river-discharge data (Stein et al., 2025): 15 panels of 32, 64, and 128 stations with at least 80% of every series observed, giving 2 pilot, 15 development, and 30 held-out windows. On the held-out windows, tuned VL-PCMCI beat PCMCI by 0.0419 F1 on 28 windows and screened PCMCI by 0.0414 on 27, but trailed sparse VAR by 0.0015, winning 16 (F1 0.2849 against 0.2864). Summed fit wall time was 2,345 s for VL-PCMCI, against 23,671 s for PCMCI and 362 s for sparse VAR, all on CPU. The windows share one panel lineage, so no significance test was declared, and the benchmark is read as descriptive. The reported result is descriptive: the same held-out windows had been evaluated in two earlier development iterations, with a different edge score or conditioning rule each time.

C.4 Global financial market panel

The 204,879 daily series of Section 5.6 describe 83,466 listed instruments traded on 74 exchanges across 115 countries in 41 currencies, including 4,025 cryptocurrency pairs with on-chain and decentralized-finance aggregates. Around these sit 1,541 foreign-exchange pairs, from the major crosses to frontier currencies, 190 equity, currency, commodity and strategy indices, and 40 futures contracts across energy, metals, grains and financials. The remaining 42,311 of the 204,879 series are related non-financial indicators, about 21% of the panel. The run returned 587 million lagged links (426 million distinct pairs), about 2,900 per series; the screen's threshold was chosen for recall, and the false-edge law of Proposition 5 predicts that many of them are chance survivors. No ground truth exists for this panel.

C.5 Scaling-limit study

Table 7 gives completion at lag 30. The VL-PCMCI runs used the cross-block refinement, applied twice without the PC step, on a screened Leiden partition with the scaling study's settings (Section C.2). Where tuned PCMCI finished ($N = 2{,}000$, $L = 1$), it scored F1 1.0000, as did VL-PCMCI. The methods ran on different hardware, so the study is a completion result, not a speed ratio.

VL-PCMCI returned 10 of 10 fits in every cell, including the lag sweep ($L = 1, 5, 10$) and the long-delay condition at $N = 2{,}000$. At $L = 1$, its F1 was 1.0000 with a median of 46 s. At $L = 1$, tuned PCMCI and screened PCMCI finished all ten fits, with medians of 4.79 h and 2.14 h. At

$L = 5$, tuned screened PCMCI needed 10.46 h. Tuned sparse VAR finished every lag-sweep cell. Default policies are reported for completeness. Default screened PCMCI scored F1 0.9504 with FDP 0.0923. Default sparse VAR scored F1 0.0231 with FDP 0.9882. Default PCMCI returned no fit. At $N = 500$, tuned PCMCI returned all ten fits in a median 10.2 h, screened PCMCI in 3.71 h and sparse VAR in 0.044 h, against 44 s for VL-PCMCI on GPUs. At $N = 20{,}000$ sparse VAR returned no fit: eight failed at a 256 GB allocation and two did not finish before the study closed.

C.6 Same-hardware scaling study: all cells

The design, the predictions, and the stopping rule of this study, including its GPU series, were fixed before any fit. Every CPU fit ran on the same CPU model with one thread, 64 GB of memory, and a limit of 235 minutes; a fit past the limit is censored, not retried. The data come from the scaling study's generator (three non-self parents per series, coefficients ± 0.15, autocorrelation 0.3, half of the parents across four planted communities, $T = 4{,}096$, $L = 10$), linear at every size, on three fresh seeds; all methods fit the same datasets. PCMCI, screened PCMCI, sparse VAR, and block merging used configurations tuned on small development systems, not retuned for larger N. The cross-block refinement used the scaling study's frozen configuration, not tuned for these systems, with random blocks that ignore the truth (four blocks up to 10,000 series, six at 30,000, and 20 at 100,000). A memory rule declared in advance refused PCMCI at 10,000 and 30,000 series and screened PCMCI at 30,000, whose dense arrays would exceed 64 GB, and a rule declared in advance launched 30,000 only for a method that finished all three fits at 10,000. The GPU series adds one GPU to the same host limits; it is a hardware reference, not part of the same-hardware comparison. Fit wall time covers screening, partitioning, and discovery, not data loading (Table 8).

Table 8: The same-hardware scaling study, linear systems: median fit wall time in seconds / mean non-self F1 over three seeds. CPU rows use one thread, 64 GB, and a 235-minute limit per fit; GPU rows add one GPU. Refinement is VL-PCMCI with the cross-block refinement in the scaling study's frozen configuration. Timeout: all three fits hit the limit.

Method	100	300	1,000	3,000	10,000	30,000	100,000
PCMCI	582 / 0.9961	5,657 / 0.9972	timeout	timeout	refused[§]	refused[§]	—
Screened PCMCI	37.3 / 0.9967	261 / 0.9928	2,684 / 0.9889	timeout	timeout	refused[§]	—
Sparse VAR	6.4 / 0.9978	69.1 / 0.9943	1,313 / 0.9833	7,135 / 0.9600	timeout	not launched[‡]	—
Block merging	39.3 / 0.9950	683 / 0.9867	8,615 / 0.9301	timeout	timeout	not launched[‡]	—
Refinement	2.6 / 0.9983	6.5 / 0.9985	21.3 / 0.9981	90.2 / 0.9980	427 / 0.9980	2,690 / 0.9975	—
Refinement, GPU	4.0 / 0.9983	7.7 / 0.9985	19.9 / 0.9981	54.7 / 0.9980	191 / 0.9980	524 / 0.9975	1,587 / 0.9963

[§]Refused by the declared memory rule. [‡]Launched only after all three fits at 10,000 finish (declared rule). —: not in the design.

On linear systems the cross-block refinement on one CPU thread finished every size and was the only method on the CPU accuracy–time frontier at every N (three seeds, no test). Its time grew as $N^{1.21}$ overall, with local exponents from 0.8 to 1.3 up to 10,000 series and 1.67 from 10,000 to 30,000, where the screen's N^2 arithmetic dominates; on one GPU the exponent was 0.89. The completion frontiers (all three fits within the limit) were 300 series for PCMCI, 1,000 for screened PCMCI and for block merging, 3,000 for sparse VAR, 30,000 for the refinement on CPU, and 100,000 on GPU. Over their completed cells the baselines' exponents were 2.07 (PCMCI), 1.86 (screened PCMCI), and 2.11 (sparse VAR), and 2.34 for block merging. The best baseline F1 was 0.998, 0.997, 0.989, and 0.960 at 100, 300, 1,000, and 3,000 series, against 0.998 for the refinement at each; sparse VAR's F1 fell with N at a penalty tuned on smaller systems.

C.7 Scaling study: levels, replicates, screen audit, and resources

Table 9 gives the primary run at each size. The 2×10^6 run resumed from saved within-block results, so its wall time and memory are not reported, and its strict-level rerun was not completed before submission. The generator, the pipeline settings, and the level 0.01 were fixed before the runs, and each size from 5×10^5 up is one run. The false-edge law's rates were estimated on independent simulated systems (Section B.5), and nothing was fitted to these runs; the 2×10^6 run was scored after its prediction had been written and was not used to set the law.

At the primary level the non-self false positives were 3,949, 19,177, 69,494, 262,672, and 1,018,316 at the five sizes, with 13, 39, 74, 148, and 316 misses. The true edges inside the learned blocks,

Table 9: The scaling study at the level 0.01 fixed in advance (cross-block refinement, one planted graph per size). Wall and memory are for discovery. GPUs are two per run up to 5×10^5. At 10^6 the run was nine segments resumed from checkpoints, the later ones on one GPU.

N	F1	Precision	Recall	FDP	Wall	Memory
10^5	0.9934	0.9870	0.99996	0.0130	1,547 s	20 GB
2.5×10^5	0.9874	0.9751	0.99995	0.0249	3,220 s	43 GB
5×10^5	0.9773	0.9557	0.99995	0.0443	≈8,700 s†	67 GB
10^6	0.9580	0.9195	0.99995	0.0805	7.18 h	≈132 GB
2×10^6	0.9218	0.8549	0.99995	0.1451	—	—

†Resumed after a scheduler cancellation. The cross-block time counts only the resumed portion.

which only the within-block step can recover, were 15,030, 15,130, 15,081, and 15,207; the cross-block refinement recovered the remaining cross-block edges with recall at least 0.99995. Fresh-seed runs agree within 0.0002 F1 at 10^5 (0.9932–0.9934) and within 0.0001 at 2.5×10^5 (0.9873–0.9874). Two fresh-seed runs at 5×10^5 scored F1 0.977314 and 0.977258, against 0.977335 for the primary run. They used 500 blocks of 1,000 variables rather than 100 of 5,000, so they replicate the result, not the partition. At 2.5×10^5 the stage times were 399 s for screening, 2,379 s within blocks, and 362 s across blocks. At 10^6, the within-block step took about 5.5 hours of segment wall time for 141 million tests, and the cross-block refinement took 1,512 s for 9.2 million tests. One-GPU and eight-GPU runs at 10^5 produced identical edge sets (405,086 edges). The eight-GPU timing was contended and is not reported. At 2.5×10^5, a second machine with three GPUs reproduced the true positives (749,961) and misses (39) exactly.

Other predictions of the theory. *Screening recall.* An earlier large-scale design paired an approximate nearest-neighbor screen with a graph of block averages that chose which blocks to merge. It failed at 2.5×10^5 variables: non-self recall 0.000644, F1 0.001257. The screen kept 0.9% of the true candidates, which caps recall near 0.009, and the graph of block averages proposed no cross-block merges, as the counterexamples of Section B.3 predict. With the exact screen and the cross-block refinement, the second design reached recall 0.99995 on the same input. *Separating sets.* The refinement was applied twice, with autodependencies in every final conditioning set. Corollary 2(a) needs the rank margin. That is plausible here, because true parents have marginal correlations of 0.13–0.20 against about 0.10 for screened non-parents, but it is not proven. A third application changed FDP by at most 0.0011, and capping the candidates per source left the false positives unchanged (19,055 against 19,177 at 2.5×10^5). Both nulls are consistent with Corollary 2(a) but do not test it. *Test level.* Successive growth factors of the false-positive count, 4.86, 3.62, and 3.78, track the N^2 ratios of 6.25, 4, and 4 more closely than the N ratios. Observed over predicted is 0.91–0.95 with the screening rate counted on four systems from the scaling study's generator, and 0.95–1.02 pooled over ten simulated systems. Tightening the level to 10^{-4} removed only 7.8% of the false positives at 10^6.

Screen audit. An audit fixed in advance re-read the run's own screen output. It counts a screened pair as unconnected when the true summary graph has no trek of length at most four between the two series; this does not exclude longer connections. The screening rate of unconnected pairs at $\theta = 0.1$ was 8.08, 8.20, 8.22, and 8.23×10^{-9} at the four sizes, and the rate at 10^6 fell inside both intervals set in advance. Of the 246,846 unconnected pairs screened at 10^6, 246,700 passed the tests; they are 93.92% of the 262,672 false positives. Their pass rate across blocks was 0.9994. Directly connected pairs at a wrong lag passed at 1.15%. The two screening-rate estimates behind the law's prediction (Table 3) come from generators identical in law and differ by 13% ($p = 0.085$). Recomputing the correlations in double precision changed them by at most 2.1×10^{-5}.

Table 10 lists every level variant. Each reran the cross-block refinement alone from the primary run's checkpoints. All variants were declared before they ran. The screening-family point is BH at level $0.01/(N^2 L)$, that is 5.3×10^{-15}, 1.3×10^{-15}, and 3.3×10^{-16}.

Split-sample confirmation (diagnostic). A design fixed before the run generated two independent 4,096-step trajectories from the same sparse VAR at 10^4, 3×10^4 and 10^5 series (two fresh graph

Table 10: Level variants of the cross-block refinement from identical checkpoints (one seed per size). Entries are non-self FDP, with F1 and recall where they were reported.

Variant	2.5×10^5	5×10^5	10^6
Primary, BH 0.01, 2 iterations	0.0249	0.0443	0.0805
Third iteration	0.0260	0.0452	0.0814
BH 10^{-4}	0.0198	0.0388	0.0747
BH $0.01/(N^2L)$	0.0005 (recall 0.981)	0.0004 (recall 0.971)	0.0004 (recall 0.957)

seeds per size), selected candidates and conditioning sets on the first trajectory only, and applied identical confirmation tests to the first (reused) and the second (independent) trajectory, at a global Bonferroni level of 0.01 over the frozen candidates. With reused data the confirmation kept a mean of 15.5, 66 and 429 false edges at 10^4, 3×10^4 and 10^5 series; on the independent trajectory it kept none in all six cases, with mean recall above 0.9999 in both arms. Because the independent arm uses a second trajectory, the diagnostic does not measure the recall cost of splitting one fixed-length trajectory, nor the gap that two portions of one trajectory need; with two seeds per size and approximate time-series calibration it supports the mechanism, not a finite-sample error guarantee. The diagnostic tests data reuse after screening; it does not run the full pipeline, and the independent arm uses twice the data.

C.8 Refinement-options comparison

On the scaling study's generator, three refinement variants ran from shared screening and within-block results: block merging with and without its PC step, and the cross-block refinement (Table 11). All scored F1 of at least 0.997; the cross-block refinement was the fastest at every size (13.5 times faster than block merging without its PC step at 10^5), and block merging with its PC step did not finish at 10^5. The large-scale runs therefore use the cross-block refinement; block merging remains useful where re-testing within-block decisions matters (Through-stitch, Section C.9).

Table 11: Refinement-options comparison, development-selected settings: mean non-self F1 and refinement time in seconds. With matched nominal settings the ordering is the same.

Variant	512	2,048	10^4	10^5
Block merging, PC step	0.9998, 28.9 s	0.9998, 106.8 s	0.9995, 515.4 s	not finished
Block merging, no PC step	1.0000, 5.7 s	0.9999, 17.7 s	0.9999, 78.1 s	0.9998, 3,281 s
Cross-block refinement	0.9999, 2.2 s	0.9998, 5.0 s	0.9997, 26.0 s	0.9995, 244 s

C.9 Stress tests

Each family of Figure 5, plus Knot (a target with more parents than the conditioning cap), was generated with 64 series (plus one driver in Through-stitch and four hub targets in Knot), $T = 4{,}096$, $L = 10$, and five seeds, on the generating partition. Every VL-PCMCI variant used settings tuned on small development systems, as did PCMCI and sparse VAR. The predictions for each family were declared before the run, and all five held (Table 12).

Table 12: Stress tests as tests of the theory (five seeds each). Each family violates one requirement; the diagnostic measures the failure the theory predicts for the cross-block refinement ("refinement"), with block merging with its PC step ("merging") and PCMCI for reference.

Family	Violated	Predicted failure	Observed (refinement / merging / PCMCI)	Prediction
Patchwork	none	none	F1 0.995 / 0.996 / 0.994	confirmed
Hidden stitch	screening recall	cancelled parent missed	parent found in 0/5 / 0/5 / 5/5 seeds	confirmed
Through-stitch	separating set	spurious edges into the driver's children	199.6 / 5.0 / 0.8 spurious edges	confirmed
Bleed	bounded persistence	more chance screen survivors	screen 5.6 times larger than Patchwork; F1 0.976 / 0.989 / 0.994	confirmed
Knot	conditioning cap	spurious hub edges	12.8 / 9.8 / 6.4 spurious edges	confirmed

Hidden stitch and Knot are shared limits: no refinement recovers a parent the screen missed, and every capped method pays for a target with more parents than the cap. Through-stitch separates the options: block merging with its PC step re-tests the within-block decisions and recovers, while the cross-block refinement and block merging without the PC step keep the driver's spurious edges. Bleed costs the cross-block refinement 0.02 F1 and block merging without the PC step 0.09.

C.10 Including more variables

Design. The system adds 40 driver series (own lag 0.5, no parents) to 2,000 series from the scaling study's generator (three non-self parents at ± 0.15, autocorrelation 0.3, half of the parents across four generating blocks, $T = 4{,}096$, $L = 10$). Each driver feeds three to six targets, in disjoint groups, at lags one to three with coefficient ± 0.30; the coefficient is the smallest of three candidates that met a correlation rule written before its calibration, which used the data-generating process only. The 200 targets come from one generating block and include every driver's children. The observed universe at fraction f is the targets plus the first f share of a fixed permutation of the other 1,800 series and of the 40 drivers, so the subsets are nested (Table 13). VL-PCMCI ran one configuration at every f: the exact screen at $|\hat{\rho}| \geq 0.075$, PCMCI within four random blocks whose labels are fixed per series across f, and two passes of the cross-block refinement at level 0.001. Accuracy is lag-resolved F1 over non-self edges among the 200 targets only, against the full-system truth, over five seeds. A false edge among the targets is *confounding* when both endpoints are children of one driver; the others (non-confounding) include selection false edges between unrelated pairs. Cost is the fit time on one thread (screen, partition, and both testing steps), with up to 12 fits sharing a node.

Table 13: Observing more of a system: VL-PCMCI on the 200 targets, seed means over five seeds. False edges are per seed; non-confounding false edges include selection false edges between unrelated pairs (8.4 to 9.0 per seed). Cost is seconds on one thread.

f	N	Drivers	F1	Precision	Recall	FDP	Confounding	Non-confounding	Cost
0	200	0	0.462	0.301	0.998	0.699	261.4	10.8	3.3
0.1	384	4	0.484	0.320	0.998	0.680	238.6	10.4	4.8
0.25	660	10	0.511	0.344	0.998	0.656	213.2	10.0	8.5
0.5	1,120	20	0.570	0.399	0.998	0.601	166.0	10.2	17.1
1	2,040	40	0.775	0.635	0.998	0.365	56.2	11.2	37.6

Summary. Target F1 rose monotonically with the observed fraction, recall stayed at 0.998, and cost grew about linearly, so the gain is precision from removing false edges that hidden drivers created. The remaining false edges are pairs of siblings in one block whose driver sits in another: within-block decisions are final in the cross-block refinement, so such a pair keeps its false edge even when the driver is observed. Block merging or the confirmation re-test of Corollary 2(b) would remove them. The size of the gain is set by the designed drivers; the direction and the attribution of what remains are what the study shows.

D Extended related work

D.1 Comparison with the closest methods

Table 15 places VL-PCMCI among methods that reduce the search space of causal discovery or report discovery at large scale. The column "Guarantee" gives the strongest formal statement in the cited paper: *oracle* (correct output given perfect CI decisions), *asymptotic* (consistency as the sample size grows) or *none* (empirical evaluation only). The column "Largest scale" gives the largest number of variables the paper reports, not a limit of the method.

D.2 Relation of prior results to the analysis

Remark 15 (Conditioning on the parents). Suppose a conditioning set lies in the past of the target and contains all of the target's parents. A test of a past candidate given that set is then null if and only if the candidate is not a parent (Lemma 2). The consistency proof of PCMCI uses this fact through the weak-union property (Runge et al., 2019), local structure learners use its non-descendant

Table 14: Largest graph each paper reports running (not a limit of the method). N: series, or variables in i.i.d. rows; $\tau_{\max}$: largest lag searched; T: time points or sample size; $N\tau_{\max}$: lagged candidates per target; Compute: hardware and wall time as stated. n.r.: not reported; —: not applicable. VL-PCMCI rows: one run each.

Method	Largest N	$\tau_{\max}$	T	$N\tau_{\max}$	Compute
Temporal, constraint-based					
PCMCI+ (Runge, 2020)	40	5	500	200	CPU
PCMCI (Runge et al., 2019)	100	5	150; 300	500	CPU
D&C DBN (Ouyang et al., 2024)	1,323	1	$1{,}000 \times 6^a$	1,323	96-core CPU, 2.7 h
CaStLe (Nichol et al., 2025)	3,900[b]	1	n.r.	—	10 s
Temporal, score-based, neural, or Granger					
DYNOTEARS (Pamfil et al., 2020)	100	1	500	100	n.r.
Neural GC (Tank et al., 2022)	100	2	46×21^a	200	n.r.
CUTS+ (Cheng et al., 2024)	1,024	n.r.	1,000	n.r.	n.r.
Voxel-wise PCGC (Wu et al., 2013)	$\approx$43,413	n.r.	225	n.r.	n.r.
i.i.d. data (scale context)					
cuPC (Zarebavani et al., 2020)	5,361	—	63	—	1 GPU, 4.76 s
FGES (Ramsey et al., 2017)	10^6	—	1,000	—	120 processors, 11.0 h
VL-PCMCI, synthetic[c]	2×10^6	30	4,096	6×10^7	n.r.[c]
VL-PCMCI, synthetic[c]	10^6	30	4,096	3×10^7	$\leq$2 GPUs, 7.18 h
VL-PCMCI, global financial market[d]	204,879	n.r.	4,198	n.r.	4–8 GPUs, $\approx$5.5 d

[a] Independent sequences $\times$ time points per sequence. [b] Learned as 39 local 3×3 grid stencils, not as one joint search over all cells. [c] Same generator; non-self recall 0.99995 at both sizes; false discovery proportion 0.08 at 10^6 and 0.15 at 2×10^6; 10.5 GPU-hours at 10^6; no end-to-end time is reported at 2×10^6. [d] Private daily panel (79% financial market series) without ground truth, run with a recall-first configuration, so precision is unknown.

form (Tsamardinos et al., 2006; Aliferis et al., 2010), and Fesanghary (2026) prove it for stable MCI pruning of lagged candidates, noting that conditioning on past variables alone can open a collider path. In lagged discovery every candidate precedes the target, so the symmetry correction of local learners is not needed. This paper does not claim the lemma.

Remark 16 (Screening recall). Marginal screening misses a true parent whose direct and indirect effects cancel. PC-simple makes the required assumption explicit as partial faithfulness (Bühlmann et al., 2010), and sure-screening theorems require a lower bound on the marginal signal (Fan & Lv, 2008; Yousuf, 2018). Screening recall is also not sufficient. A causal partition must contain a witness subset for every false adjacency (Shah et al., 2025); partition-estimation-fusion re-tests within-cluster edges once cross-cluster parents are known (Gu & Zhou, 2020); and greedy equivalence search restricted to the true skeleton converges to a wrong graph in a four-variable linear example, while adaptive relaxation restores consistency (Nandy et al., 2018). These results concern i.i.d. data; Section 4 gives their lagged counterparts.

Concentration for one dependent trajectory. For stable Gaussian VARs, deviation bounds for sample covariances scale with the maximum of the spectral density. The bound degrades as the process approaches a unit root (Basu & Michailidis, 2015). Screening theorems for dependent series use functional-dependence measures (Yousuf, 2018). Splitting one trajectory into gapped blocks costs an additive β-mixing term (Wong et al., 2016); stable Gaussian VARs are geometrically β-mixing, with constants left implicit.

Selection and the test level. With independent cleaning data, screening, selection and Bonferroni over the selected set control the family-wise error up to the probability that screening fails (Wasserman & Roeder, 2009). Aggregating many random splits keeps asymptotic control (Meinshausen et al., 2009). Without a split, filtering is harmless only if the filter statistic is independent of the test statistic under the null (Bourgon et al., 2010). The Benjamini–Yekutieli procedure controls FDR under arbitrary dependence, given valid null p-values (Benjamini & Yekutieli, 2001).

Remark 17 (Blockwise multiplicity control). Applying Benjamini–Hochberg separately inside blocks does not control FDR over their union, and a screen that selects candidates on the same data invalidates the null p-values these procedures assume. This is the reason for the third requirement of Section 4.

Further related methods. Logical deduction avoids CI tests whose outcome follows from earlier ones, without changing the oracle output (Kim & Lee, 2026). Other PCMCI variants handle multiple datasets with latent contexts (Günther et al., 2023). GRACE prunes a high-recall CI skeleton with gated neural predictors (Fesanghary & Havaldar, 2026). Beyond Shah et al. (2025) and Gu & Zhou (2020), divide-and-conquer learners use causal cuts (Cai et al., 2013), weakly constrained superstructures (Wang & Wan, 2026), separation trees (Xie & Geng, 2008) and removable variables (Mokhtarian et al., 2025). CUTS+ learns group-level Granger structure before series-level structure (Cheng et al., 2024). That coarse-to-fine design parallels the block-level graph that block merging uses. GPU acceleration also covers nonlinear CI tests (Hagedorn et al., 2022). For linear lagged data, penalized VAR estimators select lags and parents with consistency guarantees (Shojaie & Michailidis, 2010; Basu & Michailidis, 2015). Regression-coefficient methods won the Causality 4 Climate competition on edge-ranking accuracy (Weichwald et al., 2020). VARLiNGAM identifies structure from non-Gaussian noise instead of CI tests (Hyvärinen et al., 2010). CausalRivers provides real directed labels between river stations; because it also has hidden weather confounding and time-varying lags, it tests robustness rather than the assumptions of Section 4 (Stein et al., 2025).

D.3 Search for large lagged analyses

A search for lagged or Granger analyses at scale in neuroimaging, genomics and penalized VAR estimation found the largest to be voxel-wise partially conditioned Granger causality on about 4.3×10^4 fMRI voxels (Wu et al., 2013). It conditions on community-averaged signals, an approximation motivated by sparsity (Marinazzo et al., 2012). FGES was applied to about 5.1×10^4 fMRI voxels, with time points treated as independent samples (Ramsey et al., 2017). Local space–time stencils reach several thousand grid cells by assuming spatial homogeneity (Nichol et al., 2025). The search found no lagged, CI-test-based run above these scales; the statement in Section 6 that VL-PCMCI is, to the authors' knowledge, the first PCMCI-family lagged discovery at millions of series rests on it.

Method	Data	Search reduction	Guarantee	Largest scale	What VL-PCMCI adds
PCMCI family and temporal screen-then-test					
PCMCI (Runge et al., 2019)	time series	PC-step parent pre-selection, then MCI tests	oracle	$N \leq 100$	Lag-wise screen instead of a global PC step; blocks and cross-block tests; separating-set condition
PCMCI+ / LPCMCI (Runge, 2020; Gerhardus & Runge, 2020)	time series	adaptive conditioning	oracle	$N \leq 40$	Scale; lagged-only, sufficient setting
F-PCMCI (Castri et al., 2023)	time series	transfer-entropy variable filter, then PCMCI	none	8	Lag-specific screen, blocks, refinement, theory
CEDAR (Fesanghary, 2026)	time series, AR(1) autodepen-dencies	lag screen, two targeted CI tests, stable MCI	oracle (under its assumptions)	$d = 75$	General lag structure; decomposition at scale
GRACE (Fesanghary & Havaldar, 2026)	time series	high-recall CI skeleton, then L_0-gated pruning	none	$d = 100$	CI-based final decision; theory; scale
High-dimensional constraint-based search and screening					
High-dim. PC (Kalisch & Bühlmann, 2007)	i.i.d. Gaussian	degree-bounded conditioning	asymptotic, exponential rate	$p = 2{,}187$	Dependent data; screening; blocks
rPC (Sondhi & Shojaie, 2019)	i.i.d.	conditioning order capped by local separation	asymptotic	$p = 500$	Lagged; screening; blocks
PC-simple (Bühlmann et al., 2010)	i.i.d., one response	correlation screen, then conditional tests	asymptotic	$p = 4{,}088$	Many targets and lags; separating sets
PenPC (Ha et al., 2016)	i.i.d. Gaussian	penalized-regression superstructure, then PC	asymptotic, two-step	$p = 1{,}000$	Lagged; marginal screen; blocks
SIS (Fan & Lv, 2008; Yousuf, 2018)	i.i.d. / time series, one response	marginal screening	sure screening	ultra-high p	Screening, CI confirmation and separating sets
Screen and clean (Wasserman & Roeder, 2009)	i.i.d. regression	screen, select, Bonferroni on a held-out split	asymptotic FWER	regression	Per-target CI tests; dependent data
Divide and conquer over a superstructure					
Causal partition (Shah et al., 2025)	i.i.d.	superstructure, overlapping causal partition, merge	oracle (latent-robust local learner)	10,000	Lagged; sufficient local learner; separating sets from the cross-block refinement
PEF (Gu & Zhou, 2020)	i.i.d. Gaussian	correlation clustering, local learning, fusion with re-tests	partial (Prop. 5)	$p = 3{,}620$	Lagged; end-to-end conditions
ARGES (Nandy et al., 2018)	i.i.d. Gaussian	superstructure-restricted greedy search with adaptive relaxation	asymptotic, high-dim.	thousands	Constraint-based lagged analogue
DBN divide-and-conquer (Ouyang et al., 2024)	2-slice DBN	PEF adapted to two-slice networks	none	$>1{,}000$	CI tests; theory; 10^6
Scale precedents and baselines					
FGES (Ramsey et al., 2017)	i.i.d. Gaussian	parallel greedy search, one-edge faithfulness screen	large-sample (GES)	10^6 (one run)	Lagged estimand; one dependent trajectory; CI tests
cuPC (Zarebavani et al., 2020)	i.i.d. Gaussian	GPU-parallel CI tests	as PC	5,361 (real data; 4,000 synthetic)	Screening and decomposition; lagged
Voxel PCGC (Wu et al., 2013)	fMRI time series	driver-informative and community-mean conditioning	none	$\approx$43,000	Explicit separating-set condition; ground truth
CUTS+ (Cheng et al., 2024)	time series	neural Granger with coarse-to-fine grouping	none	1,024	CI tests; theory
Neural-GC (Tank et al., 2022)	time series	group-lasso penalized neural Granger	none	$p = 100$	CI tests; theory; scale
DYNOTEARS (Pamfil et al., 2020)	time series	continuous optimization with acyclicity	none (finite sample)	$d = 100$	Scale; error control through the test level
Sparse VAR (Basu & Michailidis, 2015)	stable Gaussian VAR	ℓ_1 penalty	finite-sample estimation bounds	theory	CI-based edge decisions
VL-PCMCI (this paper)	time series	lag-wise screen, blocks, cross-block tests	oracle attribution (Section 4)	2×10^6	–

Table 15: Methods that reduce the search space of causal discovery or report large-scale runs. Scales are those reported in each paper; N, p and d denote numbers of variables.