Structurally Separated DAG Learning with Multi-Scale Normalized Closure

Shuai Feng
Arizona State University
sfeng51@asu.edu

Hao Yan
Arizona State University
hyan46@asu.edu

Abstract

Continuous DAG learning commonly enforces acyclicity through scalar functions of a weighted adjacency that aggregate evidence from weighted closed walks. We add a complementary layer: matrix-valued closure logits—explicit multi-hop structural relation scores for each ordered pair—supervised alongside data fit, sparsity, and a DAGMA-style barrier on the weighted adjacency. The distinction is not scalarization of the overall objective but how path evidence is represented and exposed before aggregation. *SMNC* (scale-normalized multi-scale closure) builds these scores on a structural gate by repeated squaring in log space and divides each scale by its path-length budget, counteracting product-style attenuation along long cycles; diagonal entries support node-wise cycle supervision and off-diagonals optional pairwise reachability cues. In the linear model, a bounded structure–magnitude factorization ties strong effective edges to strong gates; a nonlinear induced gate offers another construction with the same intent. For sufficiently strong cycles, normalized logits admit a length-independent lower bound; under a unique-active-cycle assumption, the hard final-scale readout yields a polynomial log-gate gradient factor rather than exponential product-style decay (Section 3.2). On cue-free, unstandardized Gaussian linear and nonlinear ER benchmarks, these models improve over DAGMA, run with default hyperparameters at a reduced budget (Appendix D.2), on denser graphs and are close to it on sparse ones.

1 Introduction

Learning directed acyclic graphs (DAGs) from observational data is a central problem in causal discovery and graphical modeling. Exact score-based search over DAGs is NP-hard [Chickering, 1996, Chickering et al., 2004], motivating both classical combinatorial families and continuous relaxations that optimize weighted adjacencies with gradient methods [Spirtes et al., 2000, Chickering, 2002, Heckerman et al., 1995, Tsamardinos et al., 2006, Gasse et al., 2014, Raskutti and Uhler, 2018, Solus et al., 2021, Heinze-Deml et al., 2018, Kitson et al., 2023, Zheng et al., 2018, Wei et al., 2020, Deng et al., 2023a, Bello et al., 2022]. Recent analyses also emphasize that practical behavior depends strongly on optimization geometry, identifiability assumptions, and benchmark design [Ng et al., 2022, Deng et al., 2023b, 2024, Ng et al., 2024, Reisach et al., 2021].

Continuous DAG learning commonly enforces acyclicity through scalar functions of a weighted adjacency. These objectives aggregate evidence from weighted closed walks, which can exhibit strong dependence on path length, edge magnitude, and repeated cycle traversals. We investigate a complementary representation: a matrix-valued closure whose entries provide explicit multi-hop structural relation scores. This representation permits node-wise cycle supervision and pairwise reachability constraints, while scale normalization mitigates attenuation along long paths. Our full objective still combines data fit, sparsity, a DAGMA-style log-determinant barrier on the weighted adjacency, and closure-side losses; what distinguishes SMNC is how path evidence is represented and exposed before this scalar aggregation, not the presence of a scalar barrier itself. A graph is acyclic exactly when strict diagonal self-reachability is zero, so cycle evidence can be read out per node; off-diagonal logits can absorb domain relations such as ancestry [Chen et al., 2016, Li and van Beek, 2018, Ban et al., 2024, 2025] that are awkward to state as one-hop edge labels. This does not

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

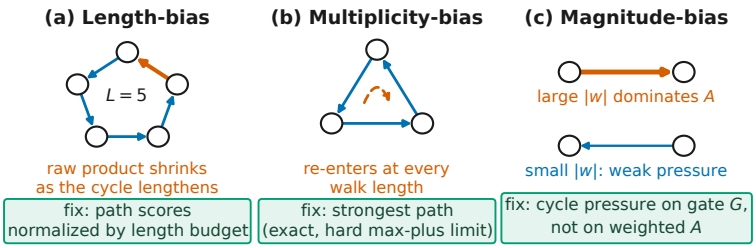

Figure 1: Effects that can arise when weighted-walk evidence is pooled into scalar acyclicity objectives, and corresponding design choices in our matrix-valued closure head: scale-normalized path scores mitigate length-dependent attenuation, the hard max-plus strongest-path limit reduces multiplicity from repeated traversals, and supervising the gate G rather than coefficient magnitudes in $A = W_f(M) \odot G(Z)$ separates structural support from effect size.

imply that DAGMA lacks a closure or that its scalar barrier is inherently defective. Section 2 reviews walk-based acyclicity penalties in this light.

When weighted-walk evidence enters only through pooled acyclicity terms, optimization can couple strongly to walk length, multiplicity, and edge magnitude (Figure 1; Appendix A.2). NOTEARS-style and DAGMA objectives exemplify this pattern [Zheng et al., 2018, Wei et al., 2020, Bello et al., 2022, Deng et al., 2023a]: matrix powers or resolvent entries encode walks, but a single scalar (trace, log-determinant, or spectral quantity) aggregates them. Familiar effects include **length dependence** through products along long cycles and coefficient schedules; **multiplicity** when the same simple cycle is counted at every multiple walk length (most acute for the log-det barrier near its domain boundary); and **magnitude dominance** when penalties act on weighted entries so large coefficient products steer gradients. DAGMA's inverse matrix provides a weighted walk-sum closure within its spectral domain, but its published acyclicity objective does not directly supervise individual closure entries—they could, in principle, feed a new pairwise loss, but DAGMA uses them only through the barrier gradient. SMNC instead constructs gate-based, scale-normalized relation scores and uses diagonal and optional off-diagonal entries as explicit supervision targets.

Our main conceptual contributions are threefold. **First**, *SMNC* exposes explicit matrix-valued structural relation scores: instead of using closure information only implicitly through the derivative of a scalar acyclicity function, it computes soft multi-hop reachability logits on a structural gate G—one score per ordered pair—for diagonal cycle suppression and optional off-diagonal relation supervision (Section 3.1). **Second**, SMNC normalizes path scores: it propagates scores in log-gate space by repeated dyadic squaring ($\lceil \log_2 d \rceil$ scales cover paths up to length d) and divides each scale by its path-length budget before readout, counteracting product-style attenuation on sufficiently strong long cycles; in the low-temperature limit a soft max-plus scan emphasizes strongest paths rather than pooling every walk length into one sum. **Third**, we **separate structure from magnitude** so closure supervision acts on support-level gates while data fit, sparsity, and the DAGMA barrier remain on the effective adjacency A: in the linear model, $A = W_f(M) \odot G(Z)$ with bounded W_f; in the nonlinear extension, an induced gate $G^{\mathrm{nl}}_{ij} = a_{ij}/(a_{ij} + c)$ yields $a_{ij} \geq \alpha \Rightarrow G^{\mathrm{nl}}_{ij} \geq \alpha/(\alpha + c)$. The requirement is that strong effective edges imply sufficiently strong structural gates; bounded WG factorization is one sufficient mechanism in the linear setting, not a universal prerequisite for SMNC. Closure scores are meaningless on a raw weighted adjacency that entangles existence and signed effect size. We therefore keep scalar walk penalties on A and evaluate SMNC on G (or G^{nl}): data fit, ℓ_1 sparsity, and the log-determinant barrier follow DAGMA on A, while $\mathcal{L}_{\mathrm{cyc}}$ and optional $\mathcal{L}_{\mathrm{rel}}$ supervise normalized logits built from the gate.

Pairwise domain information usually reports multi-hop precedence (i reaches j along some directed path), not one-hop adjacency [Chevalley et al., 2025, Yang et al., 2022]. We therefore use such cues as targets for closure logits. Used as edge labels, they would encourage the model to place direct edges on transitively related pairs, an error we call transitive overshoot.

We evaluate two versions of the model. The cue-free version uses only the diagonal cycle term and produces all synthetic benchmark results (Table 1). The cue version also gives the off-diagonal closure logits pairwise reachability cues as closure targets. A cue $i \rightsquigarrow j$ constrains whether a directed path exists, which edge-level acyclicity penalties cannot express. On the Sachs data (Section 4.3), reference-filtered cues improved recovery. The benefit depends on cue accuracy, and noisier cues derived from data alone remain to be tested at scale.

Contributions. 1) We introduce SMNC, a matrix-valued relation head with explicit per-pair closure logits on a structural gate, based on differentiable soft semiring propagation in log-gate space. 2) We normalize dyadic path scores by scale budgets and establish a cycle-length-independent lower bound on normalized logits for sufficiently strong cycles (Theorem 3.2); under a unique-active-cycle assumption, the hard final-scale readout admits a polynomially scaled log-gate gradient contribution rather than exponential product-style decay (Corollary 3.3; scope in Section 3.2). 3) We separate structure from magnitude for closure supervision—bounded $A = W_f(M) \odot G(Z)$ in the linear model and an induced ratio gate G^{nl} in the nonlinear extension—while retaining a DAGMA-style scalar barrier on A. 4) We let off-diagonal logits take optional pairwise reachability cues as closure targets; on Sachs, reference-filtered cues improved recovery in a single-seed run. 5) We evaluate on unstandardized linear and nonlinear synthetic benchmarks and on Sachs, separating cue-free runs from oracle- or reference-filtered side information.

Reachability semantics and cue interface. Write $i \rightsquigarrow j$ if a directed path of length ≥ 1 exists and set $\bar{R}_{ij}(B) = \mathbf{1}\{i \rightsquigarrow j\}$. Then $\mathrm{diag}(\bar{R}(B)) = 0$ iff B is acyclic. For DAGs use strict $R(B) = \bar{R}(B) \circ (\mathbf{1} - I)$ so $i \prec j \Leftrightarrow R_{ij}(B) = 1$. Positive reachability labels constrain closure logits, not direct edges (avoid transitive-closure bias). Optional noisy cues are masked binary targets $(O^{(o)}, M^{(o)})$ on off-diagonals for Backward-Exclusion (BEx), Ancestry (Anc), and Incomparability (Inc) semantics, closest in spirit to ancestral and side-constraint approaches but implemented here as soft BCE losses on differentiable reachability logits rather than hard feasibility constraints [Chen et al., 2016, Li and van Beek, 2018, Ban et al., 2024, 2025].
Even when pairwise reachability agrees, DAGs can differ in the hop scale at which relations first appear, motivating scale-aware multi-scale closure readouts beyond binary ancestry.

2 Related Work

Classical causal discovery. Classical causal discovery offers several complementary families: constraint-based conditional-independence tests, score-based combinatorial search, hybrid procedures, and order/permutation formulations [Spirtes et al., 2000, Chickering, 2002, Heckerman et al., 1995, Tsamardinos et al., 2006, Gasse et al., 2014, Raskutti and Uhler, 2018, Solus et al., 2021, Heinze-Deml et al., 2018, Kitson et al., 2023, Squires et al., 2020, Huang et al., 2011, 2013]. These methods establish strong algorithmic and statistical foundations, but they typically rely on discrete search spaces whose exact optimization is difficult at scale. This motivates differentiable surrogates that retain global structure while enabling gradient-based learning.

Continuous DAG optimization. Continuous DAG optimization methods replace explicit DAG search with smooth objectives over weighted adjacencies [Zheng et al., 2018, Wei et al., 2020, Deng et al., 2023a, Bello et al., 2022]. Likelihood-centric variants such as GOLEM pursue alternative graphical-model assumptions [Ng et al., 2020]. COSMO and SDCD leave that weighted-walk penalty; the next paragraph states how [Massidda et al., 2024, Nazaret et al., 2024].

Walk-based acyclicity constraints. NOTEARS-style constraints are built from matrix functions whose powers encode weighted walks and closed walks on a weighted adjacency [Zheng et al., 2018, Wei et al., 2020]. DAGMA makes this perspective explicit and replaces trace-exponential or polynomial acyclicity functions by a log-determinant characterization, motivated in part by improved detection of large cycles and better-behaved gradients [Bello et al., 2022]. The inverse inside the barrier is also a closure: $(sI - B)^{-1} = s^{-1} \sum_{k \geq 0} (B/s)^k$, $B = A \circ A$, sums the weighted walks between every ordered pair whenever the spectral radius of B is below s, and the gradient on edge (i, j) is $2A_{ij}[(sI - B)^{-1}]_{ji}$, the weighted sum of the walks from j back to i that would close a cycle through that edge. DAGMA's acyclicity objective does not directly supervise individual resolvent entries as prediction targets: they enter only through the barrier gradient, sum walks of every length (Appendix A.2), and diverge as the spectral radius approaches s. SMNC instead builds gate-based, scale-normalized relation scores and supervises their diagonal and optional off-diagonal entries explicitly (Section 3.1), while still combining closure losses with a scalar DAGMA barrier on A. Scalar polynomial surrogates built from $\mathrm{tr}(B^\ell)$ recount the same simple cycle at every multiple walk length and impose length dependence through the coefficient schedule $\{c_\ell\}$ (Appendix A.2). SDCD traces training difficulties to instability in acyclicity constraints and motivates a more stable spectral constraint together with a two-stage sparse training procedure [Nazaret et al., 2024]; the constraint is a single scalar, and its gradient is supported only on the dominant strongly connected component. COSMO avoids evaluating a cubic acyclicity constraint by parameterizing a smooth

acyclic orientation from node priorities and annealing its temperature, while noting that orientation gradients can saturate when priority differences become large [Massidda et al., 2024].

Nonlinear gradient methods. Nonlinear and broader gradient-based formulations extend continuous discovery beyond linear SEM parameterizations. Representative examples include nonlinear NOTEARS, GraN-DAG, DAG-GNN, NoCurl, DCDI, SAM, and RL-based search [Zheng et al., 2020, Lachapelle et al., 2020, Yu et al., 2019, 2021, Brouillard et al., 2020, Kalainathan et al., 2022, Zhu et al., 2020]. In parallel, identifiability and convergence analyses for nonlinear or non-Gaussian settings clarify when such estimators can recover causal structure and where failure modes arise [Shimizu et al., 2006, Hoyer et al., 2008, Peters et al., 2014, Bühlmann et al., 2014, Peters et al., 2017, Ng et al., 2022, Deng et al., 2023b, 2024, Ng et al., 2024, Reisach et al., 2021].

Side information. Side-information and relation-constrained approaches incorporate intervention-derived pairwise relations, temporal order cues, or explicit structural constraints [Chevalley et al., 2025, Yang et al., 2022, Sun et al., 2023, Moraffah et al., 2021, Chen et al., 2016, Li and van Beek, 2018, Ban et al., 2024, 2025, Cao et al., 2018]. Related continuous and time-series frameworks similarly leverage order and pairwise structure [Pamfil et al., 2020, Runge et al., 2019, Raskutti and Uhler, 2018, Solus et al., 2021]. Our method also uses pairwise relations as supervision, but applies them to closure logits (multi-hop reachability) rather than to direct-edge labels.

3 Method

3.1 Factorized adjacency and closure-side relation heads

Factorized effective adjacency. Separating signed magnitudes $W_f(M)$ from a structural gate $G(Z)$ prevents attributing large $|A_{ij}|$ to vanishing gate mass, which would weaken penalties on graph support and decouple supervision from the closure branch (the G-side computation feeding the relation heads below, as opposed to the magnitude branch W_f).

$$A = (W_f(M) \circ G(Z)) \circ (\mathbf{1} - I), \quad W_f(M) = \beta \tanh(M), \quad G(Z) = \sigma(Z/\tau_g) \circ (\mathbf{1} - I), \quad (1)$$

Here $W_f(M)$ stores signed magnitudes, $G(Z)$ stores soft structural support with off-diagonal entries in $(0, 1)$ and zero diagonal, $M, Z \in \mathbb{R}^{d\times d}$, $\beta > 0$ is a magnitude bound, and $\tau_g > 0$ is a gate temperature; data fit, sparsity, and the DAGMA barrier are evaluated on A, while closure-side cycle and relation losses are applied to G. We abbreviate this factorized backbone as WG (bounded signed-magnitude branch W_f + structural gate G). Since $|W_{f,ij}| \le \beta$, any effective edge satisfying $|A_{ij}| \ge \alpha$, for $0 < \alpha \le \beta$, necessarily has $G_{ij} \ge \alpha/\beta$. Consequently, every directed cycle in the α-thresholded effective graph corresponds to a gate cycle whose edge weights are at least α/β. The converse need not hold: a large gate value does not guarantee a large effective edge weight, and when $W_{f,ij} = 0$ one has $|A_{ij}| = 0$ regardless of G_{ij}. The factorization therefore yields a one-way link from strong effective edges to gates, not a full correspondence between thresholded A and G. Positive reachability cues act on closure of G and need not be realized by corresponding paths in a thresholded effective graph; that mismatch is a limitation of the surrogate, not a flaw in the cycle lower bound (Theorem 3.2).

Following Section 1, relation heads act on the structural gate G and produce matrix-valued logits for self-reachability and off-diagonal reachability. The main construction defines the SMNC closure head (a specific relation head, instantiated below) used in our experiments.

Dyadic closure scan. Closure-side training needs matrix-valued reachability summaries on the structural gate G: diagonal logits encode self-reachability and directed cycles (for suppression), while optional off-diagonal logits support masked reachability-style BCE, addressing relation targets such as ancestry that scalar acyclicity penalties on the effective adjacency A in Eq. (1) cannot capture alone. Our dyadic states $U_{ij}^{(k)}$ supply these logits in softened form: computed from $\log G$, each aggregates multi-hop connectivity by scoring the *best* directed explanation from i to j among routes of length at most 2^k, complementing the edge-level parameterization on A above. Naively materializing all walk lengths for all pairs does not scale; we maintain a compact sufficient-statistics stack $\{U^{(k)}\}$ indexed by dyadic path-length budgets, with repeated doubling instead of explicit path enumeration. Classical directed reachability is packaged by repeated matrix composition in the Boolean semiring $(\vee, \wedge)$ [Warshall, 1962], and graph path problems more generally admit semiring/dioid formulations [Gondran and Minoux, 2008].

Training pairs logits S_{ij} with bounded scores $P_{ij} = \sigma(S_{ij})$ (concretely P_{ij}^{un} and $P_{ij}^{(\gamma)}(\Omega)$ in the SMNC readout below; BCE supervision in Section 3.1). Along a directed route $\pi : i \rightsquigarrow j$, whole-

route openness is $\prod_{e\in\pi} G_e$; with log-gates $E_{ij} = \log G_{ij}$, $\sum_{e\in\pi} E_e$ turns products into sums, so best-route summaries are *max–sum* in E-space (equivalently max–*product* on gates). We approximate $\max_{\pi:\, i\rightsquigarrow j,\, |\pi|\le L} \sum_{e\in\pi} E_e$ for all pairs and budgets without enumerating paths using the dyadic scan in Eq. (2); Theorem 3.1 records the exact hard-$(\vee, \otimes)$ semantics. Readouts map the resulting stack $\{U^{(k)}\}$ to the logits S_{ij} used in training (e.g. $\Lambda_{ij} = U_{ij}^{(K)}$ in Eq. (4) and the SMNC readouts below). Formally, the log-gates are $E_{ij} = \log G_{ij}$ for $i \neq j$ and $E_{ii} = -\infty$, and we set $K = \lceil \log_2 d \rceil$. Thus explicit self-edges are forbidden, while self-reachability may still arise through multi-step directed cycles. The hard $(\max, +)$ dyadic scan is repeated squaring in the max-plus semiring, a standard path-algebraic construction [Gondran and Minoux, 2008]:

$$U^{(0)} = E, \qquad U^{(k+1)} = U^{(k)} \vee \big(U^{(k)} \otimes U^{(k)}\big), \qquad k = 0, \dots, K-1, \tag{2}$$

where $(a \vee b) = \max\{a, b\}$ and $(X \otimes Y)_{ij} = \max_m (X_{im} + Y_{mj})$. For training, hard maxima are replaced by log-sum-exp surrogates at closure temperature $\tau_c > 0$, preserving the dyadic doubling pattern of Eq. (2) while routing gradients through competing path explanations. We use soft max-plus (Eq. (3)) for autodiff; concretely, $a \vee_{\tau_c} b = \tau_c \log\big(\exp(a/\tau_c) + \exp(b/\tau_c)\big)$ and $(X \otimes_{\tau_c} Y)_{ij} = \tau_c \log \sum_m \exp\big((X_{im} + Y_{mj})/\tau_c\big)$. Training implements each composition as a shifted exponential-space GEMM in `float32` (linear) or a clamped approximate GEMM (nonlinear); these match the exact scan only where finite precision preserves the products (Appendices C.4 and A.5). The semiring semantics and gradient envelopes below concern exact arithmetic unless stated otherwise. The same primitives extend to general (non-dyadic) length budgets in two ways. One composes additional segment max-plus products with step sizes other than pure doubling, which needs more rounds than the $O(\log d)$ depth of repeated squaring. The other pads the cap upward to the next dyadic tier.

$$U^{(0)} = E, \qquad U^{(k+1)} = U^{(k)} \vee_{\tau_c} \big(U^{(k)} \otimes_{\tau_c} U^{(k)}\big), \qquad k = 0, \dots, K-1. \tag{3}$$

Theorem 3.1 (Exact semantics of the hard dyadic scan)**.** *For every $k \ge 0$, $U_{ij}^{(k)} = \max\{\sum_{e\in\pi} E_e :$ π a directed walk from i to j, $1 \le |\pi| \le 2^k\}$ ($\max \emptyset = -\infty$). Because $E \le 0$, whenever the walk set is nonempty the maximum is attained by a simple path if $i \neq j$ and by a simple cycle through i if $i = j$.*

The proof is given in Appendix C.2.

In the hard scan of Eq. (2), $U^{(0)} = E$ stores one-hop $\log G$ scores. The product $U^{(0)} \otimes U^{(0)}$ maximizes over intermediate m and thus collects the best concatenation of two length-≤ 1 segments, i.e. the best length-≤ 2 path in log-space; $U^{(1)} = U^{(0)} \vee (U^{(0)} \otimes U^{(0)})$ keeps whichever one-hop or two-hop route is stronger. Inductively, $U^{(k)}$ carries the best score among paths of length at most 2^k. Only $K = \lceil \log_2 d \rceil$ repeated-squaring rounds are needed to reach budgets $2^K \ge d - 1$ along simple routes, hence $O(\log d)$ depth and no path enumeration.

Each update merges a best length-$\le 2^k$ explanation from i to an intermediate m with a best length-$\le 2^k$ explanation from m to j, then maximizes over splits m. Standard matrix multiplication computes $\sum_m (A_{im} B_{mj})$. By operating in the $(\max, +)$ (max-plus) semiring, we simply replace addition with maximization and multiplication with addition, computing $\max_m (A_{im} + B_{mj})$. In the hard ($\tau_c \to 0$) limit this repeated squaring isolates the single strongest route; the smooth version used for training is discussed below.

All relation heads share the states $U^{(k)}$ of the soft scan in Eq. (3) and differ only in their readout, the map from $\{U^{(k)}\}_{k=0}^{K}$ to relation logits.

SMNC: scale-normalized multi-scale closure head. The simplest dyadic readout maps the final raw dyadic closure state to relation logits,

$$\Lambda_{ij} := U_{ij}^{(K)}, \qquad P_{ij}^{\mathrm{un}} := \sigma(\Lambda_{ij}). \tag{4}$$

This baseline preserves the structure/magnitude separation and closure-side supervision principle, but it does not scale-normalize path scores. On a unique length-L path with constant gate value $p \in (0, 1)$, the dominant raw logit scales as $L \log p$.

The name *Scale-Normalized Multi-Scale Closure* (SMNC) refers to (i) *multi-scale* states $U^{(k)}$ read at several dyadic scales k, and (ii) *scale normalization* at each scale: $\widetilde{U}^{(k)} = U^{(k)}/2^{\gamma k}$ divides accumulated log-path scores by the length budget 2^k (up to $\gamma > 0$), counteracting shrinkage of product-form scores along long routes and the associated vanishing BCE gradients (Remark C.3).

$$\widetilde{U}^{(k)} = \frac{U^{(k)}}{2^{\gamma k}}, \qquad \gamma > 0. \tag{5}$$

For a selected scale set $\Omega \subseteq \{0, 1, \ldots, K\}$, the smooth readout used in practice is

$$S_{ij}^{(\gamma)}(\Omega) = \begin{cases} \widetilde{U}_{ij}^{(k)}, & \Omega = \{k\}, \\ \tau_s \log \sum_{k \in \Omega} \exp\big(\widetilde{U}_{ij}^{(k)}/\tau_s\big), & |\Omega| > 1, \end{cases} \qquad \tau_s > 0. \tag{6}$$

The bounded relation score is $P_{ij}^{(\gamma)}(\Omega) = \sigma\Big(S_{ij}^{(\gamma)}(\Omega)\Big)$. We do not interpret $P_{ij}^{(\gamma)}(\Omega)$ as a calibrated reachability probability. The BCE-with-logits losses below use $S_{ij}^{(\gamma)}(\Omega)$ as a margin-style surrogate logit for cycle suppression and optional relation supervision.

We use the scale families (fixed subsets Ω of dyadic scales k read out jointly via Eq. (6)) $\Omega_{\mathrm{dir}} = \{0\}$, $\Omega_{\le 2} = \{0, 1\}$, $\Omega_{\le 4} = \{0, 1, 2\}$, $\Omega_{\mathrm{final}} = \{K\}$, $\Omega_{\mathrm{all}} = \{0, 1, \ldots, K\}$. The linear cycle readout Ω_{final} divides by the graph-wide dyadic budget $2^K = \Theta(d)$, not by an active cycle's length L, so it is not cycle-adaptive path-length normalization (Corollary 3.3's $\Theta(1/L)$ factor arises only at the minimal covering scale k_L; hard all-scale maximization selects the coarsest scale K by Lemma C.1). These sets expose cumulative scale-normalized relation profiles from the same structural representation. The terms direct, final-scale, and all-scale are shorthand for Ω_{dir}, Ω_{final}, and Ω_{all}. Because $U^{(k)}$ stores paths of length at most 2^k, a shorter path may also contribute to larger-scale readouts, so no family gives exact hop-bin probabilities. Exact hop-bin supervision would require residual or bin-specific readouts, which we leave to future work.

The smooth scan is itself a tempered walk sum, $\exp(U^{(k)}/\tau_c) = \sum_{\ell \le 2^k} c_{k,\ell} (G^{\circ 1/\tau_c})^{\ell}$ with fixed integers $c_{k,\ell} \ge 1$, and $0 \le U_{\mathrm{soft}}^{(k)} - U_{\mathrm{hard}}^{(k)} \le (2^k - 1)\tau_c \log(2d)$ (Lemma C.2). The soft scan therefore still aggregates walks, but with weights: each extra traversal of a cycle with gate product Q is discounted by Q^{1/τ_c} and stays in the sum. For short cycles with gates near 1, multiplicity can return. For a 2-cycle with gates 0.9 at $\tau_c = 0.1$ and $d = 128$, it multiplies the gradient about 12-fold (Appendix C.5). We use BCE-with-logits for diagonal cycle suppression and optional off-diagonal relation cues. In the linear SMNC model used in our experiments, the cycle logit is the final-scale normalized readout $S^{\mathrm{cyc}} = S^{(\gamma)}(\Omega_{\mathrm{final}})$, $\Omega_{\mathrm{final}} = \{K\}$; the nonlinear variant uses Ω_{all} (Section 3.2). The cycle target is zero self-reachability on the diagonal. Optional off-diagonal cue families, when used, apply masked BCE to $S_{ij}^{(\gamma)}(\Omega_o)$ with family-specific masks, targets, and weights. We use BCE as a margin-style relation surrogate rather than a calibrated probability model: hard $(\max, +)$ closure logits are nonpositive, so even an ideally open positive relation incurs loss at least $\log 2$. The length-attenuation analysis in Section 3.2 concerns zero-target cycle suppression, where $\partial\,\mathrm{softplus}(s)/\partial s = \sigma(s)$; positive ancestry targets use $\mathrm{softplus}(-s)$ with $\partial \ell/\partial s = \sigma(s) - 1$, which does not share the same vanishing behavior as $s \to -\infty$.

3.2 Length behavior of the scale-normalized head

The results in this section are local and concern a fixed gate matrix and hard $(\max, +)$ readouts unless noted. Theorem 3.2 is a *score* lower bound: it links thresholded cycles in A to normalized diagonal logits on G, but does not assert convergence of optimization or gradients along a training trajectory. Corollary 3.3 bounds an active-cycle gradient contribution in $\log g$ under a unique-active-cycle assumption; it is not an unconditional guarantee for the finite-τ_c soft scan or for learned parameters Z. We analyze the hard readout $S_{ij}^{(\gamma),\max}(\Omega) = \max_{k \in \Omega} U_{ij}^{(k)}/2^{\gamma k}$, with $U^{(k)}$ from Eq. (2). In exact arithmetic the smooth recursion and scale aggregation dominate their hard counterparts, so hard lower bounds carry over. The archived linear trainer evaluates the soft scan with row/column-shifted `float32` GEMMs; that implementation can zero out logits whose best compositions lie far below the shifts, invalidating the cycle bounds and gradients on affected instances (Appendix C.4). The nonlinear trainer uses a further clamped GEMM (Section 3.3); it is an approximate implementation to which the hard-scan and exact soft-scan theorems do not apply directly.

Theorem 3.2 (Length-stable cycle lower bound at $\gamma = 1$)**.** *Assume the bounded backbone in Eq.* (1)*. Suppose the thresholded effective graph contains a directed cycle C of length L, and every edge $e \in C$ satisfies $|A_e| \ge \alpha$, with $0 < \alpha \le \beta$. Let $k_L := \lceil \log_2 L \rceil$ and $K = \lceil \log_2 d \rceil$. Then every node v on the cycle satisfies $\frac{U_{vv}^{(k_L)}}{2^{k_L}} \ge \log(\alpha/\beta)$, $\frac{U_{vv}^{(K)}}{2^K} \ge \log(\alpha/\beta)$. Consequently, for both readouts the normalized self-reachability logit of a thresholded cycle is bounded below independently of L, whereas the raw logit $U_{vv}^{(K)}$ is only bounded below by $L \log(\alpha/\beta)$.*

The proof is given in Appendix C.3.

Corollary 3.3 (Scale-specific polynomial gradient factor)**.** *Assume the hard normalized readout with $\gamma = 1$. Let C be a cycle candidate of length L, and consider a selected dyadic scale a such*

that the cycle candidate is feasible at that scale. Define $S_{vv,a} := \frac{U^{(a)}_{vv}}{2^a}$. *If the cycle candidate is the unique maximizer in* $U^{(a)}_{vv}$, *then for every edge* e *on this active candidate,* $\frac{\partial S_{vv,a}}{\partial \log g_e} = 2^{-a}$. *Consequently, for the zero-target BCE cycle loss* $\ell^{\mathrm{cyc}}_{v,a} = \mathrm{softplus}(S_{vv,a})$, $\frac{\partial \ell^{\mathrm{cyc}}_{v,a}}{\partial \log g_e} = 2^{-a}\sigma(S_{vv,a})$. *At the minimal covering scale* $a = k_L = \lceil \log_2 L \rceil$, *this gives a* $\Theta(1/L)$ *scale factor. At the final scale* $a = K = \lceil \log_2 d \rceil$, *which is the readout used by the linear cycle loss, the factor is* $\Theta(1/d)$.

These bounds concern local derivatives with respect to log-gates; gradients in the trained gate logits additionally include the sigmoid Jacobian $(1 - g_e)/\tau_g$ and need not obey the same lower bound (Appendix C.5). Proofs and a discussion of active scales in all-scale readouts are given in Appendix C.5.

Lemma C.1 (stated and proved in Appendix C.5) gives coarsest-scale dominance of the hard readout: for any $\gamma > 0$, $U^{(a)}_{ij}/2^{\gamma a}$ is non-decreasing in a, so the hard Ω_{all} and Ω_{final} readouts coincide. Off-diagonal soft normalized scores need not be monotone in a (Lemma C.2), whereas exact soft diagonal states satisfy the same non-decreasing property at $\gamma = 1$ (Appendix C.5). Smooth scale aggregation ($\tau_s > 0$) spreads gradient over finer scales.

Objective used in our experiments. Linear benchmarks use $\mathcal{L}_{\mathrm{cyc}} = \frac{1}{d}\sum_v \mathrm{softplus}(U^{(K)}_{vv}/2^K)$ (Ω_{final}, $\gamma = 1$, $\tau_c = 0.1$); nonlinear ones use the soft $S^{(1)}(\Omega_{\mathrm{all}})$, $\tau_c = \tau_s = 0.1$ (with a clamped GEMM, Appendix A.5). For a thresholded length-L cycle C that uniquely maximizes $U^{(K)}_{vv}$, $v \in C$, Corollary 3.3 and Theorem 3.2 give, in the hard limit, a contribution of the C-terms to $\partial\mathcal{L}_{\mathrm{cyc}}/\partial \log g_e$ ($e \in C$) in $[L\sigma(\log(\alpha/\beta))/(2d^2),\ L/(2d^2)]$, a lower envelope proportional to L/d^2 at fixed d and active-gate floor; the realized per-edge contribution need not grow monotonically with L (on a uniform-gate cycle, the hard final-scale factor $L/(dD)\,\sigma(L\log p/D)$, $D = 2^K$, can decrease when $p < 1$). The unnormalized head gives at most $(L/d)\,\sigma(L\log p_{\max})$, $p_{\max} := \max_{e\in C} g_e$ (Appendix C.5). The $\Theta(1/L)$ rate holds at the minimal covering scale k_L, while the linear objective reads out the final scale K with factor $\Theta(1/d)$. The bound does not cover the finite-τ_c scan exactly, `float32` shifted-GEMM training, or the nonlinear clamped head.

3.3 Cycle-regularized continuous DAG learning

For linear Gaussian SEMs,

$$\mathrm{score}(A) = \frac{1}{2n}\|X - XA\|_F^2.$$

$$h_s(A) = d\log s - \log\det\big(sI - A\circ A\big). \tag{7}$$

$$\min_{M,Z}\ \mu\big(\mathrm{score}(A) + \lambda_1\|A\|_1\big) + h_s(A) + \mu_{\mathrm{cycle}}\mathcal{L}_{\mathrm{cyc}} + \mu_{\mathrm{rel}}\mathcal{L}_{\mathrm{rel}}. \tag{8}$$

Setting $\mu_{\mathrm{rel}} = 0$ yields the cue-free closure-regularized model; $\mu_{\mathrm{cycle}} = \mu_{\mathrm{rel}} = 0$ is the pure bounded-WG backbone. In the cue-free model used for the main benchmarks, $\mu_{\mathrm{rel}} = 0$ and the diagonal cycle term uses the SMNC final-scale readout $\Omega_{\mathrm{final}} = \{K\}$ in the linear solver (the nonlinear variant below uses Ω_{all} with $\tau_s = 0.1$).

Nonlinear induced-structure variant. A decoder-induced nonnegative A_{mag} is mapped to probabilities $p = \phi(A_{\mathrm{mag}}) \circ (\mathbf{1} - I)$ with $p_{ij} \in [0, 1]$; the same closure readouts and $\mathcal{L}_{\mathrm{cyc}}, \mathcal{L}_{\mathrm{rel}}$ are applied to p. This variant uses a *clamped approximate* soft scan: exponents are clipped before multiplication and products are floored, which can inject spurious path evidence and zero gradients on genuinely weak compositions (Appendix A.5); nonlinear benchmark numbers should be read as empirical outcomes of this head, not as validation of the exact theorems.

4 Experiments

4.1 Toy cyclic examples

Before the large-scale benchmarks, we run *local probes*: fixed small graphs, no data likelihood, and no search over hyperparameters. The goal is to separate two questions that scalar walk penalties blur together—whether a penalty still *senses* a long cycle in its gradient, and whether gradient steps can actually *open* that cycle during optimization.

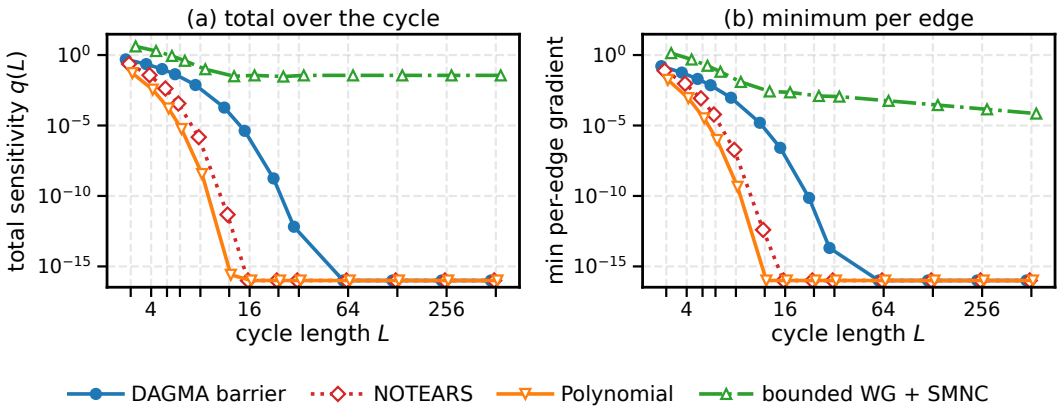

Figure 2: **Gradient signal on symmetric long cycles.** Matched directed L-cycles $X_L(\rho) = \rho B_L$ at $\rho = 0.6$, without optimization, for the walk-sum penalties, namely the DAGMA log-det barrier, the NOTEARS trace-exponential and the DAG-GNN polynomial $\mathrm{tr}[(I + A \circ A/d)^d] - d$ (all on $A = X_L(\rho)$), and for the bounded-WG induced A-space signal of the DAGMA barrier plus the SMNC final-scale cycle loss ("bounded WG+SMNC", on $G = X_L(\rho)$, $\mu_{\text{cycle}} = 1$, $\tau_g = 1$; the barrier dominates for $L \leq 8$). Each cycle is the whole graph ($d = L$). (a) Total active-edge sensitivity $q(L) = \left|\sum_{e \in C_L} \nabla_e\right|$. (b) Minimum per-edge gradient $\min_{e \in C_L} |\nabla_e|$, which equals $q(L)/L$ because every edge of a symmetric cycle receives the same gradient. Values below 10^{-16} are drawn at that floor (Appendix A.1).

Symmetric long cycles. We place a single directed L-cycle with equal edge strength $\rho = 0.6$ and measure how much decycling signal each objective sends to the cycle edges (Figure 2; probe setup in Appendix A.1). Walk-sum penalties (DAGMA, NOTEARS, DAG-GNN polynomial) lose total sensitivity $q(L)$ rapidly as L grows (log–log slopes -76.28, -22.73, -88.03). Bounded WG+SMNC decays far more slowly overall (slope -0.78), because it still contains the DAGMA barrier; the SMNC cycle term by itself is nearly flat in L on this isolated-cycle setup. Spectral penalties such as SDCD are not plotted; the SDCD comparison is in Appendix B.

Controlled single- and two-cycle optimization. Figure 2 evaluates gradients at a fixed graph. The next probe runs short full-gradient optimizations on toy cyclic supports: only cycle edges are free, the adjacency is $A = G$, and the objective is a fidelity term plus DAGMA's log-det barrier and either a raw or an SMNC cycle loss (no data term). We mark a cycle as broken when any gate on it falls below 0.1; setup and hyperparameters are in Appendix A.3.

We first match the *initial* gradient on the closing edge at $L = 4$, then freeze those cycle weights and vary L. Under that calibration, SMNC maintains stronger pressure on long cycles than the unnormalized head as L grows. At one fixed pair of weights for both heads, the unnormalized head clears short single cycles ($L \leq 8$) within 60 updates but does not break any $L \geq 16$ cycle within 500 updates. SMNC breaks $L \in \{16, 24, 32\}$, and does so in fewer updates as L increases (Figure 3), while retaining most path-gate mass on the surviving edges.

We then embed a short cycle ($L_s = 8$) and a long one on the same support with one shared cycle weight. SMNC breaks the long cycle first in every run, but the short cycle lags: joint removal of both cycles often waits until the weaker short cycle finally opens. An unnormalized closure can finish both cycles faster only when the long cycle is relatively short ($L_\ell = 24$) and never clears the longest cycles within the tested budget, even with up to 5000 updates. Appendix A.3 also reports sufficient cycle weights on isolated cycles and an s-continuation control that weakens the log-det barrier stage by stage.

4.2 Synthetic Gaussian benchmarks

Models and data. We report two cue-free variants: the explicit bounded-WG linear solver (Section 3.1) and the nonlinear induced-structure extension (Section 3.3). Data are drawn from Erdős–Rényi DAGs with $d \in \{10, 15, 20\} \cup \{30, 40, \ldots, 100\}$ and $s_0 = \kappa d$ for $\kappa \in \{2, 4, 5, 6, 7, 8\}$ when feasible. For each (d, s_0) pair we sample $N_{\text{rep}} = 5$ independent datasets of $n = 10{,}000$ observational points. Linear SEMs use edge weights from $\mathrm{Unif}([-1.5, -0.5] \cup [0.5, 1.5])$ and homoscedastic Gaussian noise with $\sigma = 0.6$. Nonlinear SEMs use width-100 one-hidden-layer MLP mechanisms

Table 1: Cue-free RAW results at $d = 100$. Each cell reports mean $\pm$ sample standard deviation over 5 datasets.

	Gaussian linear SEMs				Gaussian nonlinear SEMs			
	bounded WG + SMNC		DAGMA		induced + SMNC		DAGMA	
ER	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑
2	2.20 ± 3.19	$\mathbf{0.99 \pm 0.01}$	$\mathbf{2.00 \pm 3.08}$	$\mathbf{0.99 \pm 0.01}$	69.80 ± 23.86	$\mathbf{0.83 \pm 0.05}$	$\mathbf{68.80 \pm 23.53}$	0.82 ± 0.06
4	$\mathbf{22.80 \pm 17.05}$	$\mathbf{0.97 \pm 0.02}$	48.00 ± 41.36	0.94 ± 0.05	$\mathbf{85.40 \pm 11.28}$	$\mathbf{0.89 \pm 0.01}$	116.60 ± 8.08	0.84 ± 0.01
6	$\mathbf{20.80 \pm 15.01}$	$\mathbf{0.98 \pm 0.01}$	88.20 ± 47.33	0.93 ± 0.04	$\mathbf{127.00 \pm 23.40}$	$\mathbf{0.89 \pm 0.02}$	204.20 ± 6.02	0.80 ± 0.01
7	$\mathbf{58.80 \pm 22.17}$	$\mathbf{0.96 \pm 0.01}$	102.20 ± 47.01	0.93 ± 0.03	$\mathbf{173.60 \pm 28.08}$	$\mathbf{0.87 \pm 0.02}$	290.60 ± 12.66	0.75 ± 0.02
8	$\mathbf{248.40 \pm 103.58}$	$\mathbf{0.84 \pm 0.06}$	376.80 ± 206.54	0.78 ± 0.11	$\mathbf{247.00 \pm 36.86}$	$\mathbf{0.82 \pm 0.03}$	390.40 ± 27.61	0.69 ± 0.03

in topological order. Following the usual NOTEARS-style benchmark protocol, inputs are not standardized (Limitations).

Cue-free training setup. Main benchmark rows use diagonal closure-side cycle regularization only: no off-diagonal relation BCE and no hard reverse-edge mask. Linear runs read out $\Omega_{\text{final}} = \{K\}$ via shifted `float32` GEMM; nonlinear runs use the soft all-scale readout Ω_{all} at $\tau_s = 0.1$ with the clamped GEMM (Sections 3.1 and 3.2). Reported linear gains are therefore under the documented numerical closure, not the idealized exact scan assumed in Section 3.2.

Baselines and reporting. We compare against DAGMA, NOTEARS, GOLEM, COSMO, and SDCD. The main text emphasizes DAGMA as the closest single-adjacency continuous baseline; NOTEARS (DAGrad's default Adam run) is consistently worse than both DAGMA and SMNC in all 61 (d, s_0) settings, under RAW and TOPK and on both SEM types. Full $d = 100$ tables and across-d curves appear in Appendix D.1; baseline budgets and hyperparameters were not matched to ours and are disclosed in Appendix D.2.

Metrics and optimizer schedule. We score structure recovery on the learned effective adjacency A, not on closure logits, using pair-space SHD and directed edgewise SHD (dSHD); reversing an edge counts as two edits. We report both **RAW** thresholded outputs and **TOPK** outputs that keep up to s_0 largest-magnitude edges (a ranking diagnostic with known sparsity). Table 1 gives RAW summaries at $d = 100$; TOPK and across-d curves are in Appendix D.1. For the linear bounded-WG solver we use a DAGMA-style outer loop with $N_{\text{outer}} = 5$, $\mu_{\text{init}} = 10$, stagewise $s \in \{1.0, 0.9, 0.8, 0.7, 0.6\}$, $\lambda_1 = 10^{-2}$, $\tau_g = 1$, magnitude threshold 0.1, and 10,000 or 20,000 inner steps per stage (depending on d and ER). The backbone runs in `float64` and the closure head in `float32` (shifted GEMM; Appendix C.4). Archived reruns reproduce every reported cell to within one standard deviation (Appendix D.2).

Cue-free $d = 100$ results.

Linear SEMs. On linear Gaussian SEMs, Table 1 shows bounded WG + SMNC competitive with or better than DAGMA in denser regimes, reducing dSHD from 48.0 to 22.8 at ER=4, from 88.2 to 20.8 at ER=6, from 102.2 to 58.8 at ER=7, and from 376.8 to 248.4 at ER=8, while maintaining high directed F1. At ER=2, all methods are close under RAW evaluation.

Nonlinear SEMs. On nonlinear Gaussian SEMs, Table 1 shows the induced-structure variant with the same pattern: it improves over DAGMA in denser regimes (e.g. dSHD 116.6 vs. 85.4 at ER=4, 390.4 vs. 247.0 at ER=8) and matches DAGMA closely at ER=2. Our nonlinear solver also differs from the DAGMA baseline outside the cycle term (wider MLP, extra outer stage, different λ_1 and μ_{init}), so this comparison does not isolate SMNC alone. Table 11 reports a nonlinear ablation of hard exclusion and soft reachability supervision using oracle ancestry cues (Appendix D.4).

4.3 Real-data validation: Sachs

We evaluate on the Sachs protein-signaling data [Sachs et al., 2005] against the standard 17-edge reference DAG distributed with `bnlearn` (URL in Appendix D.3), a literature benchmark and not an absolute biological ground truth. We report single-seed (seed$= 0$) RAW recovery at output threshold 0.3 on 11,672 pooled, column-wise z-scored samples from all 14 conditions. The first row of Table 2 is the internal nonlinear backbone with the SMNC cycle term disabled; all other rows keep cycle regularization. The cue-assisted rows share 12 ancestry relations extracted from interventional shifts and filtered with the reference DAG (Appendix D.3). They compare hard reverse-edge exclusion alone, positive-only soft closure supervision without a hard mask, and hard exclusion with two-sided soft supervision (positive forward and zero reverse targets).

Table 2: Sachs RAW recovery against the 17-edge reference DAG, using seed 0 and output threshold $t = 0.3$. The first row is the internal nonlinear backbone with the cycle term disabled, not a separately executed official DAGMA baseline. The target column lists the numbers of active positive and negative soft closure targets; "–" denotes no active soft loss.

Configuration	Hard mask	Soft targets pos./neg.	dSHD ↓	F1-dir ↑
Nonlinear backbone (cycle off)	No	–	24	0.250
Induced + SMNC, no cues	No	–	14	0.533
SMNC + hard exclusion	Yes	–	14	0.500
SMNC + positive-only soft cues	No	$12/0$	10	0.722
SMNC + hard + two-sided soft cues	Yes	$12/12$	**6**	**0.833**

Enabling the cycle term lowers dSHD from 24 to 14. Hard exclusion alone leaves dSHD at 14 and lowers F1-dir from 0.533 to 0.500, positive-only soft supervision reaches dSHD 10 and F1-dir 0.722 without hard exclusions, and hard plus two-sided soft supervision reaches dSHD 6 and F1-dir 0.833. These single-seed results suggest that soft reachability supervision can complement relations the cue-free model does not reliably recover. Positive-only and hard plus two-sided supervision also differ in reverse targets and per-target weighting, so the comparison does not isolate the effect of a hard mask.

Limitations

Numerical closure. Section 3.2 analyzes an exact soft scan and hard limits. Linear training uses shifted exponential `float32` GEMMs that can underflow cycle logits (Appendix C.4), and nonlinear training uses a clamped GEMM with artificial path evidence on acyclic supports (Appendix A.5). The nonlinear barrier guard can accept points outside the M-matrix domain (Appendix D.2). The reported benchmarks were produced with these heads; exact gradient and score lower bounds should not be interpreted as guarantees on the training dynamics without a validated stable implementation.
SMNC's gradient-scaling results are local active-candidate statements (Section 3.2): the $\Theta(1/L)$ factor holds only at the minimal covering scale, while the final-scale readout $\Omega_{\text{final}} = \{K\}$ used in our cue-free linear runs (like the hard all-scale readout, Lemma C.1) weights a length-L cycle by the length-independent factor $2^{-K} = \Theta(1/d)$ rather than 2^{-k_L}; our nonlinear runs use the soft all-scale readout ($\tau_s = 0.1$), which mixes these factors across scales.
Empirical gains are demonstrated primarily on synthetic Gaussian linear and nonlinear SEM benchmarks, with main-text summaries emphasizing RAW evaluation and TOPK ranking diagnostics reported in the appendix. As in most observational DAG-learning work, causal interpretation depends on assumptions such as causal sufficiency, faithfulness or model-specific identifiability, and appropriate benchmark design [Spirtes et al., 2000, Peters et al., 2017, Heinze-Deml et al., 2018, Reisach et al., 2021]. Soft positive reachability cues constrain path-level relations and generally do not identify a unique realizing edge. In the exploratory $d = 100$ nonlinear ablation, gains from soft supervision vary with graph regime, preprocessing, and output threshold (Appendix D.4). Their benefit may also depend on cue accuracy. The Sachs cues were filtered with the reference DAG. The bounded WG + SMNC pipeline introduces additional hyperparameters (including β, τ_g, and closure-loss weights); broader robustness to graph families, sampling regimes, and hyperparameter choices remains open.
The dyadic closure scan needs $K = \lceil \log_2 d \rceil$ dense $d \times d$ products, so as d grows it costs more time and memory than a power-iteration spectral penalty. Non-closure penalties such as SDCD's are cheaper per step; our unstandardized benchmarks, which depart from SDCD's z-scoring protocol (Appendix D.2), do not test that regime (comparison in Appendix B).

Impact Statement

SMNC is a scale-normalized multi-scale relation head designed to reduce product-style length bias in closure logits. Like other causal discovery methods, it produces hypotheses: inferred graphs should be combined with domain expertise, robustness checks and, when possible, interventional validation before high-stakes use.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the National Science Foundation under Grant No. 2450946. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This research is being performed using funding received from the U.S. Department of Energy Office of Nuclear Energy's Nuclear Energy University Programs under Award Number DE-NE0009383. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
The authors declare no competing interests.

References

Taiyu Ban, Lyuzhou Chen, Xiangyu Wang, Xin Wang, Derui Lyu, and Huanhuan Chen. Differentiable structure learning with partial orders. In *Advances in Neural Information Processing Systems*, volume 37, pages 117426–117455. Curran Associates, Inc., 2024. doi: 10.52202/079017-3728. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d51cd79a85833b022841f7a2383b32d3-Abstract-Conference.html.

Taiyu Ban, Changxin Rong, Xiangyu Wang, Lyuzhou Chen, Xin Wang, Derui Lyu, Qinrui Zhu, and Huanhuan Chen. Differentiable structure learning with ancestral constraints. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=fyLsLTi1rE.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a log-determinant acyclicity characterization. In *Advances in Neural Information Processing Systems*, volume 35, pages 8226–8239. Curran Associates, Inc., 2022. doi: 10.52202/068431-0598. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/36e2967f87c3362e37cf988781a887ad-Abstract-Conference.html.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f8b7aa3a0d349d9562b424160ad18612-Abstract.html.

Peter Bühlmann, Jonas Peters, and Jan Ernest. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014. doi: 10.1214/14-AOS1260.

Xiao Cao, Yan Jin, Ji Liu, Bo Zeng, and Shuai Huang. Optimal expert knowledge elicitation for Bayesian Network structure identification. *IEEE Transactions on Automation Science and Engineering*, 15(3):1163–1177, 2018. doi: 10.1109/TASE.2017.2747130.

Eunice Yuh-Jie Chen, Yujia Shen, Arthur Choi, and Adnan Darwiche. Learning Bayesian networks with ancestral constraints. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper/6386-learning-bayesian-networks-with-ancestral-constraints.

Mathieu Chevalley, Patrick Schwab, and Arash Mehrjou. Deriving causal order from single-variable interventions: Guarantees & algorithm, 2025. URL https://arxiv.org/abs/2405.18314.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*, Lecture Notes in Statistics, pages 121–130. Springer, 1996. doi: 10.1007/978-1-4612-2404-4_12.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002. URL https://jmlr.org/papers/v3/chickering02b.html.

David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004. URL https://jmlr.org/papers/v5/chickering04a.html.

Chang Deng, Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. Optimizing notears objectives via topological swaps, 2023a. URL https://arxiv.org/abs/2305.17277.

Chang Deng, Kevin Bello, Pradeep Ravikumar, and Bryon Aragam. Global optimality in bivariate gradient-based DAG learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://papers.neurips.cc/paper_files/paper/2023/hash/3a02b6df276223b68c69ca572cb3c4a8-Abstract-Conference.html.

Chang Deng, Kevin Bello, Pradeep Ravikumar, and Bryon Aragam. Markov equivalence and consistency in differentiable structure learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 91756–91797. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a6eeb95990fc2704bacbba9b5aca3cff-Abstract-Conference.html.

Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for Bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41 (15):6755–6772, 2014. doi: 10.1016/j.eswa.2014.04.032.

Michel Gondran and Michel Minoux. *Graphs, Dioids and Semirings: New Models and Algorithms*. Operations Research/Computer Science Interfaces Series. Springer, 2008. doi: 10.1007/978-0-387-75450-5. URL https://link.springer.com/book/10.1007/978-0-387-75450-5.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995. doi: 10.1007/BF00994016.

Christina Heinze-Deml, Marloes H. Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018. doi: 10.1146/annurev-statistics-031017-100630.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/f7664060cc52bc6f3d620bcedc94a4b6-Paper.pdf.

Shuai Huang, Jing Li, Jieping Ye, Adam Fleisher, Kewei Chen, Teresa Wu, and Eric Reiman. Brain effective connectivity modeling for Alzheimer's disease by sparse Gaussian Bayesian network. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 931–939, 2011. doi: 10.1145/2020408.2020562.

Shuai Huang, Jing Li, Jieping Ye, Adam Fleisher, Kewei Chen, Teresa Wu, and Eric Reiman. A sparse structure learning algorithm for Gaussian Bayesian network identification from high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1328–1342, 2013. doi: 10.1109/TPAMI.2012.129.

Diviyan Kalainathan, Olivier Goudet, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Structural agnostic modeling: Adversarial learning of causal graphs. *Journal of Machine Learning Research*, 23(219):1–62, 2022. URL https://jmlr.org/papers/v23/19-529.html.

Neville K. Kitson, Anthony C. Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of Bayesian Network structure learning. *Artificial Intelligence Review*, 56(8): 8721–8814, 2023. doi: 10.1007/s10462-022-10351-w. URL https://doi.org/10.1007/s10462-022-10351-w.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural DAG learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rklbKA4YDS.

Andrew Li and Peter van Beek. Bayesian network structure learning with side constraints. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, volume 72 of *Proceedings of Machine Learning Research*, pages 225–236. PMLR, 2018. URL https://proceedings.mlr.press/v72/li18a.html.

Riccardo Massidda, Francesco Landolfi, Martina Cinquini, and Davide Bacciu. Constraint-free structure learning with smooth acyclic orientations. In *Proceedings of the 12th International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6W6VnVIpCr.

Raha Moraffah, Paras Sheth, Mansooreh Karami, Anchit Bhattacharya, Qianru Wang, Anique Tahir, Adrienne Raglin, and Huan Liu. Causal inference for time series analysis: Problems, methods and evaluation, 2021. URL https://arxiv.org/abs/2102.05829.

Achille Nazaret, Justin Hong, Elham Azizi, and David Blei. Stable differentiable causal discovery. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 37413–37445. PMLR, 2024. URL https://proceedings.mlr.press/v235/nazaret24a.html.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In *Advances in Neural Information Processing Systems*, 2020.

Ignavier Ng, Sébastien Lachapelle, Nan Rosemary Ke, Simon Lacoste-Julien, and Kun Zhang. On the convergence of continuous constrained optimization for structure learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8176–8198. PMLR, 2022. URL https://proceedings.mlr.press/v151/ng22b.html.

Ignavier Ng, Biwei Huang, and Kun Zhang. Structure learning with continuous optimization: A sober look and beyond. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 71–105. PMLR, 2024. URL https://proceedings.mlr.press/v236/ng24a.html.

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 Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1595–1605. PMLR, 2020. URL https://proceedings.mlr.press/v108/pamfil20a.html.

Jonas Peters, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models, 2014. URL https://arxiv.org/abs/1309.6779.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. MIT Press, 2017. ISBN 9780262037310. URL https://mitpress.mit.edu/9780262037310/elements-of-causal-inference/.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018. doi: 10.1002/sta4.183.

Alexander G. Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated DAG! causal discovery benchmarks may be easy to game. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/e987eff4a7c7b7e580d659feb6f60c1a-Abstract.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. URL https://www.science.org/doi/10.1126/sciadv.aau4996.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005. doi: 10.1126/science.1105809. URL https://www.science.org/doi/abs/10.1126/science.1105809.

Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006. URL https://jmlr.org/papers/v7/shimizu06a.html.

Liam Solus, Yuhao Wang, and Caroline Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021. doi: 10.1093/biomet/asaa104.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, 2 edition, 2000. URL https://direct.mit.edu/books/monograph/2057/Causation-Prediction-and-Search.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, volume 124 of *Proceedings of Machine Learning Research*, pages 1039–1048. PMLR, 2020. URL https://proceedings.mlr.press/v124/squires20a.html.

Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. Nts-notears: Learning nonparametric dbns with prior knowledge, 2023. URL https://arxiv.org/abs/2109.04286.

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.

Stephen Warshall. A theorem on Boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962. doi: 10.1145/321105.321107.

Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks, 2020. URL https://arxiv.org/abs/2010.09133.

Xing Yang, Chen Zhang, and Baihua Zheng. Segment-wise time-varying dynamic bayesian network with graph regularization. *ACM Trans. Knowl. Discov. Data*, 16(6), September 2022. ISSN 1556-4681. doi: 10.1145/3522589. URL https://doi.org/10.1145/3522589.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7154–7163. PMLR, 2019. URL https://proceedings.mlr.press/v97/yu19a.html.

Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. DAGs with no curl: An efficient DAG structure learning approach. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12156–12166. PMLR, 2021. URL https://proceedings.mlr.press/v139/yu21a.html.

Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric DAGs. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3414–3425. PMLR, 2020. URL https://proceedings.mlr.press/v108/zheng20a.html.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g2skStPB.

A Additional mechanistic probes

The probes in Appendices A.1, A.2 and A.4 are fixed-point local-sensitivity diagnostics, which evaluate gradients at a fixed input without training.

A.1 Symmetric long-cycle sensitivity probe

Fix a directed L-cycle on $d = L$ nodes with $X_L(\rho) = \rho B_L$, $\rho = 0.6$, and no optimization. We report $q(L) = \left|\sum_{e \in C_L} \nabla_e\right|$ and per-edge $s(L) = q(L)/L$ (symmetric cycles). Walk-sum baselines (DAGMA log-det, NOTEARS exponential, DAG-GNN polynomial) use $\nabla_e = \partial h / \partial A_e$ at $A = X_L(\rho)$. Bounded WG+SMNC maps the SMNC cycle loss on $G = X_L(\rho)$ ($\mu_{\text{cycle}} = 1$, $\tau_g = 1$) plus the DAGMA barrier through $A = W_f \circ G$ to an induced displacement on A (first-order chain rule in $\log g$, $\tau_g = 1$). Interpretation of the curves is in Section 4.1; values below 10^{-16} are plotted at that floor and log–log slopes omit exact zeros.

A.2 Walk accounting for walk-based penalties

Penalties $h(A) = \sum_{p \geq 1} c_p \operatorname{tr}(B^p)$, $B = A \circ A$, with DAGMA $c_p = 1/(p\, s^p)$, NOTEARS $c_p = 1/p!$, and DAG-GNN $c_p = \binom{d}{p} d^{-p}$ for $p \leq d$ [Yu et al., 2019, Wei et al., 2020]. On an isolated directed L-cycle with edge strength ρ, only multiples $p = kL$ contribute, so each repeat traversal $k \geq 2$ adds another term. For the exponential and polynomial penalties at typical linear-SEM weights, repeat traversals are negligible and length enters mainly through the coefficient schedule (Figure 2). For DAGMA's log-det barrier, repeat traversals can dominate near the spectral-radius boundary. These are local properties of the scalar penalties and do not by themselves predict benchmark accuracy.

A.3 Controlled decycling under the log-det barrier and closure-side cycle loss

The fixed-point probe in Appendix A.1 measures local sensitivity, but not whether cycle pressure overcomes an edge-retention cost. Here we separate reference-gradient calibration, single-cycle optimization, two-cycle optimization, and barrier continuation.

Shared construction and objectives. On a candidate support $\mathcal{E}$, all edge logits are optimized independently, all other edges are exactly masked out, and $W_{f,e} = 1$, so $A = G$. Write $g_e = G_e$, let $G^{(0)}$ be the initial gate matrix, and set $K = \lceil \log_2 d \rceil$ and $b = 2^K$. The fidelity term penalizes deviations from the initial graph:

$$\mathcal{L}_{\text{fid}}(G) = \frac{\lambda_{\text{fid}}}{2} \sum_{e \in \mathcal{E}} (g_e - g_e^{(0)})^2.$$

The raw and SMNC heads use the same closure states but different final logits:

$$\mathcal{L}_{\text{cyc}}^{\text{raw}}(G) = \frac{1}{d} \sum_{i=1}^{d} \operatorname{softplus}(U_{ii}^{(K)}),$$

$$\mathcal{L}_{\text{cyc}}(G) = \frac{1}{d} \sum_{i=1}^{d} \operatorname{softplus}(U_{ii}^{(K)}/b).$$

Thus the diagonal average includes all d nodes, whereas fidelity is a sum over candidate edges. We compare

$$\begin{aligned} \text{log-det only:} \quad & \mathcal{L}_{\text{fid}} + h_s(A), \\ \text{log-det + raw:} \quad & \mathcal{L}_{\text{fid}} + h_s(A) + \mu_{\text{cycle}} \mathcal{L}_{\text{cyc}}^{\text{raw}}, \\ \text{log-det + SMNC:} \quad & \mathcal{L}_{\text{fid}} + h_s(A) + \mu_{\text{cycle}} \mathcal{L}_{\text{cyc}}. \end{aligned}$$

Unless stated otherwise, $\lambda_{\text{fid}} = 5$, $s = 1$, and we use 500 full-gradient SGD updates with learning rate 0.2, float64 arithmetic, and the exact soft max-plus backend. A gate is removed for evaluation only when $g_e < \delta = 0.1$. The log-det-only objective is a gate-parameterized control, not a run of the complete DAGMA learning algorithm. Unless noted otherwise (the two-cycle runs below disable it), edge logits are clipped to $[-30, 30]$ in these single-cycle runs.

Reference-gradient calibration. For the single-cycle experiments, fix $d = 32$ and embed

$$1 \xrightarrow{\rho} 2 \xrightarrow{\rho} \cdots \xrightarrow{\rho} L \xrightarrow{q_0} 1, \qquad L \in \{4, 8, 16, 24, 32\}, \quad \rho = 0.8, \quad q_0 = 0.2.$$

The arrows give the initial gate values. Denote the closing edge by e_q and its logit by z_q. To separate reference amplitude from length response, calibrate only at the initial $L = 4$ graph:

$$10 \left| \frac{\partial \mathcal{L}_{\text{cyc}}^{\text{raw}}}{\partial z_q} \right|_{L=4} = \mu_{\text{cycle}}^{\text{match}} \left| \frac{\partial \mathcal{L}_{\text{cyc}}}{\partial z_q} \right|_{L=4}, \qquad \mu_{\text{cycle}}^{\text{match}} = 89.2416. \tag{9}$$

Keep these coefficients fixed across L and evaluate, without optimization,

$$g_{\text{ld}}(L) = \left| \frac{\partial h_1(A)}{\partial z_q} \right|,$$
$$g_{\text{raw}}(L) = \left| \frac{\partial [h_1(A) + 10\mathcal{L}_{\text{cyc}}^{\text{raw}}]}{\partial z_q} \right|,$$
$$g_{\text{SMNC}}^{\text{match}}(L) = \left| \frac{\partial [h_1(A) + 89.2416\mathcal{L}_{\text{cyc}}]}{\partial z_q} \right|.$$

Fidelity has zero gradient at initialization, and the two cycle terms have the same sign as log-det here; hence the joint gradients also match at $L = 4$. Table 3 shows a stronger long-cycle response after this calibration. These are aggregate edge gradients: the designated edge contributes to all L cyclic diagonal losses.

Table 3: Initial z_q gradient at $d = 32$, $s = 1$. Raw uses weight 10; SMNC uses the reference-matched weight 89.2416. This calibration is used only for the gradient diagnostic.

L	g_{ld}	g_{raw}	$g_{\text{SMNC}}^{\text{match}}$	$g_{\text{SMNC}}^{\text{match}}/g_{\text{raw}}$
4	1.696×10^{-2}	1.524×10^{-1}	1.524×10^{-1}	1.000
8	2.820×10^{-3}	1.783×10^{-1}	2.715×10^{-1}	1.523
16	7.923×10^{-5}	9.682×10^{-2}	5.257×10^{-1}	5.429
24	2.230×10^{-6}	2.213×10^{-2}	7.636×10^{-1}	34.513
32	6.277×10^{-8}	1.584×10^{-3}	9.678×10^{-1}	610.845

Single-cycle optimization with equal coefficients. We now optimize the same initial graphs with $\mu_{\text{cycle}} = 10$ for *both* closure heads, without reference matching. Table 4 shows that raw breaks the $L = 4, 8$ cycles but SMNC does not, whereas SMNC breaks the $L = 16, 24, 32$ cycles and raw does not; Figure 3 gives the first update at which each cycle breaks. For a cycle C_r with one designated closing edge $e_{q,r}$, path-gate retention after N updates is

$$R_r^{(N)} = \frac{\sum_{e \in C_r \setminus \{e_{q,r}\}} g_e^{(N)}}{\rho(L_r - 1)}.$$

It is the path-gate mass after N updates as a fraction of its initial value.

Table 4: Single-cycle optimization at $d = 32$, $s = 1$, and 500 updates. Both closure heads use $\mu_{\text{cycle}} = 10$. Entries are final closing-edge gates; bold values are below 0.1. The final column is SMNC path-gate retention.

L	Log-det only	+ raw	+ SMNC	$R^{(500)}$ (%)
4	0.1814	**0.03918**	0.1605	99.00
8	0.1966	**0.02979**	0.1471	98.77
16	0.1999	0.1045	**0.007290**	97.77
24	0.2000	0.1762	$\mathbf{8.575 \times 10^{-5}}$	97.01
32	0.2000	0.1981	$\mathbf{5.864 \times 10^{-6}}$	96.42

Two-cycle optimization with one shared coefficient. Next, embed two node-disjoint cycles in $d = 64$ nodes, with $L_s = 8$ and $L_\ell \in \{24, 32, 40, 48\}$, retaining the same ρ, q_0, fidelity, and fixed-s barrier. Each cycle has its own closing edge. Both heads use the same $\mu_{\text{cycle}} \in \{60, 80\}$ within each comparison, and both cycles share that coefficient. We set $\tau_g = 1$, $\tau_c = 0.1$, and $K = 6$, disable logit clipping, and compare 500 and 5000 updates from identical initializations. Thresholding never removes or freezes edges during training, and runs continue for their full budgets. These are deterministic controlled runs on fixed graphs.

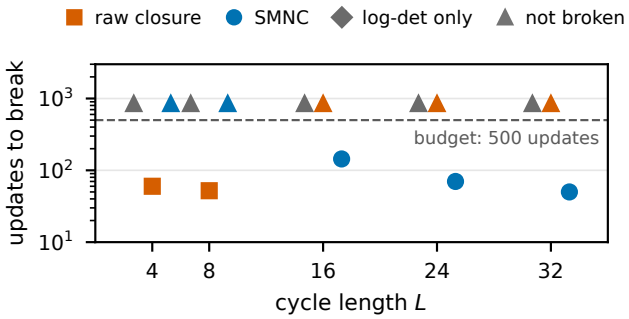

Figure 3: First update at which the single cycle of Table 4 is broken (a gate falls below $\delta = 0.1$), with the same $\mu_{\text{cycle}} = 10$ for both closure heads. Triangles mark runs not broken within the 500 updates.

Table 5 reports the first update at which both thresholded cycles are broken and the long-cycle path retention after 5000 updates. Every successful run removes only the two designated closing edges, and success persists through the final update. SMNC succeeds on all four pairs at both weights; raw succeeds only at $L_\ell = 24$, where it is faster and retains more path-gate mass. Log-det alone leaves both cycles intact.

Table 5: Two-cycle optimization at $d = 64$, $L_s = 8$, and $s = 1$. A dash means that both cycles have not been broken by update 5000. $R_\ell^{(5000)}$ is long-cycle path-gate retention. No additional path edges are removed. The log-det coefficient is fixed at one and does not depend on μ_{cycle}.

		First break of both cycles			$R_\ell^{(5000)}$ (%)	
μ_{cycle}	L_ℓ	Log-det only	+ raw	+ SMNC	Raw	SMNC
60	24	–	65	296	99.88	98.12
60	32	–	–	296	98.69	98.01
60	40	–	–	296	99.54	97.96
60	48	–	–	296	99.91	97.92
80	24	–	43	124	99.93	98.00
80	32	–	–	124	98.52	97.93
80	40	–	–	124	99.42	97.89
80	48	–	–	124	99.89	97.87

Increasing the budget from 500 to 5000 updates does not change any success/failure outcome. SMNC's long-cycle path retention nevertheless improves from 89.45%–95.08% to 97.87%–98.12%; its short-cycle retention reaches 98.57%–98.75%. Increasing the weight from 60 to 80 reduces the time to break both cycles from 296 to 124 updates, with slightly greater remaining path-gate shrinkage. The surviving raw long-cycle blocks at $L_\ell = 32, 40, 48$ have maximum absolute logit gradients below 7×10^{-16} after 5000 updates, consistent with near-stationary cancellation between fidelity and decycling terms.
Proposition C.4 gives $\max\{B_{\text{SMNC}}(8; 64), B_{\text{SMNC}}(L_\ell; 64)\} \approx 59.74$ for every tested pair, so both shared weights exceed this sufficient bound. This links the experiments to a net-gradient condition; it neither estimates the minimal successful weight nor establishes asymptotic scaling.
The earlier $d = 64$, $(L_s, L_\ell) = (4, 32)$, $\mu_{\text{cycle}} = 10$ pilot leaves both cycles intact under SMNC, so the follow-up changes both short-cycle length and weight. Moreover, changing d from 32 to 64 changes both the diagonal average and the final-scale normalizer. At fixed d, K, the two blocks are separable: this probe tests a shared coefficient across lengths, not competition between overlapping cycles.

Single-cycle barrier continuation. Return to the $d = 32$ single-cycle construction and the equal closure weights $\mu_{\text{cycle}} = 10$. We run $s = 1.0, 0.9, 0.8, 0.7, 0.6$ in sequence, warm-starting the logits, holding the fidelity and cycle coefficients fixed, and taking 500 updates per stage. Table 6 reports the first stage whose exit graph has a minimum-cost single-edge break. This schedule uses up to 2500 updates, is not compute-matched to the 500-update fixed-s run, and does not reproduce the complete benchmark schedule. Reducing s restores log-det pressure on long cycles, so log-det continuation can also remove them. SMNC's advantage here is that it breaks them already at $s = 1$.

Table 6: Single-cycle s-continuation at $d = 32$ and $\mu_{\text{cycle}} = 10$ for both closure heads. Entries give the value of s at the first successful stage; a dash denotes no minimum-cost single-edge break through $s = 0.6$.

L	Log-det only	+ raw	+ SMNC
4	–	1.0	0.6
8	–	1.0	0.6
16	0.6	0.7	1.0
24	0.6	0.6	1.0
32	0.6	0.6	1.0

Edge selection. Under fidelity, moving an initially retained edge to the threshold costs $V_e(\delta) = \lambda_{\text{fid}}(g_e^{(0)} - \delta)^2/2$. Thus $q_0 < \rho$ makes e_q the unique minimum-cost break, $q_0 = \rho$ makes all cycle edges equivalent, and $q_0 > \rho$ favors a path edge. Additional $d = 32$, $L = 32$, fixed-s runs with $q_0 \in \{0.7, 0.8, 0.9\}$ do not cross the removal threshold at $\mu_{\text{cycle}} = 10$; for $q_0 = 0.9$, the minimum gate lies on a ρ path edge. These runs serve only as edge-selection diagnostics. Their detailed values and the full optimization traces are retained in the reproduction materials. Normalization changes cycle pressure; it does not by itself identify a causally incorrect edge.

A.4 Unequal-length dominant-cycle masking probe

This two-cycle diagnostic asks whether a weaker cyclic component keeps a decycling signal when a stronger one coexists. The input is block-diagonal, $X_{L_1,L_2}(\rho_1, \rho_2) = \text{diag}(\rho_1 B_{L_1}, \rho_2 B_{L_2})$ with directed-cycle supports B_L and $\rho_1 > \rho_2$; we fix $L_1 = 8$, $\rho_1 = 0.9$ and vary $L_2 \in \{7, 8, 9\}$ and $\rho_2 \in \{0.2, 0.4, 0.6, 0.75, 0.85\}$. No training is involved. DAGMA and NOTEARS are evaluated on $A = X_{L_1,L_2}$, together with SDCD's spectral-radius gradient $uv^\top/(u^\top v)$ from power iteration on $A + 10^{-6}$ run to convergence (500 steps from the all-ones vector). The bounded WG+SMNC curve sets $W_{f,e} = 1$ and $G_e = \rho_k$ on the cycle edges of block k, takes one infinitesimal structural step under the bounded WG+SMNC objective ($\mu_{\text{cycle}} = 1$, $\tau_g = 1$, $\tau_c = 0.1$, $\beta = 5$, DAGMA barrier $s = 1.001$, $d = L_1 + L_2$, no data-fit term) and maps the signal back to the active A-space edges through the $M, Z \to A$ chain rule. Its total includes the WG-mapped DAGMA barrier, and the "SMNC term" curve is its closure-side part alone. We report the absolute weak-cycle signal $q_2 = \left|\sum_{e \in C_2} \nabla_e\right|$, which avoids ratio artifacts from changes in the dominant-cycle signal.

DAGMA's log-determinant and NOTEARS's trace-exponential are additive across the two blocks: zeroing C_1 leaves their signal on C_2 unchanged (bit-identical for DAGMA, equal to floating-point precision for NOTEARS), so their small q_2 at low ρ_2 is the weak cycle's own magnitude decay, empirically $\rho_2^{2L_2-1}$. For this block-diagonal input with $\rho_1 > \rho_2$, the spectral radius is attained by C_1 alone, so the exact spectral-radius gradient vanishes on C_2 even though it is cyclic (the plotted SDCD values, 10^{-10} to 3×10^{-8}, are the floor set by the 10^{-6} shift). This is a statement about the edges of a nondominant component when one component strictly attains the spectral radius: a cycle sharing a node with C_1 would receive some signal, and a tie between components would make the radius nondifferentiable. The SMNC cycle loss uses diagonal self-reachability scores, so each cyclic component contributes its own signal.

The key regime is small ρ_2: C_2 is still a directed cycle, but its local evidence is much weaker than C_1's. There the SMNC cycle term still gives C_2 a signal, DAGMA and NOTEARS lose it to magnitude decay, and SDCD's spectral gradient is confined to C_1. At large ρ_2 the magnitude decay is milder, and DAGMA and NOTEARS put visible pressure on C_2. A spectral penalty reaches C_2 once C_1's spectral radius falls below C_2's; in the two-cycle optimization probe this delays the weaker cycle's removal and, at low weights, leaves it unbroken within the budget (Appendix B).

A.5 Induced structural matrix in the nonlinear solver

Following the nonlinear DAGMA parameterization, we define the induced nonnegative edge-strength surrogate $A_{ij}^{\text{mag,nl}} := \left(\sum_{h=1}^{m_1} (\Theta_{j,h,i}^{(1)})^2\right)^{1/2}$, the ℓ_2 norm of the first-layer weights $\Theta^{(1)}$ linking source node i to the local hidden units of target node j; the implementation adds 10^{-12} inside the square root. When relation-side supervision is enabled, we map it to a soft structural matrix $G_{ij}^{\text{nl}} = a/(a+c)$ with $a = A_{ij}^{\text{mag,nl}}$, $c > 0$, for $i \neq j$ and $G_{ii}^{\text{nl}} = 0$. This ratio map preserves zeros and increases monotonically toward one. The SMNC relation head and the associated cycle and cue-relation losses are applied to G^{nl} with the same readout equations.

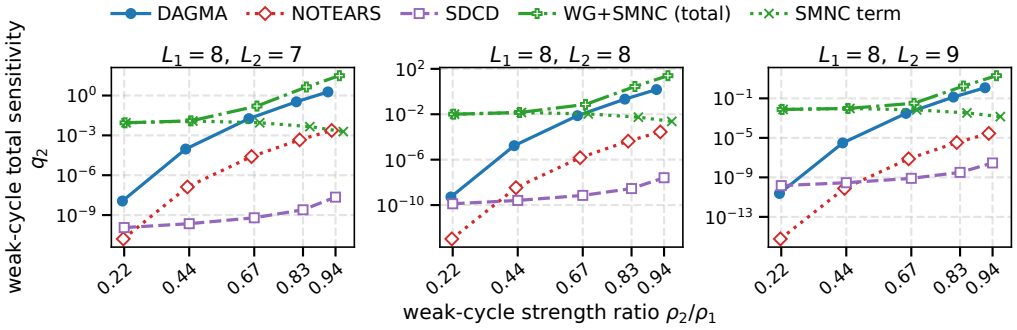

Figure 4: **Unequal-length dominant-cycle masking: weak-cycle total sensitivity** q_2 against ρ_2/ρ_1, for $L_1 = 8$, $\rho_1 = 0.9$ and $L_2 \in \{7, 8, 9\}$. The SMNC cycle term (dotted) gives the weaker cycle 0.0014–0.014 at $\mu_{\text{cycle}} = 1$, more when the cycle is weak (0.0075–0.0101 at $\rho_2 = 0.2$) than when it is strong (0.0014–0.0023 at $\rho_2 = 0.85$); the bounded WG+SMNC total adds the WG-mapped DAGMA barrier, which dominates at large ρ_2. DAGMA and NOTEARS lose weak-cycle signal through the weak cycle's own magnitude decay. SDCD's exact gradient on C_2 is zero for every $\rho_2 < \rho_1$.

Clamped GEMM in the nonlinear head. The nonlinear implementation evaluates the soft max-plus product with an older GEMM that clamps X/τ_c and Y/τ_c to $[-60, 60]$ before exponentiating and floors the product at 10^{-30}. Every composed entry is therefore at least $\tau_c \log 10^{-30} \approx -6.91$ ($\tau_c = 0.1$), including entries with no path. For the empty graph the exact scan gives $U_{vv} = -\infty$ and zero loss, whereas this head gives an all-scale diagonal logit of about 0.007 and a cycle BCE of about 0.70 per node ($d = 100$, $K = 7$, $\tau_c = \tau_s = 0.1$). Compositions below the floor have zero gradient: for an isolated cycle with gates 0.5 in $d = 64$, the per-edge gradient of this head equals that of the exact scan for $L \leq 8$ and falls to 3–8% of it for $L \geq 16$. Theorem 3.1 and Lemma C.2 therefore do not hold for this head, and the nonlinear results are empirical.

B Comparison with non-walk-based (spectral) penalties

The main text compares SMNC with penalties that pool weighted walks: NOTEARS, the polynomial penalties and DAGMA. SDCD [Nazaret et al., 2024] is a different kind of method. It constrains a spectral radius, builds neither a walk sum nor a closure, and returns one scalar. We therefore keep its comparison out of the main text and report only one probe here. SDCD's benchmark rows are in Appendix D.1, and its configuration is described in Appendix D.2.

Probe. On a single directed cycle, $\rho(A)$ is the geometric mean of the edge weights, so SDCD's penalty does not shrink with cycle length. For cycles in different strongly connected components (the probe's two cycles are disconnected blocks), $\rho(A)$ is the largest per-component value and, when that maximum is unique, its exact gradient is supported on the dominant component alone. We added SDCD's penalty to the two-cycle probe of Appendix A.3 (Table 5), with objective $\mathcal{L}_{\text{fid}} + h_1(A) + \mu\, h_{\text{SDCD}}(A)$ and $A = G$, where h_1 is the probe's log-det barrier shared by all methods. h_{SDCD} ports `PowerIterationGradient` from SDCD's released code (azizilab/sdcd, commit 909feb3) with its default 15 power iterations (warm-started, so the first update uses 30) and 10^{-6} shift, and matches the original class to machine precision. The short cycle has $L_s = 8$ and geometric mean 0.673; the long cycle has $L_\ell \in \{24, 32, 40, 48\}$ and geometric mean 0.755–0.777, so it is the stronger one.

Results. At the first update the exact spectral gradient on the short cycle is zero, and SDCD's power iteration as run sends it at most 5×10^{-4} of the total (Figure 5a), against 4%–12% for SMNC, whose share stays nonzero at every strength ratio. SMNC's split follows cycle length, though: across the sweep the short cycle receives 2%–24%, and less when it is the stronger cycle, consistent with SMNC also breaking it late. At $L_\ell = 48$ (Figure 5b–d), SDCD ($\mu = 30$) breaks the long cycle at update 8 and the short one at update 70, SMNC ($\mu = 80$) breaks them at updates 13 and 124, and raw closure ($\mu = 80$) breaks the short cycle at update 8 and never breaks the long one. In every one of the more than 160 SDCD runs in which a cycle broke (all weights, schedules and optimizers), the long, stronger cycle broke first or at the same update.

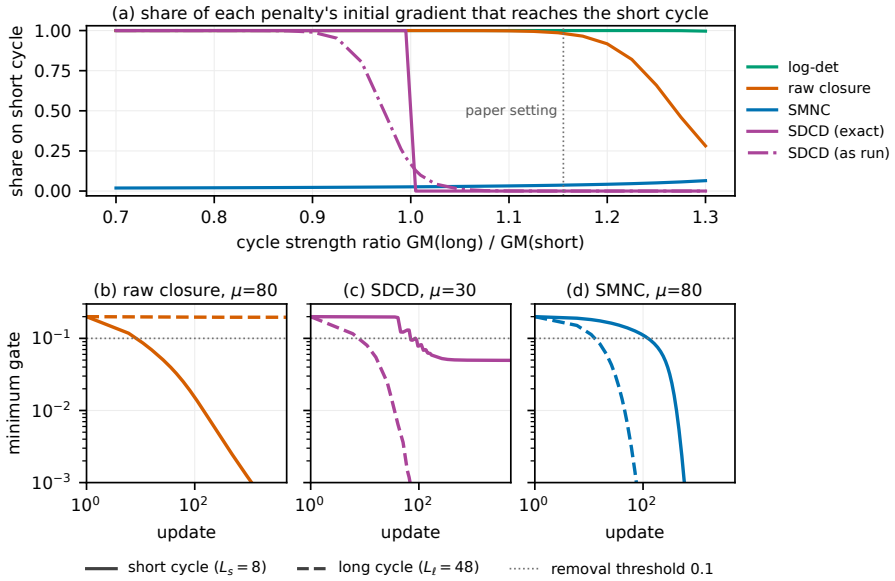

Figure 5: Probe: a short cycle ($L_s = 8$ edges) and a long cycle ($L_\ell = 48$), disconnected from each other. (a) Fraction of each penalty's first-step gradient that lands on the short cycle, against how strong the long cycle is relative to the short one (ratio of the cycles' geometric-mean edge weights; 1 means equal, dotted line: the probe's setting). Exact SDCD puts all of its gradient on whichever cycle is stronger and switches at ratio 1; "as run" is SDCD's warm-started power iteration (30 iterations by the first update). (b)–(d) Smallest gate on each cycle during training; a cycle is removed once its smallest gate falls below 0.1 (dotted). SDCD and SMNC remove the long cycle first and the short one later; raw closure never removes the long cycle.

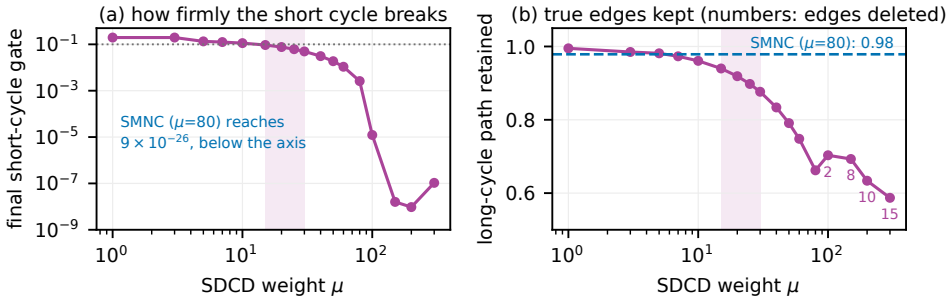

Figure 6: The same probe ($L_\ell = 48$, 5000 updates) with SDCD at a constant weight μ (how hard the penalty pushes) and its default 10^{-6} shift. (a) Final gate of the short cycle's closing edge; below the dotted line (0.1) the cycle is removed. (b) Fraction of the long cycle's path-gate mass kept; numbers give true path edges deleted. Shading: weights at which SDCD removes both cycles at every $L_\ell \in \{24, 32, 40, 48\}$ without deleting extra edges. SMNC at $\mu = 80$ is shown for reference (annotated in a, dashed in b).

Sensitivity to the weight. With a constant $\mu \in [15, 30]$, SDCD breaks both cycles at every L_ℓ without removing an extra path edge (shaded in Figure 6), but the break of the short cycle is shallow (at $L_\ell = 48$ its final gate is 0.094 at $\mu = 15$ and 0.050 at $\mu = 30$, so a removal threshold of 0.05 would leave it in place for $\mu = 15$–25), and across L_ℓ SDCD retains only 81%–94% of the long path's gate mass, against 98% for SMNC at $\mu = 80$. Below this range the short cycle survives at $L_\ell = 48$ (its gate stays at 0.113 for $\mu = 10$); above it SDCD removes true path edges. Most of the shallow break comes from the constant 10^{-6} shift: lowering it to 10^{-12} breaks both cycles at every L_ℓ for $\mu = 10$ and 30, with one exception to clean behavior: at $\mu = 30$, $L_\ell = 32$ it removes one extra path edge and keeps 96.7% of the long path.

Limits of the comparison. SDCD is not weaker at breaking cycles in general. On a single isolated cycle (the $d = 32$ protocol of Table 4, with SDCD's penalty in place of the closure head), SDCD at

the probe's common weight $\mu = 10$ (not a matched scale) breaks every length $L \in \{4, 8, 16, 24, 32\}$ within 18 updates without extra deletions, whereas SMNC at that weight breaks only $L \geq 16$. Breaking the long cycle first is not specific to SDCD: SMNC also breaks it first, at updates 13–36 versus 124–296 under SGD ($\mu = 60, 80$). The weights are not on a common scale: SDCD was swept finely (μ from 1 to 300), while raw closure and SMNC are shown only at their archived weights ($\mu = 60, 80$), so the width of each method's working range has not been compared. The probe also grafts SDCD's penalty onto the gate-only controlled problem (no MLP or sparsity term, a fidelity term in place of SDCD's likelihood, no stage-1 prescreening, float64 arithmetic and sigmoid gates in place of SDCD's weight-norm adjacency), so it characterizes the penalty alone.

C Proofs and additional technical details for Section 3

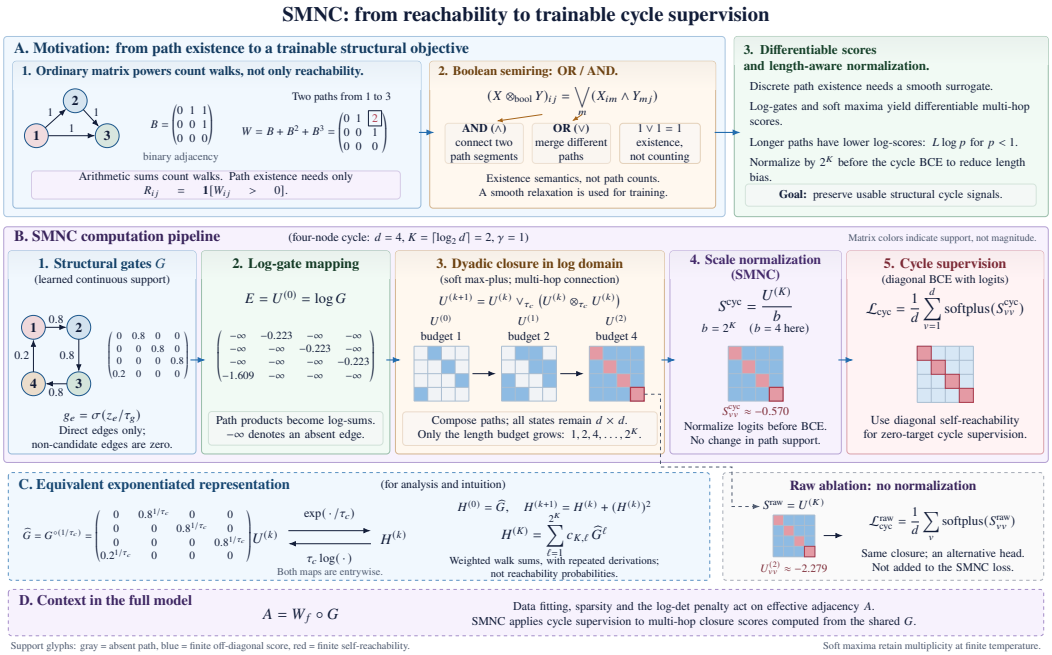

Figure 7: **Detailed SMNC computation and its exponentiated walk representation.** (A) Motivation: arithmetic matrix powers count walks, whereas Boolean closure encodes path existence. (B) A four-node cycle illustrates the continuous computation: structural gates are mapped to log-space, propagated through dyadic closure, and scale-normalized before diagonal cycle BCE. The matrices remain 4×4 while the walk-length budget increases from 1 to 2 to 4. (C) The exponentiated state provides an equivalent weighted-walk representation; the raw ablation bypasses scale normalization. (D) Connection to the effective adjacency $A = W_f \circ G$. Matrix colors indicate path support, not score magnitude.

C.1 Cue-free SMNC training: pseudocode

Algorithm 1 summarizes the cue-free training loop of the linear bounded-WG solver defined in Section 3.1–Section 3.3: the factorized effective adjacency $A = (W_f(M) \circ G(Z)) \circ (\mathbf{1} - I)$ (Eq. (1)); the log-gate scores $E = \log G(Z)$ with $E_{ii} = -\infty$ (Section 3.1); the dyadic soft max-plus closure scan at temperature τ_c (Eq. (3)), whose exact hard $(\vee, \otimes)$ semantics are given by Theorem 3.1; scale normalization of each dyadic state by $2^{\gamma k}$ (Eq. (5)); the readout on a scale set Ω (Eq. (6); Section 4.2 states the scale set used for each benchmark); the zero-target diagonal cycle BCE on the resulting cycle logit S^{cyc} (Section 3.1); and the DAGMA-style outer loop over the continuation schedule s that optimizes the total cue-free objective of Eq. (8) at $\mu_{\mathrm{rel}} = 0$ (Section 3.3).

C.2 Proof of the hard dyadic scan semantics

Proof of Theorem 3.1. Write $W^{(k)}_{ij}$ for the right-hand side, the best score of an $i \to j$ walk of length in $[1, 2^k]$. We prove $U^{(k)} = W^{(k)}$ by induction on k. For $k = 0$, $U^{(0)} = E$ and the only walks of length one are single edges, so $U^{(0)}_{ij} = E_{ij} = W^{(0)}_{ij}$ (with $E_{ii} = -\infty$ forbidding self-loops).
Assume $U^{(k)} = W^{(k)}$. The update

$$U^{(k+1)}_{ij} = \max\Big\{U^{(k)}_{ij},\ \max_m \big(U^{(k)}_{im} + U^{(k)}_{mj}\big)\Big\}$$

takes the maximum of a skip term and a concatenation term.
$(\leq)$ The skip term is the score of a walk of length at most $2^k \leq 2^{k+1}$. Each concatenation term $U^{(k)}_{im} + U^{(k)}_{mj}$ is, by the induction hypothesis, the score of an $i \to m$ walk followed by an $m \to j$ walk, each of length in $[1, 2^k]$, i.e. of an $i \to j$ walk of length in $[2, 2^{k+1}]$. Hence $U^{(k+1)}_{ij} \leq W^{(k+1)}_{ij}$.

Algorithm 1 Cue-free SMNC training of the linear bounded-WG solver (cycle-regularized continuous DAG learning)

Require: data X; outer continuation schedule $\{s_t\}_{t=1}^{N_{\text{outer}}}$; magnitude bound β; gate temperature τ_g; closure temperature τ_c; scale-normalization exponent γ; readout scale set Ω and temperature τ_s (Section 4.2 states Ω per benchmark); weights $\lambda_1, \mu_{\text{cycle}}$; initial data-fit weight μ_{init} and its decay factor; number of inner steps per stage; magnitude threshold ω

Initialize magnitude logits M and gate logits Z; $\mu \leftarrow \mu_{\text{init}}$
for $t = 1, \dots, N_{\text{outer}}$ **do** ▷ DAGMA-style outer loop, Eq. (8)
  $s \leftarrow s_t$
  for each of the stage's fixed number of inner steps (Section 4.2) **do** ▷ warm-started from stage $t-1$
    $W_f(M) \leftarrow \beta \tanh(M)$; $G(Z) \leftarrow \sigma(Z/\tau_g) \circ (\mathbf{1} - I)$ ▷ Eq. (1)
    $A \leftarrow (W_f(M) \circ G(Z)) \circ (\mathbf{1} - I)$
    $U^{(0)} \leftarrow \log G(Z)$, with diagonal set to $-\infty$ ▷ log-gate scores E
    for $k = 0, \dots, K-1$ **do** ▷ $K = \lceil \log_2 d \rceil$
      $U^{(k+1)} \leftarrow U^{(k)} \vee_{\tau_c} \left(U^{(k)} \otimes_{\tau_c} U^{(k)}\right)$ ▷ Eq. (3)
    end for
    for $k = 0, \dots, K$ **do**
      $\widetilde{U}^{(k)} \leftarrow U^{(k)} / 2^{\gamma k}$ ▷ scale normalization, Eq. (5)
    end for
    $S^{(\gamma)}(\Omega) \leftarrow \widetilde{U}^{(k)}$ if $\Omega = \{k\}$, else $\tau_s \log \sum_{k \in \Omega} \exp\big(\widetilde{U}^{(k)}/\tau_s\big)$ ▷ readout, Eq. (6)
    $S^{\text{cyc}} \leftarrow S^{(\gamma)}(\Omega)$ ▷ diagonal cycle logit, Section 3.1
    $\mathcal{L}_{\text{cyc}} \leftarrow \frac{1}{d} \sum_v \text{softplus}\big(S^{\text{cyc}}_{vv}\big)$ ▷ zero-target diagonal cycle BCE
    $h_s(A) \leftarrow d \log s - \log\det(sI - A \circ A)$ ▷ Eq. (7)
    $\mathcal{L} \leftarrow \mu\big(\text{score}(A) + \lambda_1 \|A\|_1\big) + h_s(A) + \mu_{\text{cycle}} \mathcal{L}_{\text{cyc}}$ ▷ Eq. (8), $\mu_{\text{rel}} = 0$
    Take a gradient step on $\mathcal{L}$ with respect to (M, Z)
  end for
  $\mu \leftarrow \mu \cdot$ (decay factor) ▷ DAGMA-style decrease of the data-fit weight
end for
$A \leftarrow (W_f(M) \circ G(Z)) \circ (\mathbf{1} - I)$; set $A_{ij} \leftarrow 0$ wherever $|A_{ij}| < \omega$ ▷ $\omega = 0.1$ on most datasets, Section 4.2
return A

($\geq$) Let π be any $i \to j$ walk of length $\ell \in [1, 2^{k+1}]$. If $\ell \leq 2^k$, its score is at most $W^{(k)}_{ij} = U^{(k)}_{ij}$, the skip term. Otherwise $2^k < \ell \leq 2^{k+1}$; let m be the vertex reached after the first 2^k steps of π. This splits π into an $i \to m$ walk of length 2^k and an $m \to j$ walk of length $\ell - 2^k \in [1, 2^k]$, whose scores are at most $U^{(k)}_{im}$ and $U^{(k)}_{mj}$ by the induction hypothesis. Hence the score of π is at most $U^{(k)}_{im} + U^{(k)}_{mj} \leq U^{(k+1)}_{ij}$. Maximizing over π gives $W^{(k+1)}_{ij} \leq U^{(k+1)}_{ij}$, which completes the induction.

Finally, since $E \leq 0$, every closed walk has non-positive score. If a maximizing walk repeats a vertex, deleting the closed sub-walk between the two visits does not lower its score and shortens it, so it stays within the budget. Repeating this yields a simple path if $i \neq j$. If $i = j$, first cut the walk at its first return to i and then delete closed sub-walks among the intermediate vertices; the result is a simple cycle through i (of length at least two, since $E_{ii} = -\infty$). □

C.3 Proof of the length-stable cycle lower bound

Proof of Theorem 3.2. Since $|A_e| = |W_{f,e}| G_e \leq \beta G_e$, the assumption $|A_e| \geq \alpha$ implies $G_e \geq \alpha/\beta$ for every edge $e \in C$. For any node v on the cycle, the cycle path from v back to v has length L and is feasible at every dyadic scale a such that $2^a \geq L$. Hence, for any such scale,

$$U^{(a)}_{vv} \geq L \log(\alpha/\beta).$$

Since $\log(\alpha/\beta) \leq 0$, dividing by 2^a gives

$$\frac{U^{(a)}_{vv}}{2^a} \geq \frac{L}{2^a} \log(\alpha/\beta).$$

For $a = k_L = \lceil \log_2 L \rceil$, we have $L \le 2^{k_L}$, so $L/2^{k_L} \le 1$. Therefore

$$\frac{U_{vv}^{(k_L)}}{2^{k_L}} \ge \log(\alpha/\beta).$$

For $a = K = \lceil \log_2 d \rceil$, the cycle is simple and hence $L \le d \le 2^K$, so $L/2^K \le 1$. Therefore

$$\frac{U_{vv}^{(K)}}{2^K} \ge \log(\alpha/\beta).$$

The smooth recursion and smooth readouts are log-sum-exp relaxations of the corresponding hard maxima and therefore, in exact arithmetic, dominate the hard max values (Lemma C.2). The same lower bounds follow for the corresponding smooth readouts; a finite-precision implementation can violate them where its exponential-space products underflow (Section 3.2). □

C.4 Numerically stabilized GEMM-oriented algebra

In the linear solver, each soft composition is evaluated as one GEMM in shifted exponential space: with row shifts $r_i = \max_m X_{im}$ and column shifts $c_j = \max_m Y_{mj}$, $(X \otimes_{\tau_c} Y)_{ij} = r_i + c_j + \tau_c \log \sum_m e^{(X_{im}-r_i)/\tau_c} e^{(Y_{mj}-c_j)/\tau_c}$, computed in `float32`. Masked entries ($E = -\infty$) contribute exactly zero, and entries whose product underflows are set to $-\infty$. The shift leaves $U^{(k)}$ unchanged in exact arithmetic, but because it is per row and column rather than per entry, in `float32` it can drop an (i, j) entry whose best composition lies far below $r_i + c_j$, e.g. a long weak cycle next to strong incident edges (Section 3.2). Relation logits are clipped to $[-50, 50]$. Products are floored at the smallest normal `float32` value before the logarithm, which also distorts very small positive products. The nonlinear solver uses a clamped variant of this GEMM (Appendix A.5). Exact-semantic theorems in Section 3.2 should not be read as guarantees on `float32` training dynamics without a validated stable scan.

C.5 Proofs of the gradient factor, coarsest-scale dominance, and the soft-scan excess

Proof of Corollary 3.3. By Theorem 3.1, $U_{vv}^{(a)}$ is a maximum over the finite set of closed walks through v of length at most 2^a. If the cycle candidate C is its unique maximizer, the maximum is still attained only by C in a neighborhood of the current gates, and there $U_{vv}^{(a)} = \sum_{e \in C} \log g_e$. Because $E < 0$ off the diagonal, deleting a closed sub-walk strictly increases a walk score, so a unique maximizer is a simple cycle and each of its edges appears once. Hence $\partial S_{vv,a}/\partial \log g_e = 2^{-a}$, and $\mathrm{softplus}' = \sigma$ gives $\partial \ell_{v,a}^{\mathrm{cyc}}/\partial \log g_e = 2^{-a}\sigma(S_{vv,a})$. Finally, $2^{k_L-1} < L \le 2^{k_L}$ gives $2^{-k_L} \in (1/(2L), 1/L]$, and $2^{K-1} < d \le 2^K$ gives $2^{-K} \in (1/(2d), 1/d]$. □

Lemma C.1 (Coarsest-scale dominance of the hard readout)**.** *For the hard scan and any $\gamma > 0$, $U_{ij}^{(a)}/2^{\gamma a}$ is non-decreasing in a. Hence $S_{ij}^{(\gamma),\max}(\Omega) = U_{ij}^{(\max\Omega)}/2^{\gamma \max\Omega}$, and the hard Ω_{all} and Ω_{final} readouts coincide.*

Proof of Lemma C.1. The skip term in Eq. (2) gives $U^{(a+1)} \ge U^{(a)}$ entrywise, and $U^{(a)} \le 0$ because every walk score is a sum of entries $E_e \le 0$. Hence, for any $\gamma > 0$,

$$\frac{U_{ij}^{(a+1)}}{2^{\gamma(a+1)}} \ge \frac{U_{ij}^{(a)}}{2^{\gamma(a+1)}} \ge \frac{U_{ij}^{(a)}}{2^{\gamma a}},$$

where the second step holds because dividing a non-positive number by a larger positive number cannot decrease it (the chain is read in the extended-real order and is immediate when $U_{ij}^{(a)} = -\infty$). Thus the maximum over Ω is attained at $\max\Omega$. If $U_{ij}^{(a)}$ is finite it is strictly negative (off-diagonal gates lie in $(0, 1)$), the second inequality is strict, and the active scale $\max\Omega$ is unique. □

The bound in "Objective used in our experiments". Let C satisfy the assumptions of Theorem 3.2 and be the unique maximizer of $U_{vv}^{(K)}$ for every $v \in C$. By Corollary 3.3 with $a = K$, the diagonal terms of the nodes on C contribute $\frac{1}{d\,2^K} \sum_{v \in C} \sigma(U_{vv}^{(K)}/2^K)$ to $\partial \mathcal{L}_{\mathrm{cyc}}/\partial \log g_e$ for every $e \in C$. Theorem 3.2 gives $U_{vv}^{(K)}/2^K \ge \log(\alpha/\beta)$, and $U_{vv}^{(K)} \le 0$ gives $\sigma(U_{vv}^{(K)}/2^K) \le 1/2$; with $d \le 2^K < 2d$ the contribution lies in $[L\sigma(\log(\alpha/\beta))/(2d^2),\ L/(2d^2)]$. For the unnormalized head $\Lambda = U^{(K)}$, the same terms contribute $\frac{1}{d} \sum_{v \in C} \sigma(U_{vv}^{(K)})$, and $U_{vv}^{(K)} = \sum_{e \in C} \log g_e \le L \log p_{\max}$.

Diagonal terms of other nodes add a non-negative amount wherever the hard readout is differentiable, since $U^{(K)}$ is non-decreasing in every $\log g_e$. These are derivatives in log-gate coordinates at fixed d, for the hard scan. With $g_e = \sigma(Z_e/\tau_g)$, the derivative with respect to the trained parameter carries the extra factor $(1 - g_e)/\tau_g$, which vanishes as $g_e \to 1$. At finite τ_c, soft values dominate hard values (Lemma C.2), but this does not imply a corresponding lower bound on the per-edge gradient.

Lemma C.2 (Walk expansion and excess of the soft scan)**.** *Let $U^{(k)}_{\mathrm{soft}}$ follow Eq.* (3) *and $U^{(k)}_{\mathrm{hard}}$ follow Eq.* (2)*, from the same E, and let $T := \exp(E/\tau_c) = G^{\circ 1/\tau_c}$ (zero diagonal).*
(i) $\exp(U^{(k)}_{\mathrm{soft}}/\tau_c) = \sum_{\ell=1}^{2^k} c_{k,\ell} T^\ell$ *with integer coefficients $c_{k,\ell} \geq 1$ that do not depend on G.*
(ii) $0 \leq U^{(k)}_{\mathrm{soft},ij} - U^{(k)}_{\mathrm{hard},ij} \leq (2^k - 1)\tau_c \log(2d)$ *whenever $U^{(k)}_{\mathrm{hard},ij} > -\infty$; both states are $-\infty$ at exactly the same entries.*
(iii) For $\gamma = 1$ and $a \leq b$, $\widetilde{U}^{(a)}_{\mathrm{soft},ij} \leq \widetilde{U}^{(b)}_{\mathrm{soft},ij} + \tau_c \log(2d)$, and the smooth readout of Eq. (6) *satisfies* $S^{(1)}_{ij}(\Omega_{\mathrm{final}}) \leq S^{(1)}_{ij}(\Omega_{\mathrm{all}}) \leq S^{(1)}_{ij}(\Omega_{\mathrm{final}}) + \tau_c \log(2d) + \tau_s \log(K + 1)$.

Proof. (i) Let $H^{(k)} := \exp(U^{(k)}_{\mathrm{soft}}/\tau_c)$. Since $\exp((a \vee_{\tau_c} b)/\tau_c) = e^{a/\tau_c} + e^{b/\tau_c}$ and $\exp((X \otimes_{\tau_c} Y)_{ij}/\tau_c) = \sum_m e^{X_{im}/\tau_c} e^{Y_{mj}/\tau_c}$, the soft recursion is $H^{(k+1)} = H^{(k)} + (H^{(k)})^2$ with ordinary matrix multiplication and $H^{(0)} = T$. If $H^{(k)} = \sum_{\ell=1}^{2^k} c_{k,\ell} T^\ell$ with integers $c_{k,\ell} \geq 1$, then $H^{(k+1)}$ has integer coefficients; for $\ell \leq 2^k$ they are at least $c_{k,\ell} \geq 1$, and for $2^k < \ell \leq 2^{k+1}$ they are $\sum_{\ell_1+\ell_2=\ell} c_{k,\ell_1} c_{k,\ell_2} \geq 1$, since $\ell = 2^k + (\ell - 2^k)$ with both parts in $[1, 2^k]$.
(ii) Both soft operations are monotone in their arguments and dominate the corresponding maxima, so $U^{(k)}_{\mathrm{soft}} \geq U^{(k)}_{\mathrm{hard}}$ by induction; an entry is $-\infty$ in either scan exactly when all of its terms are. For the upper bound, log-sum-exp over n terms exceeds their maximum by at most $\tau_c \log n$, and $\otimes_{\tau_c}$ has d terms while $\vee_{\tau_c}$ has two. Let $g_k \geq 0$ bound $U^{(k)}_{\mathrm{soft}} - U^{(k)}_{\mathrm{hard}}$ on finite entries. Then $(U^{(k)}_{\mathrm{soft}} \otimes_{\tau_c} U^{(k)}_{\mathrm{soft}})_{ij} \leq (U^{(k)}_{\mathrm{hard}} \otimes U^{(k)}_{\mathrm{hard}})_{ij} + 2g_k + \tau_c \log d$, so $U^{(k+1)}_{\mathrm{soft},ij} \leq U^{(k+1)}_{\mathrm{hard},ij} + 2g_k + \tau_c \log(2d)$. With $g_0 = 0$, the recursion $g_{k+1} = 2g_k + \tau_c \log(2d)$ gives $g_k = (2^k - 1)\tau_c \log(2d)$.
(iii) By (ii) and Lemma C.1, $\widetilde{U}^{(a)}_{\mathrm{soft}} \leq \widetilde{U}^{(a)}_{\mathrm{hard}} + (1 - 2^{-a})\tau_c \log(2d) \leq \widetilde{U}^{(b)}_{\mathrm{hard}} + \tau_c \log(2d) \leq \widetilde{U}^{(b)}_{\mathrm{soft}} + \tau_c \log(2d)$. The readout over the $K + 1$ scales of Ω_{all} lies between its largest term, which is at least $\widetilde{U}^{(K)}_{\mathrm{soft}} = S^{(1)}(\Omega_{\mathrm{final}})$, and that term plus $\tau_s \log(K + 1)$; applying the first claim with $b = K$ completes the proof. □

Thus, in contrast to the hard scan, soft states can be positive. Off-diagonal normalized soft scores need not be monotone in a (Lemma C.2(iii)). For diagonals in the exact soft recursion, $H^{(k+1)}_{vv} \geq (H^{(k)}_{vv})^2$, so at $\gamma = 1$, $\widetilde{U}^{(k+1)}_{\mathrm{soft},vv} \geq \widetilde{U}^{(k)}_{\mathrm{soft},vv}$. Part (i) is the tempered walk-series form used in Section 3.1: each extra traversal of a cycle with gate product Q multiplies a walk term by Q^{1/τ_c}, but the fixed counts $c_{k,\ell}$ grow rapidly with ℓ over most of the range (e.g. $c_{7,\ell} = 1, 7, 42, 231, \ldots$, peaking near 10^{25} at $\ell = 74$), so repeated traversals are discounted, not removed. On an isolated cycle the multiplicity enters the gradient through $r = \partial U^{(K)}_{vv}/\partial \log Q = \sum_m m\, \omega_m \geq 1$ (Appendix C.6); for $d = 128$ and $\tau_c = 0.1$, a 2-cycle with both gates 0.9 has $r \approx 11.7$ and one with gates 0.7 has $r \approx 1.03$.

Remark C.3 (Active scale and length behavior)*.* On a uniform-gate chain or cycle, a raw product-style path log-score scales as $L \log p$, so a zero-target BCE factor behaves like $\sigma(L \log p) \approx p^L$. SMNC instead normalizes dyadic path scores by their scale budgets. For the hard all-scale readout, $S^{(1),\max}_{vv}(\Omega_{\mathrm{all}}) = \max_{a=0,\ldots,K} U^{(a)}_{vv}/2^a$, and by Lemma C.1 the active scale is $a^\star = K = \lceil \log_2 d \rceil$ whenever $U^{(K)}_{vv}$ is finite, so the local factor is $2^{-K} = \Theta(1/d)$, the same as for Ω_{final} (the readout of our linear runs). The $\Theta(1/L)$ factor of Corollary 3.3 is a minimal-covering-scale statement, not a property of the all-scale or final-scale readouts. With smooth scale aggregation ($\tau_s > 0$, as in our nonlinear runs) and the hard scan, $\partial S^{(1)}_{vv}(\Omega_{\mathrm{all}})/\partial \log g_e = \sum_a w_a\, \partial \widetilde{U}^{(a)}_{vv}/\partial \log g_e$ with $w_a \propto \exp(\widetilde{U}^{(a)}_{vv}/\tau_s)$. Because $U^{(a)}_{vv} \leq U^{(K)}_{vv} \leq 0$, $w_a/w_K \leq \exp\big(-(2^{K-a} - 1)|\widetilde{U}^{(K)}_{vv}|/\tau_s\big)$, so finer scales contribute only when $|\widetilde{U}^{(K)}_{vv}| \lesssim \tau_s$. On an isolated length-$L$ cycle the factor is $\sum_{a \geq k_L} w_a 2^{-a} \in [2^{-K}, 2^{-k_L}]$; for example, at $d = 128$, $\tau_s = 0.1$, $L = 3$ and gates 0.95, the per-edge BCE derivative is about 3.9×10^{-2}, versus 3.9×10^{-3} for Ω_{final}.

C.6 Sufficient cycle-weight bounds for the controlled probes

This subsection analyzes the fidelity–cycle competition in Appendix A.3. The statement concerns isolated cycle supports only. Write $\mathcal{L}_{\mathrm{raw}} = \mathcal{L}_{\mathrm{cyc}}^{\mathrm{raw}}$ and $\mathcal{L}_{\mathrm{SMNC}} = \mathcal{L}_{\mathrm{cyc}}$ as defined there.

Proposition C.4 (Sufficient pressure on an intact isolated cycle)**.** *Let the allowed support be a union of node-disjoint simple directed cycles and isolated nodes. On every allowed edge, let $g_e = \sigma(z_e/\tau_g)$ with finite z_e and $\tau_g > 0$, and set $A = G$. Use the exact masked recursion in Eq.* (3)*, with one fixed temperature $\tau_c > 0$ for both soft operations, no zero-length walks, and no logit clipping. Let $K = \lceil \log_2 d \rceil$ and $b = 2^K$. Consider*

$$\mathcal{J}_{\mathrm{raw}} = \mathcal{L}_{\mathrm{fid}} + \nu h_1(G) + \mu_{\mathrm{cycle}}\mathcal{L}_{\mathrm{raw}}, \qquad \mathcal{J}_{\mathrm{SMNC}} = \mathcal{L}_{\mathrm{fid}} + \nu h_1(G) + \mu_{\mathrm{cycle}}\mathcal{L}_{\mathrm{SMNC}},$$

where $\lambda_{\mathrm{fid}} > 0$, $\nu \geq 0$, and $\mu_{\mathrm{cycle}} \geq 0$. For one cycle C_L, let its designated edge have initial gate q_0 and current gate q, with $0 < \delta < q_0 < 1$. Define

$$\begin{aligned} B_{\mathrm{raw}}(L; d) &:= \frac{\lambda_{\mathrm{fid}} q_0^2 d}{4L}(1 + \delta^{-L}), \\ B_{\mathrm{SMNC}}(L; d) &:= \frac{\lambda_{\mathrm{fid}} q_0^2 db}{4L}(1 + \delta^{-L/b}), \\ B_{\mathrm{ld}}(L) &:= \frac{\lambda_{\mathrm{fid}} q_0^2}{8}(\delta^{-2L} - 1). \end{aligned} \tag{10}$$

At any state where all edges of C_L satisfy $g_e \geq \delta$, $\mu_{\mathrm{cycle}} > B_{\mathrm{raw}}$ guarantees $\partial \mathcal{J}_{\mathrm{raw}}/\partial z_q > 0$, and $\mu_{\mathrm{cycle}} > B_{\mathrm{SMNC}}$ guarantees $\partial \mathcal{J}_{\mathrm{SMNC}}/\partial z_q > 0$. For the log-det-only objective $\mathcal{J}_{\mathrm{ld}} = \mathcal{L}_{\mathrm{fid}} + \nu h_1(G)$, $\nu > B_{\mathrm{ld}}$ gives the analogous guarantee. For several cycles, a common weight exceeding the maximum of their corresponding bounds excludes finite first-order stationary points whose thresholded graph is cyclic.

Proof. Let $Q_C = \prod_{e \in C_L} g_e$, $H^{(k)} = \exp(U^{(k)}/\tau_c)$ entrywise and $T = H^{(0)}$, so $T_{ij} = g_{ij}^{1/\tau_c}$ on allowed edges and zero elsewhere. The soft recursion is $H^{(k+1)} = H^{(k)} + (H^{(k)})^2$ with ordinary matrix products, hence $H^{(K)} = \sum_{\ell=1}^{b} c_{K,\ell} T^{\ell}$ with integer coefficients $c_{K,\ell} \geq 1$ that do not depend on the gates. On an isolated simple cycle only powers $\ell = mL$ reach a diagonal entry, $(T^{mL})_{vv} = Q_C^{m/\tau_c}$, so for every $v \in C_L$

$$u := U_{vv}^{(K)} = \tau_c \log \sum_{m=1}^{\lfloor b/L \rfloor} a_m Q_C^{m/\tau_c}, \qquad a_m = c_{K,mL}. \tag{11}$$

Since one complete traversal is feasible and $a_1 \geq 1$, $u \geq \log Q_C \geq L \log \delta$, and $r := \partial u / \partial \log Q_C = \sum_m m\omega_m \geq 1$ with $\omega_m \propto a_m Q_C^{m/\tau_c}$ summing to one; the bound on r comes from differentiating Eq. (11), since every closed walk traverses the designated edge at least once, and not from a score inequality. Because $\partial \log Q_C / \partial z_q = (1-q)/\tau_g$ and the cycle contributes L diagonal losses,

$$\begin{aligned} \frac{\partial \mathcal{L}_{\mathrm{raw}}}{\partial z_q} &= \frac{L(1-q)}{d\tau_g} r\sigma(u) \geq \frac{L(1-q)}{d\tau_g}\sigma(L \log \delta), \\ \frac{\partial \mathcal{L}_{\mathrm{SMNC}}}{\partial z_q} &= \frac{L(1-q)}{db\tau_g} r\sigma(u/b) \geq \frac{L(1-q)}{db\tau_g}\sigma((L/b)\log \delta), \end{aligned}$$

and other disconnected blocks contribute nothing to z_q. The fidelity derivative is $-\lambda_{\mathrm{fid}} q(q_0 - q)(1-q)/\tau_g$ with $q(q_0 - q) \leq q_0^2/4$. On this cycle the $s = 1$ log-det term is $-\log(1 - Q_C^2)$, with $\partial h_1/\partial z_q = \frac{1-q}{\tau_g}\frac{2Q_C^2}{1-Q_C^2} \geq 0$. Discarding this nonnegative help and using $1/\sigma(x) = 1 + e^{-x}$, either closure gradient strictly exceeds the opposing fidelity gradient under its bound in Eq. (10); for log-det alone, $\nu\, 2\delta^{2L}/(1 - \delta^{2L}) > \lambda_{\mathrm{fid}} q_0^2/4$ gives B_{ld}. At finite logits $(1-q)/\tau_g > 0$, so an intact cycle would have a nonzero designated-edge derivative, contradicting first-order stationarity. Applying this to every cycle proves the statement about the maximum bound. □

Length dependence and scope. For fixed $\delta, q_0, \lambda_{\mathrm{fid}}$ and a family with $d = \Theta(L)$, so that $b = 2^{\lceil \log_2 d \rceil} = \Theta(d)$ and $L/b = \Theta(1)$, the sufficient expressions obey $B_{\mathrm{raw}} = \Theta(\delta^{-L})$, $B_{\mathrm{SMNC}} = \Theta(L)$ and $B_{\mathrm{ld}} = \Theta(\delta^{-2L})$. They are sufficient per-cycle bounds, not minimal weights, and they are not uniform over heterogeneous cycle families: a fixed-length short cycle with growing d has an SMNC

bound that scales as $\Theta(d^2)$. The closure bounds ignore the common log-det gradient, and B_{ld} concerns an adjustable multiplier ν that is fixed at one in the reported ablation.
For the two-cycle probe ($d = b = 64$, $q_0 = 0.2$, $\lambda_{\mathrm{fid}} = 5$, $\delta = 0.1$), $B_{\mathrm{SMNC}}(L; 64) = \frac{204.8}{L}(1 + 10^{L/64})$, so $B_{\mathrm{SMNC}}(8; 64) \approx 59.74$, and the bounds for $L = 24, 32, 40, 48$ are approximately $28.77, 26.64, 26.71, 28.26$. The proposition rules out cancellation on an intact isolated cycle. It does not guarantee success within a fixed number of updates or that only a minimum-cost edge is removed, and it does not cover a trainable magnitude branch, general data-fitting losses, clipped logits, or overlapping cycles whose closed walks can avoid the designated edge.

D Additional synthetic results

D.1 Across-d overview and compact tables

Figure 8 shows RAW dSHD across graph sizes for ER $\in \{2, 4, 8\}$ on linear and nonlinear SEMs, for every method we ran. Tables 7–9 list all methods at $d = 100$ for every edge density. TOPK keeps up to s_0 edges and is an oracle-sparsity diagnostic. Results for the other graph sizes and for the remaining metrics come from the same archived runs and are omitted for length.

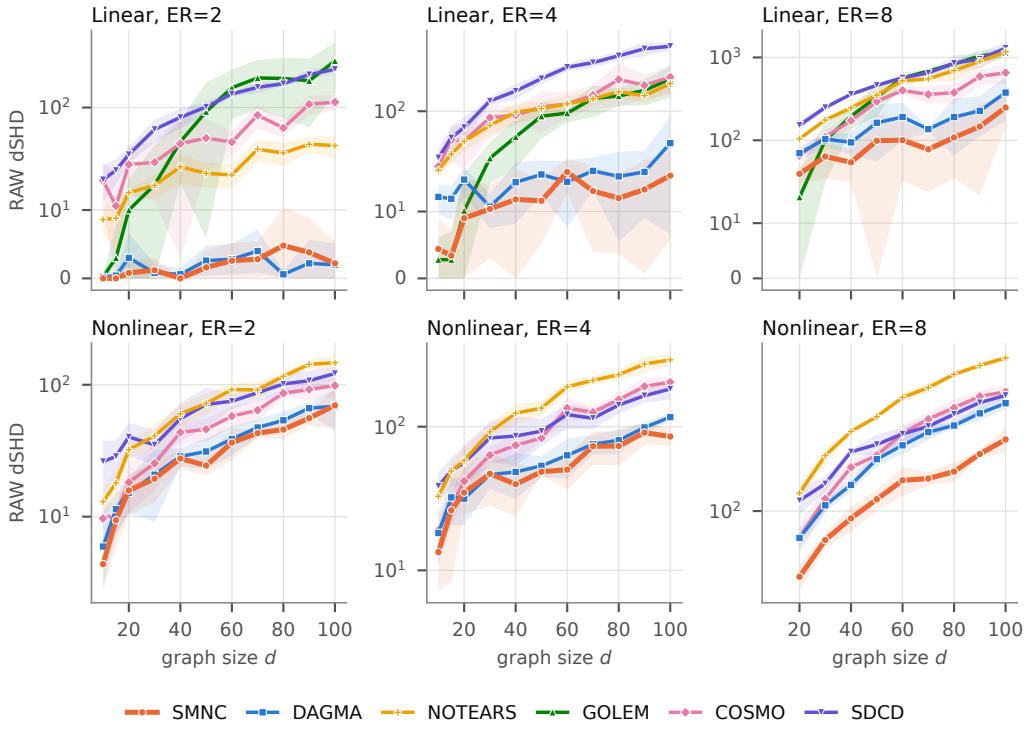

Figure 8: RAW dSHD across graph size d for ER $= 2, 4, 8$ (columns), Gaussian linear (top) and nonlinear (bottom) SEMs. Lines show the mean over 5 datasets and bands one standard deviation; lower is better. SMNC denotes bounded WG + SMNC (linear) and induced + SMNC (nonlinear). GOLEM was not run on the nonlinear SEMs.

Table 7: RAW dSHD at $d = 100$ (mean $\pm$ standard deviation over 5 datasets; lower is better). SMNC is bounded WG + SMNC (linear) and induced + SMNC (nonlinear); bold marks the best method in a row. NOTEARS is the DAGrad Adam implementation with the exponential penalty, untuned (Appendix D.2). GOLEM was not run on the nonlinear SEMs (–). GOLEM's RAW entries report its own post-hoc pruned output (Appendix D.2).

SEM	ER	s_0	SMNC	DAGMA	NOTEARS	GOLEM	COSMO	SDCD
Linear	2	200	2.20 ± 3.19	**2.00 ± 3.08**	42.60 ± 10.53	285.00 ± 148.04	112.80 ± 19.46	236.20 ± 17.84
	4	400	**22.80 ± 17.05**	48.00 ± 41.36	188.60 ± 32.79	209.80 ± 73.46	218.80 ± 71.45	444.00 ± 37.25
	5	500	**16.80 ± 6.30**	42.20 ± 16.86	247.60 ± 21.96	261.20 ± 81.94	259.20 ± 64.40	545.00 ± 39.52
	6	600	**20.80 ± 15.01**	88.20 ± 47.33	411.00 ± 83.81	529.40 ± 173.66	324.40 ± 77.75	682.60 ± 64.06
	7	700	**58.80 ± 22.17**	102.20 ± 47.01	637.40 ± 92.05	944.20 ± 72.19	424.00 ± 89.53	849.40 ± 32.16
	8	800	**248.40 ± 103.58**	376.80 ± 206.54	1170.80 ± 130.44	1136.80 ± 133.16	653.40 ± 170.38	1296.40 ± 227.89
Nonlinear	2	200	69.80 ± 23.86	**68.80 ± 23.53**	147.00 ± 21.59	–	98.60 ± 17.04	121.80 ± 14.46
	4	400	**85.40 ± 11.28**	116.60 ± 8.08	291.80 ± 24.47	–	204.20 ± 18.49	183.00 ± 28.47
	5	500	**116.40 ± 31.17**	160.40 ± 34.62	371.00 ± 41.52	–	269.00 ± 42.53	222.00 ± 15.08
	6	600	**127.00 ± 23.40**	204.20 ± 6.02	486.40 ± 20.28	–	324.00 ± 26.84	264.00 ± 32.07
	7	700	**173.60 ± 28.08**	290.60 ± 12.66	609.00 ± 18.28	–	412.40 ± 31.39	335.80 ± 16.59
	8	800	**247.00 ± 36.86**	390.40 ± 27.61	692.60 ± 26.41	–	450.80 ± 45.69	431.00 ± 15.89

Table 8: RAW F1-dir at $d = 100$ (mean $\pm$ standard deviation over 5 datasets; higher is better). Conventions as in Table 7.

SEM	ER	s_0	SMNC	DAGMA	NOTEARS	GOLEM	COSMO	SDCD
Linear	2	200	**0.99 ± 0.01**	**0.99 ± 0.01**	0.88 ± 0.03	0.58 ± 0.15	0.74 ± 0.04	0.56 ± 0.02
	4	400	**0.97 ± 0.02**	0.94 ± 0.05	0.72 ± 0.05	0.78 ± 0.06	0.75 ± 0.07	0.47 ± 0.05
	5	500	**0.98 ± 0.01**	0.96 ± 0.02	0.70 ± 0.02	0.78 ± 0.06	0.76 ± 0.05	0.43 ± 0.04
	6	600	**0.98 ± 0.01**	0.93 ± 0.04	0.60 ± 0.07	0.66 ± 0.09	0.76 ± 0.05	0.29 ± 0.10
	7	700	**0.96 ± 0.01**	0.93 ± 0.03	0.51 ± 0.05	0.49 ± 0.05	0.73 ± 0.05	0.09 ± 0.04
	8	800	**0.84 ± 0.06**	0.78 ± 0.11	0.32 ± 0.05	0.21 ± 0.06	0.65 ± 0.08	0.02 ± 0.01
Nonlinear	2	200	**0.83 ± 0.05**	0.82 ± 0.06	0.50 ± 0.09	–	0.69 ± 0.06	0.71 ± 0.03
	4	400	**0.89 ± 0.01**	0.84 ± 0.01	0.45 ± 0.07	–	0.67 ± 0.05	0.76 ± 0.04
	5	500	**0.88 ± 0.03**	0.82 ± 0.04	0.43 ± 0.11	–	0.64 ± 0.09	0.76 ± 0.01
	6	600	**0.89 ± 0.02**	0.80 ± 0.01	0.34 ± 0.04	–	0.64 ± 0.04	0.75 ± 0.04
	7	700	**0.87 ± 0.02**	0.75 ± 0.02	0.25 ± 0.04	–	0.59 ± 0.04	0.71 ± 0.02
	8	800	**0.82 ± 0.03**	0.69 ± 0.03	0.24 ± 0.05	–	0.62 ± 0.06	0.66 ± 0.02

Table 9: TOPK dSHD at $d = 100$ (mean $\pm$ standard deviation over 5 datasets; lower is better). TOPK keeps up to s_0 edges and is an oracle-sparsity diagnostic. Conventions as in Table 7.

SEM	ER	s_0	SMNC	DAGMA	NOTEARS	GOLEM	COSMO	SDCD
Linear	2	200	**1.60 ± 2.19**	2.00 ± 3.08	48.80 ± 13.01	94.00 ± 38.03	96.00 ± 17.32	155.20 ± 15.53
	4	400	**20.00 ± 14.35**	51.60 ± 42.76	227.60 ± 44.91	110.00 ± 30.17	183.20 ± 48.67	424.00 ± 44.07
	5	500	**19.20 ± 10.16**	43.40 ± 19.36	316.00 ± 25.69	153.60 ± 42.27	221.60 ± 60.60	580.00 ± 35.97
	6	600	**21.20 ± 16.41**	88.40 ± 46.40	498.40 ± 73.59	299.20 ± 114.88	280.40 ± 57.70	882.80 ± 112.81
	7	700	**49.20 ± 10.26**	101.80 ± 46.14	686.80 ± 72.30	656.80 ± 102.22	370.80 ± 66.28	1280.00 ± 36.33
	8	800	**246.00 ± 99.12**	348.60 ± 174.32	1109.60 ± 98.00	1231.20 ± 63.70	551.60 ± 116.66	1553.20 ± 20.43
Nonlinear	2	200	**61.00 ± 20.40**	82.80 ± 21.98	208.40 ± 23.00	–	114.80 ± 22.21	102.00 ± 11.58
	4	400	**85.60 ± 11.63**	137.60 ± 5.73	370.80 ± 54.20	–	196.80 ± 18.47	202.40 ± 23.38
	5	500	**116.40 ± 31.17**	192.00 ± 55.71	442.00 ± 84.30	–	250.80 ± 29.69	276.40 ± 13.52
	6	600	**127.00 ± 23.40**	229.60 ± 35.76	519.20 ± 32.91	–	291.20 ± 31.58	393.60 ± 53.56
	7	700	**173.60 ± 28.08**	292.00 ± 17.44	650.00 ± 25.02	–	371.60 ± 21.65	502.40 ± 44.57
	8	800	**247.00 ± 36.86**	369.60 ± 14.79	692.00 ± 81.30	–	430.00 ± 24.33	665.60 ± 50.43

D.2 Baseline protocol

Baseline optimization budgets, numerical precision, and hyperparameters were not matched to our own method's protocol. We disclose the re-verified configuration behind the linear DAGMA, GOLEM, COSMO and NOTEARS numbers and the nonlinear DAGMA and NOTEARS numbers in Table 10. Every entry below was re-verified against the configuration recorded with each method's own run outputs and against the defaults of the benchmark driver that invoked the method. The table notes give the sources. The SDCD configuration was not archived. The SDCD numbers are consistent with SDCD's default settings applied to the unstandardized benchmark data. Such a run would be outside SDCD's documented protocol, which z-scores each variable before fitting.

Provenance of the SMNC numbers. The SMNC entries in Table 1 and in Appendix D.1 come from reruns of SMNC on the benchmark datasets. We archived the configuration, the dataset file and the seed of every rerun, so each reported cell can be recomputed from its `metrics.json`. They replace the values in the reviewed version, which came from runs whose configuration files were not archived. The reviewed and rerun values agree to within one reported standard deviation in every dSHD cell, RAW and TOPK, and the differences have no consistent sign.

For the linear SEMs, the archived configuration sets some values per dataset. The optimizer is Adam with learning rate 5×10^{-3} and betas $(0.99, 0.999)$; the magnitude bound is $\beta = 5$, and training runs $T = 5$ stages with $s = 1.0, 0.9, 0.8, 0.7, 0.6$. On 283 of the 305 datasets it uses $\lambda_1 = 10^{-2}$. It uses $\lambda_1 = 1.2 \times 10^{-2}$ on 2 datasets and 1.25×10^{-2} on 15, all with $d \in \{60, \ldots, 90\}$ and ER$\in \{6, 8\}$, and $\lambda_1 = 3 \times 10^{-2}$ on the five $d = 100$, ER$= 8$ datasets. The magnitude threshold is 0.1 on 300 datasets and 0.3 on those five. The first four stages run 2×10^4 steps each when $d \geq 40$ and ER≥ 5 (140 datasets), and 10^4 steps otherwise (165 datasets). The last stage always runs 10^4 steps. The cycle weight μ_{cycle} is multiplied by 0.1 after each stage, and the cycle term is applied at every second gradient step. The configuration sets $(\mu_{\text{init}}, \mu_{\text{cycle}}) = (10, 0.1)$ on 289 datasets, and $(0.3, 0.3)$ on the 16 datasets whose sample covariance has a condition number of at least 3.2×10^8. No dataset lies on

that cut: the two nearest lie at 3.08×10^8 and 3.14×10^8 below it and at 3.31×10^8 and 3.46×10^8 above it. The tolerance of the barrier's inverse-nonnegativity check is 0. For the nonlinear SEMs, it uses Adam with learning rate 2×10^{-4}, hidden width 20, $\lambda_2 = 5 \times 10^{-3}$, threshold 0.3, $T = 5$ stages of 10^4 steps each, $s = 1$, and $(\mu_{\text{init}}, \mu_{\text{cycle}}) = (0.3, 0.1)$, with μ_{cycle} multiplied by 0.1 after each stage and the cycle term applied at every step. It uses $\lambda_1 = 1.25 \times 10^{-2}$ on the five $d = 100$, ER$= 5$ datasets and $\lambda_1 = 10^{-2}$ on the other 300. The archive records these values for every dataset but not how they were chosen: how the per-dataset λ_1 and threshold values and the condition-number switch were selected is not recorded.

Compute resources. NOTEARS was run through DAGrad on an NVIDIA A30 GPU; at $d = 100$ its median fit time was 29.9 s for linear and 1332 s for nonlinear SEMs. DAGMA's median fit time at $d = 100$ was 42.6 s (linear) and 705.7 s (nonlinear); the GPU model was not recorded with these times. The SMNC runs on the Sachs data used an NVIDIA A30 with PyTorch 2.4.0, and their run time was not recorded. The hardware, run time and memory of the synthetic SMNC runs were not recorded, and neither were the total compute and the compute of preliminary or failed runs.

Licences. DAGMA and NOTEARS are released under the Apache-2.0 licence, and SDCD under the MIT licence. DAGrad, through which the NOTEARS numbers were produced, has a non-commercial licence. GOLEM's upstream code is Apache-2.0, but we did not confirm the licence of the copy we ran, and we did not verify a licence for the COSMO code. The Sachs data [Sachs et al., 2005] come with no stated licence; our runs read a laboratory CSV copy of them, and the 17-edge reference DAG is the one distributed with bnlearn (Section 4.3).

Scope of the attribution. No benchmark in this paper isolates scale normalization from the closure head and the factorization: raw closure is compared with SMNC only in the controlled probes of Appendix A.3, and the benchmark comparison with DAGMA also differs in iteration budget, hyperparameters and, for nonlinear SEMs, network width (Table 10).
We also ran DAGMA on the 45 linear ER$= 8$ datasets ($d \in \{20, 30, \ldots, 100\}$) with the archived SMNC settings. These cover λ_1, the threshold, the learning rate, the step counts and the μ and s schedules. DAGMA's mean RAW dSHD was 215.3, against 172.5 with the settings in Table 10 and 104.2 for the SMNC rerun. The SMNC rerun had a lower dSHD than the matched DAGMA run on 37 of the 45 datasets.
A trial point is outside the domain of h_s in Eq. (7) when $s \leq 0$ or $\rho(A \circ A) \geq s$. A positive determinant does not certify the domain: $sI - A \circ A$ must be a nonsingular M-matrix. Linear DAGMA, as run from the authors' solver, rejects a step when the inverse of $sI - A \circ A$, offset by 10^{-16}, has a negative entry, which tests the M-matrix domain. Nonlinear DAGMA evaluates $-\log|\det(sI - A \circ A)| + d \log s$ and rejects a step only when that value is negative, and our nonlinear SMNC solver rejects a step only when the value is non-finite or negative. Both can therefore accept points outside the domain: a point with a negative determinant, or with a positive determinant but a nonpositive principal minor, gives a finite value that is not the barrier above. Our linear SMNC solver uses the inverse test, with tolerance 0 in the archived runs. GOLEM's acyclicity penalty is finite for every real adjacency, so it does not reject steps for leaving this domain. COSMO parameterizes an acyclic orientation from node priorities, so every parameter value stays in that class and there is no log-det domain to leave. SDCD constrains a spectral radius rather than h_s, so it has no log-det domain and does not reject steps on these grounds.

D.3 Sachs reachability-cue construction

Training protocol and reproduction. The reported runs use the pooled, column-wise z-scored Sachs data, seed 0, four outer stages, initial $\mu_{\text{init}} = 0.1$ with factor 0.1, and output threshold 0.3. The cycle-enabled variants use initial $\mu_{\text{cycle}} = 0.3$ with factor 0.1; the active soft losses use initial $\mu_{\text{rel}} = 0.5$ with factor 0.99. The cycle-off row uses the same internal nonlinear backbone with $\mu_{\text{cycle}} = 0$. Archived CPU/GPU reruns reproduce the thresholded directed graphs of all four cycle-enabled 12-cue/no-cue variants. These are fixed-seed reproductions, not additional independent-seed trials; the cycle-off row was not included in that reproduction study.

Reference network. We use the standard 17-edge Sachs reference DAG $\mathcal{G}_{17}$ distributed with `bnlearn` (`https://www.bnlearn.com/research/sachs05/`) on the eleven variables Raf, Mek, Plcg, PIP2, PIP3, Erk, Akt, PKA, PKC, P38, Jnk. Its edges are PKC→PKA, Raf, Mek, P38, Jnk; PKA→Raf, Mek, Erk, Akt, P38, Jnk; Raf→Mek; Mek→Erk; Erk→Akt; Plcg→PIP3, PIP2; PIP3→PIP2.

Table 10: Verified optimization protocol for the benchmark baselines. "Own-code default" is the budget or behavior the method's own vendored script uses when no override is passed. "Used" is the value actually recorded with the runs behind the reported numbers. SMNC's own protocol is stated in Sections 4.2 and 4.2, and its archived per-dataset settings are given in this appendix.

Method	Steps used (vs. own default)	Precision	RAW col.	Notes & hyperparameters
DAGMA (linear)	90,000 (4×20k warm + 10k final) vs. 180,000 (4×30k + 60k)	float64	thresh.	$w_{\text{threshold}} = 0.3$; λ_1=0.03, lr= 3×10^{-4} are the vendored script's own defaults, unmodified; only the iteration budget was reduced.[a]
DAGMA (nonlinear)	40,000 (3×10k warm + 10k final) vs. 230,000 (3×50k + 80k)	float64	thresh.	MLP with one hidden layer of width 10, as in DAGMA's own nonlinear example; $\lambda_1 = 0.02$, $\lambda_2 = 0.005$, lr= 2×10^{-4}, $T = 4$, $\mu_{\text{init}} = 0.1$, μ factor 0.1, $s = 1$, $w_{\text{threshold}} = 0.3$, all equal to the defaults of DAGMA's nonlinear `fit`; only the iteration budget was reduced.[d]
NOTEARS (DAGrad)	5×1,000 Adam steps vs. up to 100 outer L-BFGS-B solves in NOTEARS's own code	float64	thresh.	DAGrad's PyTorch implementation of the NOTEARS objective with the exponential penalty $\operatorname{tr} e^{A\circ A} - d$ and Adam, **not** the original SciPy L-BFGS-B solver. Linear: ℓ_2 loss, $\lambda_1 = 0.1$, lr= 10^{-3}, $h_{\text{tol}} = 10^{-8}$, $w_{\text{threshold}} = 0.3$. Nonlinear: a one-hidden-layer sigmoid MLP of width 40, $\lambda_2 = 0.01$, same budget and threshold. These are the runner's defaults, the same for every dataset.[e]
GOLEM	50,000 vs. 100,000 (half the script's default)	float32 (script default)	**proc.**	**RAW columns report GOLEM's post-hoc pruned ("processed") output.** Its raw output is dense (e.g. directed-edgewise SHD 9500 for one d=100, ER=4 run, vs. 209.80 ± 73.46, the table's processed-output mean over 5 datasets at that setting). $\lambda_1 = 0.02$, $\lambda_2 = 5.0$ are the script's own equal-variance defaults, unmodified; only the iteration budget was reduced.[b]
COSMO	2,000 epochs, batch 64, at one fixed config; n/a own default (grid-search method)	float64	thresh.	**evaluated at one fixed hyperparameter point**: the geometric midpoint of each parameter's published search range (lr= 3.16×10^{-3}, ℓ_1-reg= 3.16×10^{-4}, ℓ_2-reg= 2.24×10^{-3}, priority-reg= 1.73×10^{-3}, temperature= 7.07×10^{-4}, shift= 10^{-2}), with threshold= 0.3. COSMO's own documented protocol runs a grid search per dataset and keeps the configuration with the best validation ROC-AUC.[c]

[a] Steps used are the benchmark driver's defaults (`warm_iter`= 20000, `max_iter`= 10000, $T = 5$) passed to the DAGMA authors' `DagmaLinear.fit`, whose own documented defaults are `warm_iter`= 3×10^4, `max_iter`= 6×10^4; the outer loop runs $(T-1) \cdot$ `warm_iter` + `max_iter` steps.

[b] GOLEM's run script defaults to 10^5 iterations in `float32` with equal-variance $\lambda_1 = 2 \times 10^{-2}$, $\lambda_2 = 5.0$; every archived run records `float32` and 5×10^4 iterations. GOLEM records both its raw and its processed output; the RAW columns use the processed one.

[c] COSMO's benchmark driver hardcodes the six values listed above, each the geometric midpoint of the log-uniform search range published with COSMO's code; COSMO's documented grid-search-and-validate invocation is not called. `threshold`, `batch_size`, and `max_epochs` are fixed generic parameters in both COSMO's grid file and its default configuration.

[d] All 305 archived nonlinear DAGMA runs (61 settings with 5 datasets each) record the same configuration (hidden width 10, `warm_iter`=`max_iter`= 10^4, $T = 4$); DAGMA's `DagmaNonlinear.fit` defaults to `warm_iter`= 5×10^4 and `max_iter`= 8×10^4, and its outer loop runs $(T-1) \cdot$ `warm_iter` + `max_iter` steps. The MLP width has no default in DAGMA's code; width 10 is the value used in DAGMA's own example. The numerical precision is not recorded with these runs.

[e] All 305 archived linear and 305 archived nonlinear DAGrad runs (61 settings with 5 datasets each) record this configuration and ran on an NVIDIA A30. On 5 linear runs, at $(d, \text{ER}) \in \{(40, 2), (40, 8), (50, 5), (70, 2), (90, 2)\}$, the returned h exceeds the tolerance 10^{-8} and the RAW output is not a DAG; these runs are scored as returned. The runs record neither the DAGrad nor the NOTEARS code revision. NOTEARS's TOPK column is the oracle edge-count projection of its raw weights, a diagnostic and not the baseline's standard output.

Cue sources. A cue $i \rightsquigarrow j$ means that i is upstream of j in the closure sense, possibly through a multi-hop path. Cues come from matched intervention/reference comparisons in the continuous Sachs conditions: Mek (`u0126`), Akt (`aktinhib`), PKC (`g0076`) and PIP2 (`psitect`) interventions, each against its matching `cd3cd28` or `cd3cd28_icam2` background, and PKA activation (`b2camp.csv`) against both backgrounds. The PKA rows use a PKA-activation condition and not a stimulus-matched background, so we treat that evidence as weaker. For each target i and response $j \neq i$ we compare X_j under the intervention and under the matched reference by the Kolmogorov–Smirnov statistic D_{ij}, the robust median effect $E_{ij} = (\operatorname{median} X_j^{\mathrm{int}} - \operatorname{median} X_j^{\mathrm{ref}})/(\mathrm{IQR}\, X_j^{\mathrm{ref}} + 10^{-8})$, and $S_{ij}^{\mathrm{stat}} = D_{ij}|E_{ij}|$. Pairs are retained when $D_{ij} \geq 0.20$, $|E_{ij}| \geq 0.30$ and $S_{ij}^{\mathrm{stat}} \geq 0.20$, and pairs seen in several backgrounds are ranked by $\max_b S_{ij}^{\mathrm{stat},(b)} + 0.25 \operatorname{mean}_b S_{ij}^{\mathrm{stat},(b)} + 0.20\, n_{\mathrm{bg}}(i,j)$, where n_{bg} is the number of supporting backgrounds.

Reference-filtered 12-cue set. This step is oracle-assisted: we use the reference closure to filter unstable directions. Akt and PIP2 are excluded as cue sources, because they produced more no-path or reverse-direction candidates in preliminary screening; they may still be responses. The set takes the seven highest-scoring PKC-sourced pairs, the two Mek-sourced pairs whose responses are downstream of Mek in the reference (Erk, Akt), and the three highest-scoring PKA-sourced pairs from `b2camp.csv` (PKC is excluded as a response because it is upstream of PKA in $\mathcal{G}_{17}$); responses are restricted to the eight protein-kinase nodes (excluding Plcg, PIP2, PIP3). The 12 relations are PKC$\rightsquigarrow$Mek, P38, Akt, Raf, Jnk, Erk, PKA, Mek$\rightsquigarrow$Akt, Erk and PKA$\rightsquigarrow$Mek, Jnk, P38. All are consistent with the transitive closure of $\mathcal{G}_{17}$, so we describe this as a data-induced, reference-filtered weak-cue experiment and not as unsupervised cue extraction.

Encoding as closure-side supervision. For the positive-only variant, the active target set consists of the 12 selected forward pairs with target 1, and no hard exclusions are applied. The hard-only variant uses the same pairs to forbid reverse direct edges but has no active relation loss. The hard+soft variant applies these hard exclusions and supervises both the 12 forward targets at 1 and the 12 reverse targets at 0. Let M_{ij}^{cue} be the binary supervision mask and O_{ij}^{cue} its binary target. For either active soft loss, the implementation averages BCE over the supervised entries:

$$\mathcal{L}_{\mathrm{rel}} = \frac{\sum_{i \neq j} M_{ij}^{\mathrm{cue}}\, \ell_{\mathrm{BCE}}(S_{ij}, O_{ij}^{\mathrm{cue}})}{\sum_{i \neq j} M_{ij}^{\mathrm{cue}}}.$$

Here S_{ij} is the SMNC reachability logit. In the 12-cue comparison, the denominators are 12 for positive-only and 24 for two-sided supervision. Thus, comparing positive-only with hard+soft changes the hard mask, reverse targets, and per-target weighting; it is not a strict factorial comparison with an identical soft objective. None of these cues requires a positively supervised pair to be a direct edge.

D.4 Nonlinear reachability-cue ablation

Table 11 compares No cues, Hard-only, Soft-only, and Hard+Soft in the nonlinear SMNC model. Cue-assisted variants use oracle ancestry. For each $i \rightsquigarrow j$, Hard-only forbids the reverse direct edge $j \to i$, while Soft-only assigns forward and reverse closure targets of 1 and 0 without hard exclusion. Hard+Soft combines both operations. A positive reachability target does not require a direct edge $i \to j$.

We use $d = 100$, ER$\in \{2, 4, 5, 6, 7, 8\}$, and five matched datasets per setting, with unstandardized and standardized inputs. All variants use $\mu_{\mathrm{init}} = 0.3$, with initial $\mu_{\mathrm{rel}} = 0.01$ for soft supervision. The shared output thresholds $t \in \{0.3, 0.5\}$ are applied to the same trained matrices. No cues is this ablation's control; the main cue-free benchmark uses a separate protocol.

Adding soft supervision to hard exclusion yields more consistent recovery gains at $t = 0.5$. At $t = 0.3$, gains occur in denser settings, while sparse-setting outcomes are mixed. These comparisons are descriptive: the supervision weight and thresholds were explored using synthetic ground truth without independent validation.

Table 11: Nonlinear soft-supervision ablation with oracle ancestry cues at $d = 100$. Entries report mean $\pm$ sample standard deviation over five matched datasets. Both soft variants use two-sided closure targets. Bold marks the best mean for each metric within a row, including ties.

		No cues		Hard-only		Soft-only		Hard+Soft	
t	ER	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑	dSHD ↓	F1-dir ↑
(a) Unstandardized inputs									
0.3	2	65.8 ± 22.3	0.839 ± 0.051	**61.8 ± 15.8**	**0.847 ± 0.040**	80.6 ± 24.6	0.817 ± 0.051	75.0 ± 16.4	0.827 ± 0.038
	4	87.6 ± 12.0	0.889 ± 0.014	**81.4 ± 13.9**	**0.897 ± 0.017**	88.0 ± 11.4	0.891 ± 0.012	84.8 ± 16.6	0.895 ± 0.020
	5	117.2 ± 31.5	0.876 ± 0.034	102.0 ± 27.5	0.892 ± 0.029	116.2 ± 31.2	0.879 ± 0.033	**100.2 ± 33.1**	**0.896 ± 0.035**
	6	127.6 ± 21.0	0.886 ± 0.020	115.8 ± 24.7	0.897 ± 0.024	128.0 ± 20.6	0.888 ± 0.020	**113.6 ± 23.7**	**0.901 ± 0.022**
	7	179.6 ± 28.7	0.860 ± 0.024	153.6 ± 31.7	0.880 ± 0.027	178.4 ± 22.5	0.865 ± 0.019	**146.4 ± 27.3**	**0.888 ± 0.023**
	8	244.2 ± 37.9	0.825 ± 0.031	216.4 ± 30.3	0.848 ± 0.023	229.6 ± 35.8	0.838 ± 0.028	**200.8 ± 24.6**	**0.862 ± 0.018**
0.5	2	47.6 ± 18.5	0.873 ± 0.047	**42.2 ± 10.2**	0.887 ± 0.029	44.6 ± 20.5	0.886 ± 0.051	**42.2 ± 13.6**	**0.892 ± 0.036**
	4	86.6 ± 8.7	0.881 ± 0.013	80.2 ± 11.9	0.890 ± 0.018	**76.4 ± 4.2**	**0.897 ± 0.007**	76.8 ± 16.0	0.896 ± 0.024
	5	143.6 ± 14.7	0.835 ± 0.020	130.4 ± 12.4	0.851 ± 0.016	138.4 ± 20.2	0.843 ± 0.026	**121.2 ± 16.8**	**0.863 ± 0.021**
	6	183.2 ± 8.4	0.823 ± 0.010	164.6 ± 15.2	0.843 ± 0.017	168.0 ± 12.1	0.840 ± 0.013	**156.0 ± 14.2**	**0.852 ± 0.015**
	7	278.4 ± 25.6	0.756 ± 0.027	252.4 ± 32.0	0.781 ± 0.034	263.6 ± 17.8	0.773 ± 0.018	**237.8 ± 34.8**	**0.796 ± 0.036**
	8	420.0 ± 75.3	0.642 ± 0.092	385.0 ± 56.5	0.682 ± 0.062	403.2 ± 77.5	0.662 ± 0.092	**359.6 ± 56.1**	**0.709 ± 0.060**
(b) Standardized inputs									
0.3	2	233.4 ± 11.3	0.442 ± 0.034	**196.2 ± 17.2**	0.521 ± 0.041	254.8 ± 11.2	0.462 ± 0.024	201.8 ± 15.3	**0.560 ± 0.026**
	4	360.6 ± 13.5	0.505 ± 0.018	305.8 ± 15.3	0.574 ± 0.025	376.4 ± 15.6	0.515 ± 0.020	**301.6 ± 15.8**	**0.607 ± 0.017**
	5	440.4 ± 40.9	0.481 ± 0.035	360.6 ± 30.3	0.574 ± 0.031	433.0 ± 54.9	0.514 ± 0.049	**356.8 ± 42.4**	**0.602 ± 0.040**
	6	506.4 ± 40.8	0.482 ± 0.040	418.0 ± 48.3	0.570 ± 0.055	496.4 ± 38.9	0.523 ± 0.038	**403.6 ± 30.6**	**0.608 ± 0.029**
	7	572.0 ± 20.3	0.485 ± 0.028	485.6 ± 22.6	0.562 ± 0.028	552.2 ± 20.0	0.533 ± 0.026	**454.2 ± 22.9**	**0.615 ± 0.025**
	8	651.8 ± 49.4	0.477 ± 0.063	547.2 ± 32.2	0.560 ± 0.041	650.2 ± 39.8	0.510 ± 0.045	**521.4 ± 38.1**	**0.600 ± 0.031**
0.5	2	209.4 ± 8.6	0.421 ± 0.039	177.2 ± 14.0	0.499 ± 0.041	211.4 ± 8.6	0.462 ± 0.024	**174.6 ± 17.3**	**0.547 ± 0.040**
	4	332.2 ± 12.9	0.468 ± 0.016	302.2 ± 11.6	0.520 ± 0.024	337.2 ± 11.5	0.478 ± 0.016	**290.0 ± 17.7**	**0.555 ± 0.034**
	5	419.8 ± 21.4	0.420 ± 0.040	373.2 ± 20.9	0.492 ± 0.038	404.6 ± 24.1	0.454 ± 0.038	**361.8 ± 22.1**	**0.525 ± 0.033**
	6	518.4 ± 28.9	0.376 ± 0.044	466.0 ± 26.4	0.444 ± 0.040	499.4 ± 26.9	0.419 ± 0.043	**449.8 ± 17.1**	**0.479 ± 0.027**
	7	596.0 ± 22.7	0.363 ± 0.034	546.0 ± 17.1	0.424 ± 0.026	573.2 ± 17.3	0.410 ± 0.028	**514.0 ± 21.5**	**0.483 ± 0.028**
	8	683.0 ± 41.5	0.351 ± 0.059	620.4 ± 31.8	0.418 ± 0.047	664.2 ± 34.7	0.391 ± 0.047	**605.0 ± 32.5**	**0.444 ± 0.046**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: *The abstract and the Contributions paragraph separate the score lower bound (Theorem 3.2) from the conditional hard-readout gradient factor (Corollary 3.3, unique-active-cycle assumption, fixed d, Section 3.2), and they also state the bounded factorization (Eq.* (1)*), the SMNC head (Section 3.1), and cue-free gains over DAGMA (Table 1). Section 3.2, the method text, Limitations, and Appendices C.4 and A.5 state that exact theorems and gradient envelopes do not apply to* `float32` *shifted-GEMM linear training or to the nonlinear clamped head.*

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: *The Limitations section opens with numerical closure issues (shifted* `float32` *GEMM underflow, nonlinear clamped GEMM; Appendices C.4 and A.5); it also states that gradient-scaling results are local active-candidate statements whose $\Theta(1/L)$ factor holds only at the minimal covering scale; that empirical gains are mainly on unstandardized Gaussian benchmarks; causal-interpretation assumptions; cue limitations; and hyperparameter robustness. Sachs single-seed scope is in Section 4.3 and Table 2.*

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: *Every theorem, lemma, corollary and proposition states its assumptions and has a numbered proof: Theorem 3.1 in Appendix C.2; Theorem 3.2 in Appendix C.3; Corollary 3.3, Lemma C.1 and Lemma C.2 in Appendix C.5; Proposition C.4 in Appendix C.6. The gradient-factor results carry the dominant/unique active-candidate assumption in their statements. The nonlinear implementation's clamped GEMM is disclosed as a case in which Theorem 3.1 and Lemma C.2 do not hold (Appendix A.5), and the "Objective used in our experiments" paragraph and the Remark in Appendix C.5 state which readout each benchmark objective uses and what the theorems cover for it.*

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: *Sections 4.2 and 4.2 specify the data-generating process, the optimizer schedule and the SMNC readout used in each regime. Appendix D.2 gives the per-dataset SMNC settings from the archived configurations and the budgets, precision and hyperparameters of each baseline (Table 10). The answer is [No] because how the per-dataset hyperparameters and the condition-number switch were chosen is not recorded, and the SDCD configuration was not archived, as stated there.*

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: *The paper contains no working code or data link; the only external URL is the Sachs reference DAG in Section 4.3. The authors will release the code and the archived per-dataset configurations at* `https://github.com/oOLearnCodeOo/causalclosure`. *The repository is not public at the time of writing, so the answer is [No].*

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: *Sections 4.2 and 4.2 give the protocol: the data-generating settings, the DAGMA-style outer loop and its stagewise schedule, which SMNC readout drives the cycle loss (final-scale for linear, soft all-scale for nonlinear), and the evaluation (RAW vs. TOPK,*

pair-space and directed edgewise SHD). Appendix D.2 states the optimizer, learning rates, μ_{cycle}, β, thresholds and the nonlinear solver settings for the benchmark runs. The answer is [No] because how the per-dataset hyperparameters and the condition-number switch were chosen is not recorded.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: *Tables 1 and 11, together with the compact tables in Appendix D.1, report mean $\pm$ sample standard deviation over the $N_{\text{rep}} = 5$ independently sampled datasets per (d, s_0) setting (Section 4.2 and the table captions), and the appendix overview figure plots mean $\pm$ one standard deviation over the same datasets (Appendix D.1). The source of variability is the independent dataset draw, not only initialization. The Sachs results (Table 2) are single-seed (seed$= 0$) and carry no error bars, as stated in Section 4.3 and the caption; this is a limitation of the real-data evaluation separate from the 5-dataset synthetic benchmarks.*

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: *Appendix D.2 (Compute resources) gives the hardware and median fit times where they were recorded: NOTEARS through DAGrad (NVIDIA A30; at $d = 100$, 29.9 s linear and 1332 s nonlinear), DAGMA (42.6 s linear and 705.7 s nonlinear, GPU model not recorded) and the Sachs SMNC runs (NVIDIA A30, PyTorch 2.4.0). The hardware, run time and memory of the synthetic SMNC runs, the total compute, and the compute of preliminary or failed runs were not recorded, so the answer is [No].*

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: *The paper uses only synthetic data generated by the authors and the public, de-identified Sachs protein-signaling dataset (Section 4.3), involves no human subjects, crowdsourcing, or personal data, and the Impact Statement discusses appropriate-use considerations for causal-discovery outputs.*

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: *The Impact Statement (immediately before the references) discusses the intended benefit of the method and the main risk of misuse: as with other causal-discovery methods, inferred graphs are hypotheses and should be combined with domain expertise, robustness checks and interventional validation before high-stakes use.*

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: *The paper releases no pretrained models, generative models, or scraped datasets; it studies a structure-learning algorithm evaluated on synthetic data and the public Sachs dataset, which pose no dual-use or high-misuse risk.*

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: *The licences of the baseline code we ran are listed in Appendix D.2 (Licences): DAGMA and NOTEARS Apache-2.0, SDCD MIT (the licence file of the SDCD routine adapted for the supplementary figure code is included,* `figure_src/sdcd_two_cycle/LICENSE_SDCD.md`*), and DAGrad non-commercial. We did not confirm the licence of the GOLEM copy and did not verify a licence for the COSMO code. The Sachs protein-signaling data [Sachs et al., 2005] and the* `bnlearn` *17-edge reference DAG (Section 4.3) are used without a stated licence or terms of use. The answer is [No] for these reasons.*

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: *The paper does not release a new dataset, pretrained model, or similarly packaged asset; its contribution is an algorithm/method (SMNC and the $W_f \odot G$ factorization) fully described in the text itself (Section 3.1) and summarized as pseudocode in Algorithm 1 (Appendix C.1).*

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: *The paper involves no crowdsourcing or human-subjects research; all experiments use synthetic SEM data and the pre-existing, de-identified Sachs protein-signaling dataset (Section 4.3).*

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: *The paper involves no human-subjects research requiring IRB review; see the Crowdsourcing/human-subjects item above.*

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: *LLMs are not an important, original, or non-standard component of the core method: SMNC and the $W_f \odot G$ factorization (Section 3.1) are a closure/semiring-based differentiable construction with no LLM in the modeling pipeline, and the reported experiments (Section 4.2–4.2) do not use an LLM as part of data generation, training, or evaluation.*

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.