HIDDEN EQUIVALENCES IN SHEAF NEURAL NETWORKS: THE SHARE CREDITED TO MATRIX-VALUED TRANSPORT IS NOT INTRINSIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Sheaf neural networks replace scalar edge weights with learned restriction maps, and their gains are routinely credited to this matrix-valued transport. We show that the share of the gain so credited is not intrinsic to a task or an architecture. The algebra its restriction maps generate constrains the propagation operator of a sheaf layer: by the Wedderburn decomposition it fixes the block structure and which blocks act irreducibly, so scalar maps give a real edge-weighted operator, one-angle rotations a complex-weighted (magnetic) one, and of our six rungs only the dense one acts irreducibly. It does not fix the realizable family, since restricted families can generate the same algebra. On this chain of algebras we build an attribution protocol: an exactly nested six-level ladder audited in function space, a depth-matched no-graph baseline, and a necessity diagnostic that can refuse to answer. The share credited to matrix-valued transport is then 53% and 11% on two heterophilous tasks at two layers and 13% and 17% at four, and it moves with width (46% to 73%), with the operator (11% to 35%) and with the order of credit (13% to 94%). The released operator moves the absolute sheaf-exclusive gain by less than 0.01 at two layers (a point estimate, not an equivalence test) but reverses the sign of the four-layer depth effect on Minesweeper (-0.034 against $+0.035$), four fifths of which a 2×2 experiment traces to the normalization alone, acting through the scalar gate as a spectral bound we prove predicts. What the scalar control is allowed to do matters as much as what the sheaf adds: in a preregistered test, a learned degree-3 spectral response on those rungs absorbs 92% of the gain credited to matrix-valued transport on one task, while on the other the interaction runs the other way and is not resolved under a matched training budget. Two later controls on four-layer Minesweeper reduce the gain credited to matrix-valued transport by 98% with learned one-hop filtering and by 71% with fixed three-hop filtering. The larger reduction therefore requires no wider receptive field. On an independently published directed sheaf architecture the capacity effect is -29.6 points unconditionally, positive conditional on convergence, and not significant once an identity-containing parameterization removes the collapse. Claims about what matrix-valued transport contributes should therefore state the operator, depth and order of credit under which they were measured.

1 INTRODUCTION

Message passing fixes *who* exchanges information and largely fixes *how*. Cellular sheaves relax the second half: each incidence carries a learned linear restriction map, so neighboring features need only agree after relation-specific transport (Hansen & Ghrist, 2019; Hansen & Gebhart, 2020; Bodnar et al., 2022). The resulting models are strictly more expressive, and a growing literature reports gains on heterophilous, directed and higher-order tasks (Bodnar et al., 2022; Barbero et al., 2022a; Duta et al., 2023; Mule et al., 2026; Bamberger et al., 2025).

Expressivity is not an inductive bias, and a performance gain is not an attribution. Two problems recur in how such gains are reported (Appendix G tabulates ten sheaf-network papers: of the four that compare learned restriction-map families on real benchmarks, one reports trivial-sheaf equivalents of the same model, and none of the ten reports, on its real benchmarks, the same model with

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

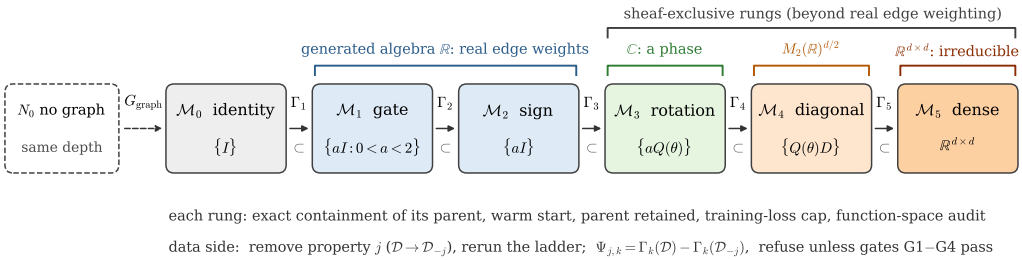


Figure 1: The attribution protocol. Increments Γ_k are taken along an exactly nested chain, and each rung is labelled by the algebra its maps generate (Theorem 1): the first two rungs are real edge weighting, the third is a phase, and only the dense rung acts on the stalk irreducibly. A depth-matched no-graph model fixes where graph value turns positive, and a data-side intervention asks which rung depends on a given property.

the graph removed). **(P1) Bundled capabilities.** A restriction map can attenuate a message, invert its sign, or mix coordinates. Attenuation is available to ordinary learned edge weighting and sign inversion to signed propagation; only coordinate mixing requires matrix-valued transport. A comparison that activates several of these at once cannot say which produced the gain, and the familiar list "scalar, diagonal, orthogonal, dense" is not even a chain: diagonal and orthogonal are incomparable branches, so differences along it are not marginal capacity effects. **(P2) Nominal rather than effective capability.** Even a correct inclusion chain is stated over raw restriction-map families, whereas the realized function is the composition of operator, residual update, normalization and nonlinearity. We show this gap is not a technicality.

Recent work has started to ask whether learned sheaves are needed at all (Hernandez Caralt et al., 2026; Liu, 2026; Grover & Bourgerie, 2026). These studies intervene on the model: they replace or freeze maps and measure the loss. We complement them on two sides: we characterize exactly when two nominally different capabilities are the same function, and we make the attribution protocol itself auditable, including a data-side intervention with explicit conditions under which it must refuse to answer. Nothing in the protocol is specific to sheaves. Any graph model whose edges carry learned transport, whether weights, gates, signs, phases or per-edge matrices, admits the same three questions: which algebra its maps generate, whether its nominal families are nested as functions, and whether its gain survives a depth-matched no-graph baseline. Sheaf networks are the case in which all of these rungs coexist in one parameterization.

Contributions. **(i) Exact collapse results** (Section 3). Normalization makes a layerwise constant restriction scalar and a learned propagation strength the same function family (Proposition 1); the statement holds for the augmented symmetric normalization of released sheaf code, exactly and on any graph (Proposition 2); more generally every scalar-valued sheaf layer factors as an edge-weighted scalar operator $M \otimes I_d$ (Theorem 1). Averaging the couplings of two independently parameterized directions turns orthogonal maps into attenuators; released implementations are unaffected (Appendix E). **(ii) An audited capability ladder** (Section 4): an exactly nested six-level chain, warm-started rung by rung, with an effective-function audit. Non-nested optimization alone moves a measured capacity effect by up to 0.043 in *either* direction. **(iii) A necessity diagnostic that can refuse** (Section 4.1), calibrated on a known-truth battery in which the raw interaction cannot distinguish necessity from intervention collapse. **(iv) Attribution results** (Section 5). The share of the gain owed to sheaf-exclusive capabilities is not an intrinsic quantity: it is 53% versus 11% on two real tasks at two layers and 13% versus 17% at four, and it moves with width, with the operator and with the order of credit. At two layers the absolute sheaf-exclusive gain is not detectably changed when our operator is replaced by that of released code; at four layers the operator decides, on one task, even the direction of the depth effect, and a 2×2 experiment traces most of that to the normalization acting through the scalar gate. Under a data-side intervention the total effect on the full model is not detectably changed by depth under both operators, while the rungs that collect it depend on the operator (Figure 7). A preregistered test shows that most of that share, on one task, is a substitute for a spectral freedom the scalar control was denied. On a published directed

sheaf architecture the sign of the measured capacity effect depends on whether one conditions on convergence. A directional prediction of our own is not supported by the data and is reported as such (Appendix A).

2 SETUP

A cellular sheaf $\mathcal{F}$ on a graph assigns a stalk to each vertex and edge and a restriction map $F_{v \trianglelefteq e}$ to each incidence; for a 0-cochain x, $(\delta_{\mathcal{F}} x)_e = F_{v \trianglelefteq e} x_v - F_{u \trianglelefteq e} x_u$ and $L_{\mathcal{F}} = \delta_{\mathcal{F}}^{\top} \delta_{\mathcal{F}}$. We study the normalized residual layer

$$h \leftarrow \mathrm{ReLU}\Big(\mathrm{LN}\Big(h - \tau \tilde{L}_{\mathcal{F}} h\Big)\Big), \qquad \tau = \tfrac{1}{2}, \tag{1}$$

with a linear encoder and readout, stalk dimension $d = 4$ and hidden width 64. Our implementation follows the common practice of storing an undirected graph as two directed edges and treats each stored direction $u \to v$ as an edge with its own learned map R_{uv} acting on the source, the target-side map being the identity: $z_{uv} = h_v - R_{uv} h_u$. The operator is row normalized, $\tilde{L}_{\mathcal{F}} h = \tilde{D}^{-1} \delta^{\top} z$, where $\tilde{D}$ counts stored incidences (twice the undirected degree). All claims concern the function realized by Eq. (1), not the restriction-map family in isolation. Published models differ in both normalization and assembly; Section 3 treats those differences explicitly rather than assuming them away.

3 WHEN NOMINAL CAPABILITIES COINCIDE

3.1 NORMALIZATION TURNS CONSTANT TRANSPORT INTO PROPAGATION STRENGTH

Let T be a model with a learned *layerwise propagation strength* τ and identity maps, and C a model with fixed $\tau = \frac{1}{2}$ and a learned *layerwise constant* restriction scalar c. As raw operators they differ: T changes the outer update, C changes the scalar inside the coboundary.

Proposition 1 (Row normalization)**.** *For the layer of Section 2 with $R_{uv} = cI$, $\tilde{L}_c = \frac{1+c^2}{2} I - cD^{-1}A$, hence*

$$h - \tfrac{1}{2}\tilde{L}_c h = a(c)\big[(1 - \tau_{\mathrm{eff}})h + \tau_{\mathrm{eff}} D^{-1} A h\big], \qquad a(c) = \tfrac{3+2c-c^2}{4}, \quad \tau_{\mathrm{eff}}(c) = \tfrac{2c}{3+2c-c^2}.$$

For $0 < c < \sqrt{3}$, $a(c) > 0$ and τ_{eff} spans $(0, 1)$. Since layer normalization removes the positive factor $a(c)$, C realizes every update of T in that range although $\frac{1}{2}\tilde{L}_c \neq \tau \tilde{L}_1$.

Proof. $h - \frac{1}{2}\tilde{L}_c h = \frac{3-c^2}{4} h + \frac{c}{2} D^{-1} A h$; matching coefficients gives $a(1 - \tau_{\mathrm{eff}}) = \frac{3-c^2}{4}$ and $a\tau_{\mathrm{eff}} = \frac{c}{2}$. The roots of $3 + 2c - c^2$ are -1 and 3, so $a > 0$ on $(0, \sqrt{3})$, with $\tau_{\mathrm{eff}} \to 0$ as $c \to 0^+$ and $\tau_{\mathrm{eff}}(\sqrt{3}) = 1$. □

With the implemented $\epsilon = 10^{-5}$ the two layers differ, at each node, by a relative error of at most $\frac{7}{18}\,\epsilon/\sigma_v^2$ in the normalized features, σ_v^2 being the feature variance before normalization (Lemma 1, Appendix D). Mapped through the proposition, fitted learned-strength checkpoints reproduce their logits to 3.7×10^{-4}, and two constant scalars recover 92.5% of the gate gain on Roman-empire and 62.0% on Minesweeper (Appendix A.4). The result is not an artifact of row normalization. Released sheaf code (Bodnar et al., 2022; Purificato et al., 2025) uses an *augmented symmetric* normalization, $\tilde{L} = (D_F + I)^{-1/2}(D_F - A_F)(D_F + I)^{-1/2}$ with $D_F = \mathrm{diag}\big(\sum_{e \ni v} F_{v \trianglelefteq e}^{\top} F_{v \trianglelefteq e}\big)$.

Proposition 2 (Augmented symmetric normalization)**.** *With constant scalar maps c and $s = c^2$, on* any *graph without isolated vertices*

$$\tilde{L}_s \;=\; S^{1/2}\, L_{\mathrm{sym}}\, S^{1/2}, \qquad S = \mathrm{diag}\Big(\frac{s\, d_v}{s\, d_v + 1}\Big), \quad L_{\mathrm{sym}} = I - D^{-1/2} A D^{-1/2}.$$

A constant map is therefore exactly *the ordinary normalized Laplacian conjugated by a node-wise gate $S_{vv} \in (0, 1)$ that depends only on s and the degree: it sets a propagation strength per node and carries no matrix-valued content, with no appeal to layer normalization. On a k-regular graph the gate is constant and $\tilde{L}_s = \frac{sk}{sk+1} L_{\mathrm{sym}}$, a strength $\tau \frac{sk}{sk+1} \in (0, \tau)$; in general $\tilde{L}_s \to L_{\mathrm{sym}}$ as $s \to \infty$ and $\tilde{L}_s \to 0$ as $s \to 0$.*

Proof. $D_F = sD$ and $A_F = sA$, so $\tilde{L}_s = s\,(sD+I)^{-1/2}(D-A)(sD+I)^{-1/2}$. Write $D - A = D^{1/2}L_{\mathrm{sym}}D^{1/2}$; the diagonal factors combine to $\big(sD(sD+I)^{-1}\big)^{1/2} = S^{1/2}$ on each side. □

We checked Proposition 2 against the released Laplacian builder of Bodnar et al. (2022) rather than a reimplementation: the assembled operator matches $s\,(sD+I)^{-1/2}(D-A)(sD+I)^{-1/2}$ to within 10^{-8} on regular and irregular test graphs, and on regular graphs equals $\frac{sk}{sk+1}(I - A/k)$ to within 1.2×10^{-7} (Appendix H).

Both propositions concern a *layerwise constant*. The next result removes that restriction and says, for any family of maps, how much of the stalk a layer can use.

Theorem 1 (Structure algebra)**.** *Let $\mathcal{S} \subset \mathbb{R}^{d\times d}$ contain every restriction map a layer can produce (edge-, direction- and feature-dependent maps allowed) and let $\mathcal{A}$ be the unital algebra generated by $\mathcal{S}$ and its transposes. Under either assembly and either normalization of this paper every $d \times d$ block of the operator lies in $\mathcal{A}$. Hence there are an orthogonal $P \in O(d)$ depending only on $\mathcal{A}$, division algebras $\mathbb{D}_i \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$ and integers n_i, m_i with $\sum_i m_i n_i \dim_{\mathbb{R}} \mathbb{D}_i = d$ such that, up to a permutation of coordinates,*

$$(I_n \otimes P)^{\top}\, \tilde{L}_{\mathcal{F}}\, (I_n \otimes P) = \bigoplus_i L_i \otimes I_{m_i}, \qquad L_i \in M_n\big(M_{n_i}(\mathbb{D}_i)\big).$$

The layer is a direct sum of independent operators. The multiplicities m_i are repeated copies; only the block sizes $n_i \dim_{\mathbb{R}} \mathbb{D}_i$ measure how much of the stalk is used.

The proof (Appendix D) is the Wedderburn decomposition of a transpose-closed matrix algebra, plus the observation that the block normalizer $(D_F + I)^{-1/2}$ is a polynomial in an element of $\mathcal{A}$. The decomposition is classical and computable (Murota et al., 2010; Maehara & Murota, 2010); a representation-wise splitting is known for group-valued gain graphs (Cavaleri et al., 2021) and the flat case for bundle networks (Bamberger et al., 2025). What we add is its reading as the expressible class of a sheaf layer, normalization included, and its use as an axis of attribution.

Corollary 1. (i) Scalar maps $F = cI_d$ *with c signed and edge-dependent: $\mathcal{A} = \mathbb{R}$ and $\tilde{L}_{\mathcal{F}} = M \otimes I_d$, where*

$$\textit{averaged: } M_{vv} = \tfrac{1}{2d_v}\textstyle\sum_{u\sim v}(1 + c_{vu}^2), \qquad M_{vu} = -\tfrac{1}{2d_v}(c_{uv} + c_{vu});$$

$$\textit{product: } M = (D_c + I)^{-\frac{1}{2}} L_c (D_c + I)^{-\frac{1}{2}}, \qquad (L_c)_{vv} = \textstyle\sum_{u\sim v} c_{vu}^2,\ (L_c)_{vu} = -c_{vu}c_{uv}.$$

This is signed edge-weighted message passing; the two directions of an edge enter through one symmetric weight and the stalk plays no role (for positive weights at $d = 1$ this is noted by Bodnar et al. (2022)). (ii) Rotations sharing one angle*, $F = a\,(I_{d/2} \otimes R(\theta))$: $\mathcal{A} \cong \mathbb{C}$, and under $\mathbb{R}^d \cong \mathbb{C}^{d/2}$ the operator is $M_{\mathbb{C}} \otimes I_{d/2}$ with $M_{\mathbb{C}}$ given by (i) after $c \mapsto z = ae^{i\theta}$, $c^2 \mapsto |z|^2$ and conjugation of transposed factors. For the single product $M_{\mathbb{C}}$ is Hermitian, a magnetic Laplacian (Zhang et al., 2021) with learned moduli. That $SO(2)$ connection Laplacians are magnetic Laplacians is classical (Fanuel et al., 2017; Singer & Wu, 2012), and complex-valued sheaf Laplacians generalizing the magnetic one are used for directed graphs (Fiorini et al., 2025; Mule et al., 2026); the point here is the converse, that a common* real *parameterization is one.* (iii) Diagonal maps*: $\mathcal{A} = \mathbb{R}^d$, d independent real edge-weighted layers (as noted by Bodnar et al., 2022).* (iv) Rotated diagonal maps $(I_{d/2} \otimes R(\theta))D$*: $\mathcal{A} = M_2(\mathbb{R})^{d/2}$, $d/2$ independent two-dimensional sheaf layers.* (v) Generic dense maps*: $\mathcal{A} = \mathbb{R}^{d\times d}$, irreducible.*

All five cases were checked against the implemented layers to 10^{-15} in double precision (Appendix H). The ladder of Section 4 thus climbs a chain of algebras, $\mathbb{R} \subset \mathbb{C} \subset M_2(\mathbb{R})^{d/2} \subset \mathbb{R}^{d\times d}$: rungs 1–2 are real edge weighting, rung 3 is complex edge weighting, and only the last rung acts on the stalk irreducibly. Irreducibility over $\mathbb{R}$ does not require the full matrix algebra ($\mathbb{H} \subsetneq M_4(\mathbb{R})$ acts irreducibly on $\mathbb{R}^4$; Appendix F), and the theorem classifies the operator's block structure, not the network's function class (Section 7).

Counting a constant restriction scalar as a capability distinct from propagation strength therefore over-counts, and learned maps do become near-constant (Section 5.5). The normalization also bounds what *any* rung can do.

Proposition 3 (Normalization bounds the propagation operator)**.** *Under the augmented symmetric normalization, for arbitrary restriction maps and either assembly, $P = I - \frac{1}{2}\tilde{L}_{\mathcal{F}}$ is symmetric with*

spectrum in $(0, 1]$*: the layer can only attenuate. Under row normalization this fails already for a constant scalar map* $c > 1$*, whose operator has eigenvalues* $a(c)[(1 - \tau_{\text{eff}}) + \tau_{\text{eff}}\mu] < 0$ *for* μ *near* -1*, and whose self-weight* $(3 - c^2)/4$ *is negative for* $c > \sqrt{3}$*.*

The proof (Appendix D) is $0 \preceq L \preceq 2D_F \prec 2(D_F + I)$, edge by edge.

An assembly pitfall. One further collapse concerns implementations. If each stored direction of an undirected edge is treated as its own edge with its own orthogonal map, the couplings of the pair average to $\frac{1}{2}(g + h^\top)$, whose singular values are $|\cos(\varphi_i/2)| \leq 1$: an attenuation (Proposition 4, Appendix E). Our first implementation was of this kind. The three released sheaf models whose assembly code we inspected form the single product $F_{u \trianglelefteq e}^\top F_{v \trianglelefteq e}$ and are unaffected, but none of the papers with edge-level maps we read (Appendix G) says how the two stored directions are assembled in code. Section 5 therefore reports both assemblies. Appendix F draws three further consequences: below the rotated-diagonal rung, stalk dimension is width (Proposition 5); the operator identifies an edge's maps only up to a gauge (Proposition 6); and a balanced sign field or a holonomy-free angle field is a node-wise conjugation of the previous rung (Proposition 7).

4 An audited attribution protocol

Standard 1: report graph value. Let N_0 be a no-graph predictor sharing encoder, readout, normalization, depth, split and training protocol, and R the risk. With $G_{\text{graph}}(\mathcal{M}_k) = R(N_0) - R(\mathcal{M}_k)$ and $\Gamma_j = R(\mathcal{M}_{j-1}) - R(\mathcal{M}_j)$ we have $G_{\text{graph}}(\mathcal{M}_k) = G_{\text{graph}}(\mathcal{M}_0) + \sum_{j \leq k} \Gamma_j$. Rungs below $k^\dagger = \min\{k : G_{\text{graph}}(\mathcal{M}_k) > 0\}$ *repair* an aggregation rule that is not yet useful; rungs above it *exploit* one that is. Equal increments support different claims in the two regimes.

Standard 2: contain the identity exactly, and warm start. Γ_k can be read as the effect of a capability only if $\mathcal{M}_{k-1} \subseteq \mathcal{M}_k$ as function classes. An identity that is only approached does not qualify: a $\tanh$ incidence multiplier in $(-1, 1)$ cannot attain 1. We use $R = I(1 + s) + \text{diag}(\delta) + \text{offdiag}(O)$ with new heads zero-initialized (a function-preserving expansion, Chen et al., 2016), train each rung from its fitted parent, keep the parent as an eligible validation candidate because warm starts can hurt generalization (Ash & Adams, 2020), and require the child's training loss not to exceed the parent's. Without these safeguards the estimate moves in either direction: a warm start alone changed Γ_1 from $+0.085$ to $+0.042$ on Minesweeper and from $+0.027$ to $+0.039$ on Tolokers (Appendix A).

Standard 3: attribute to cumulative capabilities. For $d \geq 2$ we use the exactly nested chain $\mathcal{M}_0 = \{I\} \subset \mathcal{M}_1 = \{aI : 0 < a < 2\} \subset \mathcal{M}_2 = \{aI\} \subset \mathcal{M}_3 = \{aQ_\theta\} \subset \mathcal{M}_4 = \{Q_\theta D\} \subset \mathcal{M}_5 = \mathbb{R}^{d \times d}$ ($Q_\theta = I_{d/2} \otimes R(\theta)$ a block rotation sharing one angle, D diagonal; a general $Q \in SO(d)$ would already generate $M_d(\mathbb{R})$), adding gating, sign, rotation, anisotropic scaling and unrestricted mixing. Rungs 3–5 are *sheaf-exclusive* in the sense that real edge weighting cannot realize them (Theorem 1); rung 3 alone is commutative, a phase. We report the absolute sheaf-exclusive gain $\Gamma_3 + \Gamma_4 + \Gamma_5$ and its share $\eta_{\text{sheaf}} = (\Gamma_3 + \Gamma_4 + \Gamma_5)/\sum_j \Gamma_j$.

Effective-function audit. Each nominal inclusion is checked by parent–child logit agreement at the embedded parameters, randomized tangent novelty, and a finite perturbation witness. Logit agreement is checked in float32 against a tolerance of 2×10^{-4}; any exceedance is recomputed in float64 and must fall below 10^{-9}, so rounding is distinguished from a genuine nesting failure without relaxing the tolerance (Appendix A).

4.1 SND: a necessity diagnostic that can refuse

For a data-side intervention $I_j(\mathcal{D}) = \mathcal{D}_{-j}$ that removes property j, in the spirit of remove-and-retrain (Hooker et al., 2019) and of perturbation taxonomies of graph benchmarks (Liu et al., 2022), the interaction $\Psi_{j,k} = \Gamma_k(\mathcal{D}) - \Gamma_k(\mathcal{D}_{-j})$ asks whether rung k's value depends on property j. A positive Ψ supports a bounded necessity claim only if four gates pass: (G1) the aggregation rule adds value; (G2) the intervention removes its target without injecting signal or changing nuisance statistics; (G3) the task retains predictive support after intervention, $\text{LCI}_{0.95}[\text{Ret}]$ above a fixed

floor, where Ret is the parent's above-chance performance after intervention relative to before (risk form $1 - R_{-j}(\mathcal{M}_{k-1})/R_{\text{null}}$ with floor 0.05 on the regression battery, score form with floor 0.30 on the classification screens); (G4) the tested rung is itself non-inert, i.e. the interval of Γ_k excludes $(-\epsilon_\Gamma, \epsilon_\Gamma)$ with $\epsilon_\Gamma = 0.01$. G3 is indispensable because a vanishing post-intervention capacity effect is equally consistent with removing the property and with destroying the task; G4 because a difference of two inert increments carries no information. The thresholds are fixed, not tuned per dataset, and a failed gate yields a refusal, not a null. Appendix D shows that no function of $(\Gamma_k(\mathcal{D}), \Gamma_k(\mathcal{D}_{-j}))$ alone can separate necessity from an intervention with side effects, so some gate of this kind is unavoidable.

5 Experiments

Datasets, splits, optimization and the unit of inference are in Appendix A, with full tables; intervals are 95% t intervals over split-level means unless noted. We ask, in turn, what the share of the gain credited to matrix-valued transport depends on (Section 5.1), why two operators disagree about it at depth (Section 5.2), what a data-side intervention adds (Section 5.4), and whether the findings transfer to a published architecture (Section 5.5).

5.1 The share is not an intrinsic quantity

Credit depends on order; the ladder takes the conservative one. In a full 2^3 lattice over learned strength, edge gate and block rotation, with every cell fitted for every unit, rotation earns $+0.159$ on Roman-empire when added first and $+0.023$ when added last: between 94% and 13% of the same total, with a Shapley share of 41%. On Minesweeper the range is 79% to 53% (Table 40). Added first, a rotation is credited with what a scalar can do (Appendix E). The ladder places every scalar capability before any matrix-valued one, so the sheaf-exclusive gain is what remains after edge weighting has had its turn: a last-in credit, which is the minimum over all orders whenever gains are submodular, as they are in this lattice (Table 41). Shares below are conditional on this chain.

The ladder, and its dependence on depth. Table 1 gives the audited ladder at two and four layers (40 units per cell, A100). The two-layer results replicate an earlier run on other hardware, except for an ordering of rotation above gate on Minesweeper, which is not claimed (Appendix A).

Table 1: Audited six-level ladder at two depths, re-run on A100 hardware. Entries are means over n optimizer units (seed $\to$ official split = seed mod 10); Γ_k are adjacent score increments (accuracy, or ROC AUC for Minesweeper and Tolokers). "Sheaf" is $\Gamma_3 + \Gamma_4 + \Gamma_5$ with a 95% t interval over the ten official split means, and η is its share of the identity-to-dense gain. Depth is the number of propagation layers; all other settings are identical between the two depths.

Dataset	L	n	$S(I)$	S(dense)	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Sheaf (95% CI)	η
Minesweeper	2	40	0.7959	0.9049	+0.0396	+0.0116	+0.0376	+0.0181	+0.0020	$+0.0577\ [+0.0485, +0.0670]$	53.0%
Minesweeper	4	40	0.7418	0.9047	+0.1280	+0.0139	+0.0090	+0.0057	+0.0063	$+0.0209\ [+0.0132, +0.0287]$	12.9%
Roman-empire	2	40	0.5812	0.7467	+0.1457	+0.0010	+0.0018	+0.0040	+0.0131	$+0.0188\ [+0.0158, +0.0219]$	11.4%
Roman-empire	4	40	0.3606	0.7779	+0.3473	+0.0009	+0.0199	+0.0233	+0.0259	$+0.0691\ [+0.0657, +0.0725]$	16.6%
Tolokers	2	40	0.7758	0.8346	+0.0397	+0.0055	+0.0086	+0.0038	+0.0012	$+0.0135\ [+0.0101, +0.0170]$	23.1%
Tolokers	4	40	0.7552	0.8171	+0.0322	+0.0097	+0.0086	+0.0060	+0.0054	$+0.0201\ [+0.0147, +0.0254]$	32.4%
Amazon-ratings	2	40	0.4713	0.4812	+0.0073	+0.0010	+0.0012	+0.0002	+0.0000	$+0.0015\ [-0.0002, +0.0032]$	15.7%
Amazon-ratings	4	40	0.4687	0.4779	+0.0060	+0.0019	+0.0001	+0.0014	-0.0003	$+0.0012\ [-0.0001, +0.0026]$	13.5%

Depth changes the allocation, in opposite directions on the two tasks (paired contrasts in Table 17). On Minesweeper the endpoint does not move but the sheaf-exclusive gain falls from $+0.058$ to $+0.021$ and η_{sheaf} from 53% to 13%: identity propagation degrades at depth four and the gate absorbs the difference (under this operator; see below). On Roman-empire the sheaf-exclusive gain *rises* 3.7-fold, from $+0.019$ to $+0.069$, yet η_{sheaf} moves only from 11% to 17%, because the gate increment in its denominator is inflated by an identity baseline that collapses to 0.361: against a depth-matched no-graph model, graph value at the identity rung falls from -0.078 to -0.295, so about 85% of the four-layer gate increment is repair (Standard 1) in this row-normalized layer. η_{sheaf} alone can thus hide a change that the absolute gain shows; we report both. The audited

Table 2: Assembly × normalization on Minesweeper: sheaf-exclusive gain at two and four layers, its split-paired depth difference, and the gate increment Γ_1. Ten official splits per cell, the same seeds in every cell (laptop GPU); †: interval excludes zero. At four layers the four cells start and end within 0.012 of each other (identity 0.742–0.745, dense 0.900–0.912).

Assembly	Normalization	Gain, $L{=}2$	Gain, $L{=}4$	$L{=}4$ minus $L{=}2$	Γ_1, $L{=}2$	Γ_1, $L{=}4$
averaged	row	$+0.051\,[+0.033, +0.069]$	$+0.017\,[+0.011, +0.023]$	$-0.034\,[-0.055, -0.013]^{\dagger}$	$+0.047$	$+0.126$
averaged	aug.-sym.	$+0.067\,[+0.053, +0.081]$	$+0.077\,[+0.057, +0.097]$	$+0.010\,[-0.009, +0.030]$	$+0.035$	$+0.060$
single product	row	$+0.056\,[+0.038, +0.074]$	$+0.030\,[+0.020, +0.039]$	$-0.026\,[-0.046, -0.006]^{\dagger}$	$+0.047$	$+0.118$
single product	aug.-sym.	$+0.058\,[+0.047, +0.070]$	$+0.093\,[+0.080, +0.106]$	$+0.035\,[+0.024, +0.045]^{\dagger}$	$+0.038$	$+0.067$

ladder costs accuracy (0.747 vs. 0.793 under free training on Roman-empire; 0.778 and 0.905 at four layers against 0.804 and 0.922 for Bodnar et al. (2022)). Split by generated algebra (Table 19), most of the sheaf-exclusive gain on Minesweeper at two layers is commutative ($+0.038$ of $+0.058$: a phase suffices) and, under the released operator, gauge-trivial: replacing the fitted angle field by its nearest holonomy-free field costs 0.009 of a 0.024 rotation increment, removing it costs 0.042 (Appendix F). On Roman-empire the gain is almost entirely non-commutative and grows with depth ($+0.017 \rightarrow +0.049$); the irreducible four-dimensional rung never carries more than 9% of the total gain, and the non-commutative rungs together at most 20%. A parent-continuation control, which trains the fitted parent again under the child's own budget, leaves the sheaf-exclusive gain at $+0.045\,[+0.026, +0.064]$ of $+0.063$ on Minesweeper at two layers and at $+0.015\,[+0.012, +0.019]$ of $+0.016$ on Roman-empire, and the depth reversal of Section 5.2 survives the control (Appendix A.11). Non-commutativity can be bought inside a division algebra too: a quaternion rung ($\mathbb{H}$), warm-started from rotation, adds a resolved $+0.008$ and $+0.006$ at two layers and $+0.017$ and $+0.024$ at four (Appendix F).

Are the null increments optimization failures? Mostly not. When a parent is retained (increment exactly zero), the fit logs tell whether the child lowered training cross-entropy by at least 1% without validating better (*fit-only*) or never descended (*no-descent*). On Amazon-ratings, whose sheaf-exclusive gain is practically zero, every retained rung is fit-only. No-descent occurs at four layers on Minesweeper (19% of rung fits) and Tolokers (30%); dropping the affected units leaves the depth contrasts in place (Table 18).

5.2 The operator decides the sign of a depth effect

Does the attribution survive the released operator? We re-ran the ladder with the single-product assembly and augmented symmetric normalization of released code, changing nothing else (Tables 24 and 20). *At two layers* the sheaf-exclusive gain does not move (paired difference $+0.000\,[-0.006, 0.006]$ on Roman-empire; $+0.058$ against $+0.051$ on Minesweeper, not resolved), while everything scalar does: identity propagation on Roman-empire scores 0.712 instead of 0.578, its graph value turns from -0.081 to $+0.054$, and η_{sheaf} rises from 11% to 35% with its numerator fixed. *At four layers* the gain rises with depth under both operators on Roman-empire, where the released operator now has a negative graph value too (-0.074), but on Minesweeper the direction reverses: -0.034 under ours, $+0.035$ under the released one, from the same start and end points.

Normalization decides the sign. Table 2 crosses the two factors on Minesweeper. At two layers neither is resolved. At four layers the normalization moves the sheaf-exclusive gain by $+0.060\,[0.038, 0.082]$ and $+0.063\,[0.048, 0.078]$ under the two assemblies, the assembly moves it by $+0.013$ and $+0.016$, and their interaction is $+0.003\,[-0.016, 0.022]$: the effects add, and normalization accounts for about four fifths of the gap. The mechanism is at the gate, which collects about $+0.12$ of a common total under row normalization and half of that under the augmented symmetric one (-0.066, -0.052). Proposition 3 predicts where the difference comes from and the fitted gates confirm it: at four layers every row-normalized gate leaves $(0, 1]$ and switches layers off, while every released-normalization operator stays inside it (Table 39). The smaller assembly effect has the sign of Proposition 4. With row normalization the gain falls with depth under either assembly (-0.034, -0.026); with the augmented symmetric one it does not ($+0.010$, not resolved, and $+0.035$). On two further datasets (Appendix A) the effect of the normalization on the depth difference itself is $+0.044\,[0.007, 0.081]$ and $+0.061\,[0.035, 0.086]$ here, $-0.019\,[-0.028, -0.010]$ on Tolokers, of the

Table 3: What the scalar control is allowed to do. G is the sheaf-exclusive gain $S(\mathcal{M}_5) - S(\mathcal{M}_2)$ under the fixed update and under a learned degree-3 response, D their seed-paired difference (the preregistered quantity) and D^{cont} the same with both arms measured against their own continued parent. The last columns separate the two things a higher degree changes at once: free coefficients at the restricted arm's one-hop reach, and three hops with θ frozen at the expansion of $(I - \frac{1}{2}L)^3$, which widens a layer without freeing a coefficient. Both controls were run after the preregistered comparison, not as part of it. Ten seeds per cell, seed $\rightarrow$ official split; †: interval excludes zero. Appendix A.10 gives the degree sweep, the reach controls, the endpoints and a third dataset.

Dataset	L	G fixed	G free	D	D^{cont}	1 hop, free coeff.	3 hops, fixed coeff.
Minesweeper	2	$+0.058\,[+0.047, +0.070]^\dagger$	$+0.004\,[+0.000, +0.007]^\dagger$	$-0.055\,[-0.067, -0.043]^\dagger$	$-0.036\,[-0.044, -0.027]^\dagger$	$-0.054\,[-0.065, -0.043]^\dagger$	$-0.042\,[-0.055, -0.030]^\dagger$
Minesweeper	4	$+0.093\,[+0.080, +0.106]^\dagger$	$+0.008\,[+0.003, +0.012]^\dagger$	$-0.085\,[-0.099, -0.072]^\dagger$	$-0.078\,[-0.092, -0.064]^\dagger$	$-0.091\,[-0.103, -0.079]^\dagger$	$-0.066\,[-0.079, -0.053]^\dagger$
Roman-empire	2	$+0.018\,[+0.013, +0.023]^\dagger$	$+0.024\,[+0.019, +0.028]^\dagger$	$+0.006\,[-0.002, +0.013]$	$-0.001\,[-0.005, +0.002]$	$-0.004\,[-0.010, +0.003]$	$+0.003\,[-0.002, +0.009]$
Roman-empire	4	$+0.058\,[+0.052, +0.064]^\dagger$	$+0.074\,[+0.063, +0.086]^\dagger$	$+0.017\,[+0.004, +0.029]^\dagger$	$+0.008\,[-0.010, +0.027]$	$+0.004\,[-0.002, +0.011]$	$+0.006\,[-0.004, +0.015]$

opposite sign, and unresolved on Amazon-ratings, where there is no gain to allocate: that the depth behaviour belongs to the operator replicates, which normalization is favoured does not.

5.3 What the scalar control is allowed to do decides the share

The scalar rungs are constrained in a second way that has nothing to do with transport: every layer propagates with the one fixed polynomial $h \mapsto h - \frac{1}{2}Lh$, so whatever a scalar response could have done with other coefficients is credited to matrix-valued transport. In a preregistered experiment (Appendix A.10) the free arm instead learns the coefficients of $h \mapsto \sum_{k \le K} \theta_k L^k h$ per layer, θ shared by all channels and initialized at the fixed update, leaving the rungs, their nesting, the warm starts and the seeds unchanged. The preregistered quantity is the seed-paired interaction $D = G_{\mathrm{free}} - G_{\mathrm{restricted}}$ of the sheaf-exclusive gain.

The answer (Table 3) is a substitution on one task and no resolved interaction on the other. On Minesweeper 92% of the gain credited to matrix-valued transport is available to the scalar rungs once their spectral response is free, 94% against the continued parents (shares are point estimates). On Roman-empire the uncontrolled point estimate runs the other way, and nothing survives that control. A higher degree changes both the reach of a layer and its response; controls run after the preregistration separate them. Freeing only the coefficients at one hop produces a reduction of comparable magnitude (98% at four layers, surviving the continuation control), so the substitution does not need a wider receptive field. A fixed three-hop filter also reduces the gain attributed to matrix-valued transport by 71% on four-layer Minesweeper, without learning polynomial coefficients. This reduction is smaller than that achieved by learned one-hop filtering, with a paired contrast of $+0.025$ $[0.020,\ 0.031]$. On Roman-empire that arm, like every other, resolves nothing.

5.4 The data side: a budget that depth and operator reallocate

Calibration. A four-cell unit test with known answers classifies all four correctly, and the raw interaction there does not separate genuine necessity ($+0.320$) from an invalid intervention that destroys the task ($+0.323$); only the retention gate does (Table 12, Appendix H). On six real datasets, permuting node features within label strata (an oracle intervention applied across splits, whose side effects on nuisance statistics are audited in Appendix H), Roman-empire's large gate increment does not detectably change when 91% of its feature autocorrelation is removed ($\Psi = -0.0001\,[-0.005, 0.005]$), Tolokers shows a positive interaction ($+0.023\,[0.012, 0.035]$), but its nuisance audit fails G2, because the permutation also removes most of the degree–feature dependence there (Appendix H); the screen therefore refuses the property-specific reading and reports dependence on the intervention. Three further datasets are refused by G4 and one is unresolved (Table 13).

The effect of an intervention is a budget, and depth reallocates it. The damage an intervention does to the dense model telescopes exactly over any chain of models, so with $\Psi_{j,k} = \Gamma_k(\mathcal{D}) - \Gamma_k(\mathcal{D}_{-j})$ the drop $S(\mathcal{M}_5; \mathcal{D}) - S(\mathcal{M}_5; \mathcal{D}_{-j})$ equals $\Psi_{j,0} + \sum_{k \ge 1} \Psi_{j,k}$ (Appendix A), and along the nested ladder each term is the change of an audited increment: so the $\Psi_{j,k}$ are shares of one measured total. Repeating the ladder under intervention at both depths (Appendix A), the total does not move detectably with depth (0.099 and 0.094 on Minesweeper; 0.020 and 0.025 on Roman-empire), while its allocation does. On Minesweeper the interaction leaves rotation ($+0.030 \rightarrow +0.003$) for the

gate ($-0.009 \to +0.047$, same sign on all ten splits): the same feature-dependent signal is collected by matrix-valued transport in a shallow model and by a scalar gate once depth supplies a multi-hop path. On Roman-empire the property-inert scalar rungs carry the largest increments, while the sheaf-exclusive part rises from $+0.007$ to $+0.024$: a large increment and a property-specific one are different things.

What generalizes, and what does not. Under the released operator the total is again unchanged by depth $(0.101, 0.104)$ and the identity rung again gives up half of its part (-0.029 against -0.030), but the sheaf-exclusive rungs collect it ($+0.020 \to +0.039$, difference $+0.019\,[0.003, 0.034]$) rather than the scalar ones, a depth change the operator moves by $+0.053\,[0.020, 0.085]$ (Table 25). Degree-preserving rewiring of 22% of Minesweeper's edges reproduces the invariant total and the drift toward the gate but not the two-layer concentration at rotation (Table 26), and doubling the width moves both share and gain ($46\% \to 73\%$, $+0.051 \to +0.070$; Table 27): the invariant total holds across operators and interventions, the collecting rung does not.

5.5 Transfer to a published architecture

We audited the released code of a directed sheaf hypergraph network (Mule et al., 2026) under the authors' best configurations, one or two flags per comparison (Appendix B). **Sign reversal.** At stalk dimension one a single flag switches between trivial and learned maps at equal parameter count: learned maps score 50.5 ± 25.2 against 80.1 ± 2.7 (-29.6 points), but the learned arm is bimodal, six of ten runs collapsing to about 30% and the rest averaging 81.8, so the sign depends on convergence. **Saturation.** An identity-containing parameterization removes the collapse, and the best learned arm then exceeds the trivial control by $+0.85$ points ($[-0.71, 2.41]$): the survivor-conditional advantage was a selection effect (Table 33). **Family labels.** With only the family token changed, the increment to the common general endpoint is -1.2 points from the diagonal branch and $+6.7$ from the orthogonal one on Telegram (Table 35): which incomparable branch is called the baseline decides the sign.

6 Related work

Sheaf Laplacians as a learning primitive (Hansen & Ghrist, 2019; Hansen & Gebhart, 2020) led to neural sheaf diffusion (Bodnar et al., 2022) and to variants that compare restriction-map families as a design choice; a second line of work intervenes on the model, asking whether learnable Laplacians are needed, whether holonomy is used, why depth loses signal, how to filter it spectrally, and what labels leave unidentified (Hernandez Caralt et al., 2026; Grover & Bourgerie, 2026; Bourgerie et al., 2026; Borgi et al., 2025; Liu, 2026). We ask instead what a comparison of families identifies, which Theorem 1 answers at the level of the operator. Appendix G places the work in this literature and audits what ten of these papers report.

7 Discussion

What to report. Since the share is not intrinsic, a claim that matrix-valued transport helps should state: (1) a depth-matched no-graph baseline and the side of $k^\dagger$ each increment lies on; (2) exact containment, warm starts, a continued-parent control and the collapse rate; (3) increments along a nested chain labelled by generated algebra, absolute and as a share; (4) the normalization, the edge assembly and the spectral response the scalar control is allowed; (5) for a data-dependence claim, the retention and inertness gates and the refusals.

Limitations. Shares are conditional on one exact chain. The depth study uses one row-normalized layer and the released operator, ten seeds per cell, two datasets for the ladder (a third partially) and one for the intervention budget, below the published accuracy; "does not move detectably" means intervals of width ± 0.01, not an equivalence test; the substitution test uses one family of filters, evaluates reach alone with one fixed filter, and has a single run on its third dataset; Theorem 1 and Proposition 3 concern the operator, not the network; and the published-architecture audits cover one model family.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing author-designed experiments, writing, running, maintaining, and debugging experimental and analysis code, drafting selected passages, restructuring sections, polishing the writing, and organizing related work. The research questions, experimental designs, and mathematical derivations were developed by the authors, who made all final scientific decisions. Generative AI was not used to generate datasets. All reported results came from executed runs whose logs were retained. We reviewed all AI-assisted code and text and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Appendix A specifies datasets, splits, architecture, optimizer, selection rule and the unit of inference, and states the hardware used for each table. Proofs of all propositions are in the main text; Appendix H describes the numerical checks against released code and the known-truth generator. For the published architectures we used the authors' released code and configurations; we list every deviation, which is a single changed flag per comparison plus one compatibility fix to a data loader for a newer library version (Appendix H). Code and run records will be released.

REFERENCES

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural network for message diffusion on graphs. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=scI9307PLG.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection laplacians. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 28–36. PMLR, 2022a.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, and Pietro Liò. Sheaf attention networks. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022b. URL https://openreview.net/forum?id=LIDvgVjpkZr.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. In *Advances in Neural Information Processing Systems*, volume 35, pp. 18527–18541, 2022.

Alessio Borgi, Fabrizio Silvestri, and Pietro Liò. Polynomial neural sheaf diffusion: A spectral filtering approach on cellular sheaves. *arXiv preprint arXiv:2512.00242*, 2025. URL https://arxiv.org/abs/2512.00242.

Rémi Bourgerie, Šarūnas Girdzijauskas, and Viktoria Fodor. Deep neural sheaf diffusion. *arXiv preprint arXiv:2605.19021*, 2026. URL https://arxiv.org/abs/2605.19021.

Luke Braithwaite, Iulia Duta, and Pietro Liò. Heterogeneous sheaf neural networks. *arXiv preprint arXiv:2409.08036*, 2024. URL https://arxiv.org/abs/2409.08036.

Matteo Cavaleri, Daniele D'Angeli, and Alfredo Donno. A group representation approach to balance of gain graphs. *Journal of Algebraic Combinatorics*, 54(1):265–293, 2021. doi: 10.1007/s10801-020-00977-w.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735. PMLR, 2020. URL https://proceedings.mlr.press/v119/chen20v.html.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2Net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.05641.

Iulia Duta, Giulia Cassara, Fabrizio Silvestri, and Pietro Lio. Sheaf hypergraph networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12087–12099, 2023.

Michaël Fanuel, Carlos M. Alaíz, and Johan A. K. Suykens. Magnetic eigenmaps for community detection in directed networks. *Physical Review E*, 95(2):022302, 2017. doi: 10.1103/PhysRevE.95.022302.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 101–109, 2019. doi: 10.1145/3298689.3347058. URL https://arxiv.org/abs/1907.06902.

Stefano Fiorini, Hakan Aktas, Iulia Duta, Stefano Coniglio, Pietro Morerio, Alessio Del Bue, and Pietro Lio. Sheaves reloaded: A directional awakening. *arXiv preprint arXiv:2506.02842*, 2025.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Ankit Grover and Rémi Bourgerie. Do sheaf neural networks use holonomy? a measure–intervene–control study. *arXiv preprint arXiv:2607.19514*, 2026. URL https://arxiv.org/abs/2607.19514.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks. *arXiv preprint arXiv:2012.06333*, 2020.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Ferran Hernandez Caralt, Guillermo Bernárdez Gil, Iulia Duta, Pietro Liò, and Eduard Alarcón Cot. Joint diffusion processes as an inductive bias in sheaf neural networks. In *Geometry-Grounded Representation Learning and Generative Modeling*, 2024. URL https://arxiv.org/abs/2407.20597.

Ferran Hernandez Caralt, Mar Gonzàlez i Català, Adrián Bazaga, and Pietro Liò. On the necessity of learnable sheaf laplacians. *arXiv preprint arXiv:2603.05395*, 2026. URL https://arxiv.org/abs/2603.05395.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Tsit-Yuen Lam. *A First Course in Noncommutative Rings*, volume 131 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2001. doi: 10.1007/978-1-4419-8616-0.

Renming Liu, Semih Cantürk, Frederik Wenkel, Sarah McGuire, Xinyi Wang, Anna Little, Leslie O'Bray, Michael Perlmutter, Bastian Rieck, Matthew Hirn, Guy Wolf, and Ladislav Rampášek. Taxonomy of benchmarks in graph representation learning. In *Proceedings of the First Learning on Graphs Conference*, 2022. URL https://arxiv.org/abs/2206.07729.

Yi Liu. Learned, relied upon, or necessary? separating checkpoint dependence from task-level value in sheaf GNNs. *arXiv preprint arXiv:2607.25387*, 2026. URL https://arxiv.org/abs/2607.25387.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/092359ce5cf60a80e882378944bf1be4-Abstract-Conference.html.

Takanori Maehara and Kazuo Murota. A numerical algorithm for block-diagonal decomposition of matrix *-algebras with general irreducible components. *Japan Journal of Industrial and Applied Mathematics*, 27(2):263–293, 2010. doi: 10.1007/s13160-010-0007-8.

Emanuele Mule, Stefano Fiorini, Antonio Purificato, Federico Siciliano, Stefano Coniglio, and Fabrizio Silvestri. Directional sheaf hypergraph networks: Unifying learning on directed and undirected hypergraphs. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=d1hQkcDy7c.

Kazuo Murota, Yoshihiro Kanno, Masakazu Kojima, and Sadayoshi Kojima. A numerical algorithm for block-diagonal decomposition of matrix *-algebras with application to semidefinite programming. *Japan Journal of Industrial and Applied Mathematics*, 27(1):125–160, 2010. doi: 10.1007/s13160-010-0006-9.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1e2agrFvS.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

Antonio Purificato, Giulia Cassarà, Federico Siciliano, Pietro Liò, and Fabrizio Silvestri. Sheaf neural networks for graph-based recommender systems. *ACM Transactions on Recommender Systems*, 4(2), 2025. doi: 10.1145/3742898.

Amit Singer and Hau-Tieng Wu. Vector diffusion maps and the connection Laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012. doi: 10.1002/cpa.21395.

Olga Zaghen. Nonlinear sheaf diffusion in graph neural networks. *arXiv preprint arXiv:2403.00337*, 2024. URL https://arxiv.org/abs/2403.00337.

Thomas Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4(1):47–74, 1982.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A neural network for directed graphs. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27003–27015, 2021.

A Full experimental results

A.1 Setup

All node-classification experiments use the public splits shipped with each dataset: the ten official splits of Roman-empire, Minesweeper, Tolokers, Amazon-ratings and Questions (Platonov et al., 2023), the ten standard $48/32/20$ splits of the WebKB and Actor graphs (Pei et al., 2020), and, for Cora, independently generated random $60/20/20$ splits. The metric is accuracy, except ROC AUC for the binary tasks (Minesweeper, Tolokers, Questions). The model is the layer of Eq. (1) with a linear encoder, hidden width 64, stalk dimension 4, dropout 0.3, and a linear readout; restriction maps are predicted per stored directed edge by a two-layer MLP on the concatenated endpoint features, with every newly activated head zero-initialized. Optimization uses Adam (learning rate 0.01, weight decay 10^{-5}), at most 300 epochs, validation every five epochs and patience 40 evaluations. Checkpoints are selected on validation only and each fitted model is evaluated on the test split exactly once. A *unit* is one (split, optimizer seed) pair; seeds map to splits by seed mod 10. Unless stated otherwise, intervals are 95% t intervals over split-level means, so that optimizer repeats within a split are averaged before inference rather than treated as independent. Experiments ran on a laptop GPU (RTX 4060, PyTorch 2.6) and, for the depth study and all published-model audits, on A100 GPUs (PyTorch 2.0.1); each table states which.

A.2 Graph value comes first, and it is not a property of the data alone

Table 4 instantiates Standard 1 on six datasets. Five have $k^\dagger = 0$; Roman-empire alone has $k^\dagger = 1$, with $0.06510/0.14293 = 45.6\%$ of its first increment *repairing* harmful identity propagation rather than exploiting useful aggregation. Cora and Questions have inert first rungs, and Amazon-ratings sits inside the pre-specified equivalence margin. Reporting a Γ_k without saying which side of $k^\dagger$ it occupies conflates two different claims.

Table 4: Strictly nested graph-value decomposition for the $p = 2$ Laplacian experiment. Risks are one minus accuracy for Cora, Roman-empire, and Amazon-ratings and one minus ROC AUC for the other datasets. Intervals are paired 95% t intervals over official splits, except Cora, where 40 independently generated splits are the units. C16 uses the pre-specified internal-log equivalence margin $\epsilon_\Gamma = 0.01$.

Dataset	$G_{\mathrm{graph}}(\mathcal{M}_0)$	Γ_1 (95% CI)	$k^\dagger$	C16 / reading
Roman-empire	-0.06510	$+0.14293\ [0.14038, 0.14547]$	1	pass; 45.6% repair
Minesweeper	$+0.28744$	$+0.04181\ [0.03150, 0.05213]$	0	pass; utilization
Tolokers	$+0.04067$	$+0.03902\ [0.03315, 0.04490]$	0	pass; utilization
Amazon-ratings	$+0.04398$	$+0.00681\ [0.00495, 0.00867]$	0	fail; sub-margin
Cora	$+0.13095$	$-0.00115\ [-0.00299, 0.00068]$	0	fail; inert
Questions	$+0.02020$	$-0.00001\ [-0.00004, 0.00001]$	0	fail; inert, C11-marginal

A.3 Optimization protocol is part of the estimand

Exact function-class inclusion is not sufficient if successive rungs are optimized independently. Replacing random initialization of the gate rung by a warm start from the fitted identity rung changed Γ_1 *in both directions*: Questions from -0.01868 to -0.00001, eliminating a spurious negative decision; Minesweeper from $+0.08510$ to $+0.04181$; Tolokers from $+0.02652$ to $+0.03902$. Signed shifts spanned $[-0.04329, +0.01867]$, and the largest discrepancy occurred for a clearly positive rather than a borderline estimate. Non-nested optimization therefore has no empirically stable bias direction.

A 2^3 factorial over initialization, parent retention, and a parent training-loss cap (Table 5) shows that no single safeguard explains the bias. On Questions and Cora, retaining the parent as an eligible validation candidate was the dominant correction; on Roman-empire and Minesweeper every gate trajectory supplied an eligible checkpoint, so retention and filtering were inactive and initialization alone moved the estimate, in opposite directions on the two datasets. A continued-identity arm shows Minesweeper gains $+0.01662$ from the extra optimization budget alone; after subtracting it the safeguarded gate margin is still $+0.02628\ [0.01878, 0.03378]$.

Table 5: Main effects in the 2^3 optimization-protocol factorial. Each entry is the change in measured gate effect with a paired 95% t interval over independent splits. The ten official splits are the units after averaging two optimizer replicates; Cora uses 20 independently generated splits. "Continue I" is the score change from continuing optimization of the fitted identity model.

Dataset	warm − random	retain − drop	CE-filter − unconstrained	continue I
Roman-empire	$+0.01085\ [0.00722, 0.01447]$	0	0	$+0.00068\ [-0.00122, 0.00258]$
Minesweeper	$-0.04167\ [-0.04551, -0.03782]$	$\lvert\Delta\rvert < 10^{-6}$	$\lvert\Delta\rvert < 10^{-6}$	$+0.01662\ [0.01096, 0.02228]$
Questions	$+0.00821\ [0.00465, 0.01177]$	$+0.01438\ [0.01178, 0.01698]$	$-0.00097\ [-0.00181, -0.00013]$	$-0.00033\ [-0.00131, 0.00065]$
Cora	$-0.00083\ [-0.00304, 0.00138]$	$+0.00720\ [0.00395, 0.01044]$	$-0.00217\ [-0.00363, -0.00071]$	$-0.00074\ [-0.00180, 0.00033]$

These separate three obligations that are often conflated: initialize from the parent to control the optimization path, retain the parent so that "no upgrade" is selectable, and cap training risk to audit empirical nesting. None of them guarantees test-risk monotonicity, which is a generalization question.

A.4 Proposition 1, verified

We first separate global propagation strength from edge differentiation in an exactly nested 2×2 design: I fixed identity propagation, T adding two learned layerwise strengths, G adding a positive edge gate at fixed strength, TG both. The edge-differentiation margin after controlling strength is $\Delta_{\mathrm{edge}} = S(TG) - S(T)$.

Learned layerwise strength alone reproduces $0.1273/0.1441 = 88.4\%$ of the fixed-strength gate gain on Roman-empire, with fitted strengths moving from 0.5 to means of 0.292 and 0.265 — identifying attenuation directly as the dominant capability — and 55.2% on Minesweeper. A positive edge-specific margin nevertheless survives on both.

Table 6: Propagation-strength × edge-differentiation factorial. Entries are mean accuracy, except Minesweeper, which uses ROC AUC. Intervals for Δ_{edge} are paired 95% t intervals over independent splits. This is an independently trained batch from Table 4; its $G - I = 0.14410$ and that table's $\Gamma_1 = 0.14293$ estimate the same contrast under different random trajectories and are not expected to be numerically identical.

Dataset	$S(I)$	$S(T)$	$S(G)$	$S(TG)$	Δ_{edge} (95% CI)
Roman-empire	0.5809	0.7082	0.7250	**0.7263**	$+0.0181\ [0.0154, 0.0208]$
Minesweeper	0.7962	0.8207	0.8406	**0.8453**	$+0.0246\ [0.0160, 0.0331]$
Cora	**0.8817**	0.8810	0.8781	0.8774	$-0.0036\ [-0.0074, 0.0002]$

We then test the normalization-induced overlap of Proposition 1 directly. Let C replace the two learned strengths by two learned layerwise constant restriction scalars at fixed strength; T and C have the same number of active scalar parameters. Each fitted T checkpoint was mapped analytically into C through the proposition. The immediate $T \to C$ map changed training cross-entropy by approximately zero, with mean absolute logit discrepancy 3.69×10^{-4} (Roman-empire), 2.74×10^{-5} (Minesweeper) and 5.96×10^{-8} (Cora) — consistent with the finite layer-normalization constant $\epsilon = 10^{-5}$ and float aggregation, which is what makes the equivalence approximate rather than exact.

Table 7: Function-space audit of propagation strength (T), layerwise constant transport (C), and edge-conditioned transport (G). Entries are mean accuracy, except Minesweeper, which uses ROC AUC. Intervals are paired 95% t intervals over independent splits. "C share" is the descriptive ratio $[S(C) - S(I)]/[S(G) - S(I)]$ and is not a causal mediated fraction.

Dataset	$S(I)$	$S(T)$	$S(C)$	$S(G)$	$G - C$ (95% CI)	C share
Roman-empire	0.5814	0.7084	0.7120	**0.7226**	$+0.0106\ [0.0085, 0.0127]$	92.5%
Minesweeper	0.7957	0.8205	0.8236	**0.8407**	$+0.0171\ [0.0096, 0.0246]$	62.0%
Cora	**0.8845**	0.8847	0.8846	0.8833	$-0.0013\ [-0.0040, 0.0014]$	–

On Roman-empire C recovers 92.5% of the observed $G - I$ gain, leaving a positive edge-specific remainder of $+0.0106$; on Minesweeper the recovery is 62.0% with remainder $+0.0171$. Cora resolves neither component. Final C scores exceed T by $+0.00352\ [0.00132, 0.00572]$ and $+0.00305\ [0.00044, 0.00567]$, but these small differences do not establish a new capability: finite-ϵ normalization, parameterization and optimization all remain different. The robust conclusion is the one Proposition 1 predicts — most of the Roman-empire gate gain, and a majority of the Minesweeper gain, is available to a two-parameter constant-transport family.

A.5 Proposition 4, verified

We tested the unsigned sublattice $\{I\} \subset \{aI, Q\} \subset \{aQ\}$, where aI and Q are incomparable branches and aQ is their join, with $J = [S(aQ) - S(aI)] - [S(Q) - S(I)]$.

Table 8: Strictly nested gate–rotation factorial on three real datasets. Entries for I, Q, aI, and aQ are mean accuracy, except Minesweeper, which uses ROC AUC. Interaction intervals are paired 95% t intervals over independent splits. Negative J indicates sub-additivity on the reported score scale.

Dataset	$S(I)$	$S(Q)$	$S(aI)$	$S(aQ)$	J (95% CI)
Roman-empire	0.5784	0.7397	0.7261	**0.7486**	$-0.1388\ [-0.1417, -0.1359]$
Minesweeper	0.7947	0.8838	0.8335	**0.9042**	$-0.0184\ [-0.0298, -0.0069]$
Cora	0.8829	0.8800	0.8833	0.8809	$+0.0005\ [-0.0004, +0.0013]$

Gate and rotation gains are strongly non-additive. On Roman-empire rotation alone gains 0.1613 and gating alone 0.1477, yet the joint model improves on the better parent by only 0.0225. Cora resolves neither, and its fits frequently return to an embedded parent checkpoint (32 of 40 Q fits, 36 of 40 aI, 36 of 40 joint), a negative control showing the protocol does not force extra capacity to appear useful.

Proposition 4 explains why. We split the block rotation into its symmetric and antisymmetric angular components, A retaining only ψ and R only ϕ, and tested the strict lattice I, A, R, AR (Fig. 2). On Roman-empire attenuation alone gains $+0.1467\,[0.1435, 0.1499]$ and reciprocal rotation alone $+0.1553\,[0.1498, 0.1608]$, but rotation after attenuation retains only $+0.0252\,[0.0211, 0.0293]$. Minesweeper differs mechanistically: rotation after attenuation remains $+0.0620\,[0.0561, 0.0678]$ while attenuation after rotation is only $+0.0053\,[0.0018, 0.0087]$. So the symmetric component is not a formality — on one dataset it carries almost the whole apparent rotation effect.

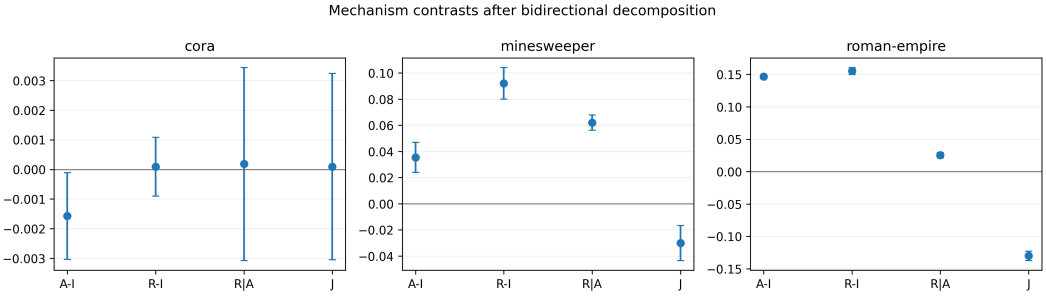


Figure 2: Bidirectional decomposition of block rotation into the implicit attenuation A predicted by Proposition 4 and reciprocal rotation R. Points are paired mean score contrasts, bars are 95% t intervals over independent splits.

Finally we crossed layerwise strength, scalar gate and rotation in the complete eight-cell lattice. Rotation survives both simpler controls: $Q \mid TG = +0.0226\,[0.0194, 0.0258]$ on Roman-empire and $+0.0568\,[0.0406, 0.0731]$ on Minesweeper (Fig. 3). The negative gate–rotation interaction is therefore not exhausted by a shared ability to correct global propagation strength.

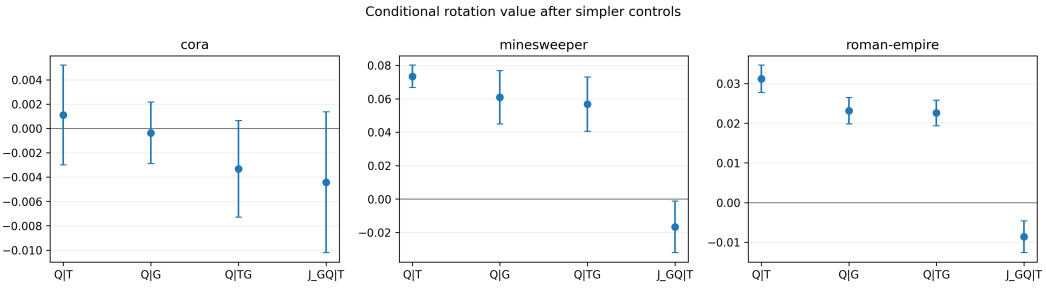


Figure 3: Conditional block-rotation value in the strict strength–gate–rotation lattice. Bars are 95% split-level t intervals.

A.6 The audited capability ladder

We completed the six-level ladder on Roman-empire and Minesweeper using 40 optimizer units per dataset. Every rung was warm-started from its fitted parent, retained the parent as an eligible validation candidate, and was constrained by the parent training-loss cap. Largest parent-embedding discrepancies were 7.06×10^{-5} and 9.92×10^{-5}, within the declared 10^{-4} tolerance; all transitions had positive finite-witness changes, so no rung was collapsed.

The two tasks allocate credit very differently. On Roman-empire 88.8% of the endpoint gain lies in the gate-plus-sign segment and only 11.2% in the sheaf-exclusive segment; its fingerprint is concentrated, $\max_k \pi_k = 0.883$ and $\exp[H(\pi)] = 1.63$. On Minesweeper rotation is the largest single increment, $\Gamma_3 = +0.04317\,[0.03414, 0.05220]$, exceeding gate at $+0.03720$; the matrix-valued segment contributes 58.5% and the fingerprint spreads over an effective 3.76 rungs. This is an existence result, not a trend estimate: the same audited ladder assigns fivefold different sheaf-exclusive shares on two tasks, and on one real dataset a capability unavailable to scalar propagation is the dominant rung *in this run*. Section A.9 reports an independent re-run in which the sheaf-exclusive magnitude replicates but the ordering of rotation over gate does not.

Table 9: Audited six-level capacity ladder. Entries are adjacent paired score increments Γ_k with 95% t intervals over ten official split means. Roman-empire uses accuracy and Minesweeper ROC AUC. Bold entries have lower interval endpoints above the pre-specified internal-log practical margin $\epsilon_\Gamma = 0.01$.

Increment	Roman-empire	Minesweeper
Γ_1: identity → gate	$+\mathbf{0.14448}$ $[\mathbf{0.14078}, \mathbf{0.14817}]$	$+\mathbf{0.03720}$ $[\mathbf{0.03139}, \mathbf{0.04302}]$
Γ_2: gate → sign	$+0.00093$ $[-0.00029, 0.00215]$	$+0.00794$ $[0.00467, 0.01122]$
Γ_3: sign → rotation	$+0.00273$ $[0.00180, 0.00365]$	$+\mathbf{0.04317}$ $[\mathbf{0.03414}, \mathbf{0.05220}]$
Γ_4: rotation → diagonal	$+0.00409$ $[0.00291, 0.00527]$	$+\mathbf{0.01711}$ $[\mathbf{0.01338}, \mathbf{0.02084}]$
Γ_5: diagonal → dense	$+0.01144$ $[0.00868, 0.01419]$	$+0.00339$ $[0.00056, 0.00621]$

Table 10: Parameter and effective-function audit of the six-level ladder. Δp_{raw} counts newly trainable parameters; Δp_{fwd} excludes parameters masked out of the realized forward operator. "Coordinates" is the number of newly emitted restriction coordinates per edge and layer. Novelty is the mean randomized tangent-novelty ratio and is a local proxy, not a formal effective dimension.

Rung	$\Delta p_{\text{raw}}/\Delta p_{\text{fwd}}$	Coordinates	Roman novelty	Mine novelty
Gate	8,322/8,322	1	1.000	1.000
Sign	66/66	1	0.334	0.292
Rotation	66/66	1	0.970	0.800
Diagonal	264/264	4	0.696	0.511
Dense	1,056/792	12	0.843	0.599

Sign is practically inert on both. The pre-specified prediction that the dense rung would collapse, extrapolated from a frame-aggregation classifier theorem, was *not* supported: both dense increments exclude zero. Its failure bounds that transfer rather than contradicting the theorem. Adjacent increments telescope exactly (0.16366 and 0.10882, zero discrepancy at every unit), and the parameter audit separates 1,056 raw new dense parameters from the 792 that survive the off-diagonal mask.

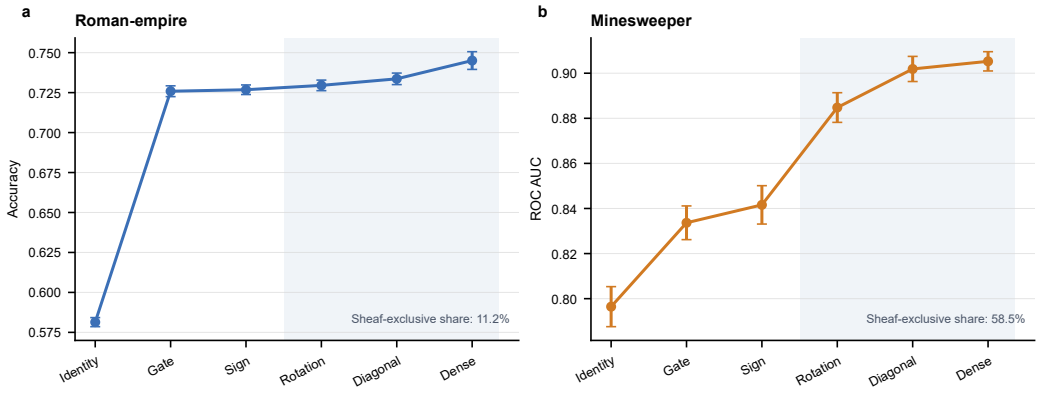


Figure 4: Audited capability ladders. Shading marks rungs whose incoming increments require matrix-valued transport; labels give their share of the identity-to-dense gain.

A.7 A DIRECTIONAL PREDICTION THE DATA DO NOT SUPPORT

The structure-group argument of Section 3 suggests that *which* capability is useful should depend on *why* the graph needs fixing. Where identity propagation already helps, the missing capability is alignment, supplied by Γ_3. Where a no-graph baseline beats identity propagation, the missing capability is selective suppression, which orthogonal transport cannot supply and which first appears at $\mathcal{M}_4$. That gives a directional prediction on

$$D = (\Gamma_4 + \Gamma_5) - \Gamma_3, \qquad D_{\text{graph-harmful}} > D_{\text{graph-helpful}}.$$

We tested it, and the data do not support it.

We test a *difference* rather than the ratio $(\Gamma_4 + \Gamma_5)/\Gamma_3$, because Γ_3 is not bounded away from zero and a denominator crossing zero produces meaningless values — the failure mode this paper's own diagnostic is built to refuse. Regime membership is *measured* by $G_{\text{graph}}(\mathcal{M}_0)$ under this architecture, not taken from published MLP-minus-GCN gaps, and datasets whose entire sheaf-exclusive segment lies within $\pm\epsilon_\Gamma$ of zero are excluded rather than averaged in as zeros.

Table 11: Test of the directional prediction $D_{\text{harmful}} > D_{\text{helpful}}$ with $D = (\Gamma_4 + \Gamma_5) - \Gamma_3$, at two layers. Regime is *measured* by $G_{\text{graph}}(\mathcal{M}_0)$. Intervals are 95% t intervals over the ten official split means. "Inert": the whole sheaf-exclusive segment lies within ± 0.01, so D carries no allocation information. WebKB and Actor were run on a laptop GPU, the other rows on A100 hardware.

Dataset	Regime	$G_{\text{graph}}(\mathcal{M}_0)$	Γ_3	$\Gamma_4 + \Gamma_5$	D (95% CI)
Roman-empire	harmful	-0.0783	$+0.0018$	$+0.0171$	$+0.0153$ $[+0.0116, +0.0190]$
Wisconsin	harmful	-0.3593	$+0.0054$	$+0.0142$	$+0.0088$ $[-0.0127, +0.0303]$
Cornell	harmful	-0.3209	$+0.0189$	$+0.0068$	-0.0122 $[-0.0380, +0.0137]$
Texas	harmful	-0.2182	$+0.0230$	$+0.0034$	-0.0196 $[-0.0455, +0.0063]$
Actor	harmful	-0.0537	-0.0034	$+0.0003$	inert
Minesweeper	helpful	$+0.2866$	$+0.0376$	$+0.0201$	-0.0175 $[-0.0295, -0.0055]$
Tolokers	helpful	$+0.0384$	$+0.0086$	$+0.0050$	-0.0036 $[-0.0061, -0.0011]$
Amazon-ratings	helpful	$+0.0556$	$+0.0012$	$+0.0003$	inert
Questions	helpful	$+0.0200$	-0.0013	$+0.0020$	inert

The prediction is not supported. Among graph-harmful tasks only Roman-empire separates from zero in the predicted direction ($D = +0.0153$). The three WebKB tasks are unresolved: their intervals all contain zero, and two of the three point estimates (-0.0122, -0.0196) lie on the side of the graph-helpful tasks (Minesweeper -0.0175, Tolokers -0.0036). The WebKB test sets hold 37–51 nodes, an accuracy quantum of 0.020–0.027, which is coarser than the increments being compared; with that resolution the design cannot confirm the prediction, and its point estimates give it no encouragement. Actor, Amazon-ratings and Questions are excluded as inert. We report this as an unsupported prediction rather than omitting it. It agrees with the paper's other negative finding on scalar summaries: in an exploratory six-dataset check the learned-gate increment was uncorrelated with identity graph value ($r = -0.395$, $p = 0.438$), average degree ($r = -0.025$) and log average degree ($r = -0.196$). Coarse regime labels do not predict capacity allocation at the gate rung, and nothing here suggests they do at the rotation rung.

A.8 SND: CALIBRATION, SCREEN, AND FINGERPRINT

Table 12 is a falsification-oriented unit test. All four pre-specified cells are classified correctly, and the decisive observation is that a large positive Ψ is *not* sufficient: the intervention-collapse cell produces essentially the same raw interaction ($+0.323$) as genuine property-specific necessity ($+0.320$). Only the task-retention gate separates them. Equal pre- and post-intervention capacity effects correctly identify generic capacity, and an inert parent rung blocks interpretation even when the intervention itself is benign.

We then screened six real datasets. Within each true-label stratum we permuted a fraction of node feature vectors, leaving graph, labels, split and class-conditional feature multiset exactly unchanged; the intervention dose is the measured relative reduction in feature Moran's I, not the nominal fraction.

The screen separates capacity value from property-specific attribution. Roman-empire has by far the largest gate increment, yet destroying 90.9% of its measured feature spatial autocorrelation leaves that increment unchanged ($\Psi = -0.0001\,[-0.0053, 0.0051]$): its gate benefit is not attributable to this feature–topology correspondence. Tolokers shows the opposite pattern with a smaller increment but a practically positive interaction. Cora and Questions show why a null Ψ is uninformative when the tested rung is inert.

Finally we repeated the complete warm-started ladder after a 25% within-label permutation and formed $\Psi_{j,k}$ for every rung.

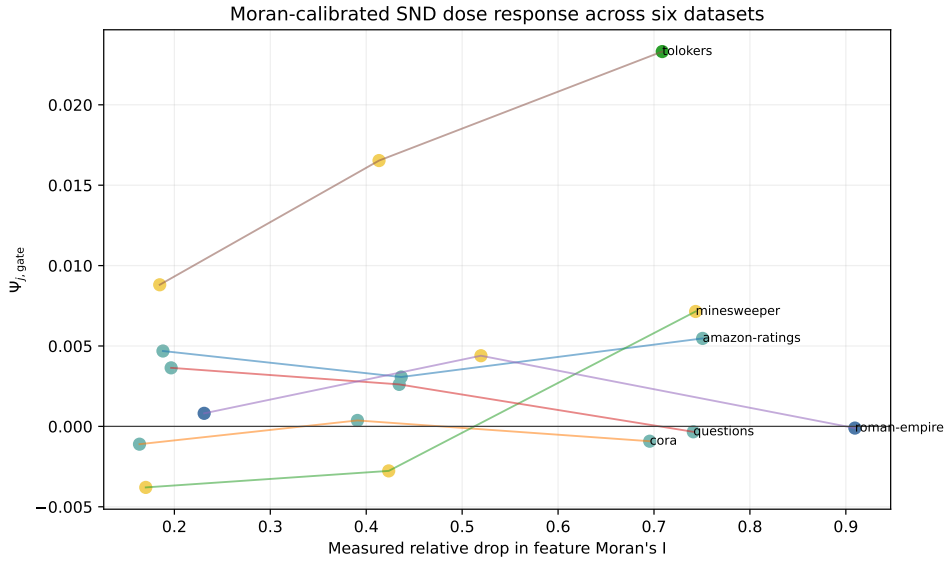


Figure 5: Moran-calibrated dose response. Colors encode the validity-gated decision rather than the sign alone.

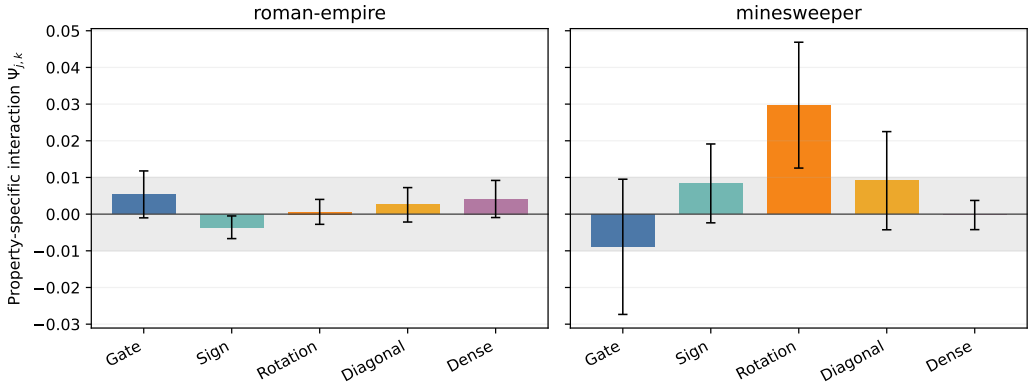


Figure 6: Property-specific capability fingerprints. The gray band is the practical-equivalence region $[-0.01, 0.01]$.

Table 12: **Known-truth calibration of the SND decision rule.** Means are over 40 independent graph draws. Γ^+ and Γ^- are the capacity effects before and after intervention, $\Psi = \Gamma^+ - \Gamma^-$, and retention measures remaining parent-task support. The raw interaction is nearly identical in the necessary and collapse cells; the validity gate is what distinguishes them.

Known mechanism	Γ^+	Γ^-	Ψ	Retention	SND decision
Property-specific necessity	$+0.320$	-0.000	$+0.320$	0.58	Supported positive
Property irrelevant	-0.000	-0.000	-0.000	0.95	Capacity inert
Generic capacity benefit	$+1.460$	$+1.460$	0.000	0.28	Null/equivalent
Intervention collapse	$+0.318$	-0.005	$+0.323$	-0.01	Non-identifiable

Table 13: Six-dataset SND screen at nominal permutation fraction 0.5 (ten split-level units). "Moran drop" is the measured relative reduction in feature Moran's I; retention is the intervened identity score above chance divided by its original counterpart. C16 requires the original gate increment to clear $\epsilon_\Gamma = 0.01$. Decisions combine intervention integrity, C16, retained task support, and the paired Ψ interval. Tolokers passes those but fails G2 under the nuisance audit of Appendix H: its interaction is reported and the property-specific reading is refused.

Dataset	Moran drop	$\Gamma_{\text{gate}}(P)$	$\Psi_{j,\text{gate}}$ (95% CI)	Retention	Decision
Amazon-ratings	0.751	$+0.0107$	$+0.0055\,[-0.0014, +0.0123]$	1.02	C16-inert
Cora	0.696	-0.0011	$-0.0009\,[-0.0035, +0.0017]$	1.03	C16-inert
Minesweeper	0.743	$+0.0528$	$+0.0071\,[-0.0146, +0.0289]$	0.56	unresolved
Questions	0.741	-0.0000	$-0.0003\,[-0.0045, +0.0038]$	0.90	C16-inert
Roman-empire	0.909	$+0.1430$	$-0.0001\,[-0.0053, +0.0051]$	0.96	equivalent
Tolokers	0.709	$+0.0412$	$+0.0233\,[+0.0119, +0.0347]$	1.03	G2 failed: refused

On Minesweeper rotation falls from $\Gamma_3^+ = +0.03259$ to $\Gamma_3^- = +0.00287$, the only positive practical interaction, $\Psi_{j,3} = +0.02972\,[0.01256, 0.04687]$. The property-specific effect is thus not an endpoint artifact: it localizes to the same sheaf-exclusive rung that dominated the original ladder. Roman-empire resolves no rung positively. The control arm of this batch gives $\Gamma_3^+ = +0.03259$ where the main ladder gives $+0.04317$. These are independently trained batches of the same contrast and are not expected to agree numerically, but the 0.0106 gap is about a third of the reported $\Psi_{j,3}$, so the interaction should be read with that batch-to-batch variability in mind; the A100 re-run of Section A.9 gives $+0.0376$ for the same increment.

A.9 Does the attribution survive a change of depth and of accuracy?

The ladder above was first measured at two propagation layers, where its dense endpoint is about 0.75 on Roman-empire and 0.90 on Minesweeper, against published 0.804 and 0.922 for a neural sheaf diffusion model and 0.918 and 0.990 for a bundle model. An attribution measured below competitive accuracy may not describe models operating at it, so we first separate the causes of the gap and then repeat the audited ladder at a second depth.

Table 14: Property-specific interactions $\Psi_{j,k}$ along the six-level ladder after a 25% within-label feature permutation. Entries are means with paired 95% t intervals over ten official splits. Bold denotes a lower interval endpoint above the practical margin 0.01; values whose complete interval lies inside $[-0.01, 0.01]$ are practically equivalent.

Capability increment	Roman-empire	Minesweeper
Gate	$+0.00538[-0.00101, +0.01178]$	$-0.00891[-0.02735, +0.00952]$
Sign	$-0.00358[-0.00667, -0.00049]$	$+0.00838[-0.00237, +0.01913]$
Rotation	$+0.00062[-0.00278, +0.00401]$	$\mathbf{+0.02972[+0.01256, +0.04687]}$
Diagonal	$+0.00254[-0.00215, +0.00723]$	$+0.00912[-0.00427, +0.02250]$
Dense	$+0.00413[-0.00093, +0.00919]$	$-0.00025[-0.00422, +0.00373]$

Table 15: The effect of a 25% within-label feature permutation on the dense model, split exactly into the part already present at the identity rung, the part collected by the scalar rungs and the part collected by the sheaf-exclusive rungs ($\sum_k \Psi_{j,k}$ over each block; blocks as in Section 4). Paired 95% t intervals over ten official splits; †: split-paired depth difference excludes zero. Per-rung values are in Table 16.

Dataset	Collected at	Two layers	Four layers	Four minus two
Minesweeper	Identity rung	$+0.061\,[+0.055, +0.067]$	$+0.031\,[+0.028, +0.034]$	$-0.030\,[-0.036, -0.024]^\dagger$
	Scalar (gate, sign)	$-0.001\,[-0.022, +0.021]$	$+0.058\,[+0.052, +0.064]$	$+0.059\,[+0.035, +0.083]^\dagger$
	Sheaf-exclusive	$+0.039\,[+0.019, +0.058]$	$+0.005\,[-0.004, +0.013]$	$-0.034\,[-0.058, -0.010]^\dagger$
	Total	$+0.099\,[+0.091, +0.107]$	$+0.094\,[+0.082, +0.106]$	$-0.005\,[-0.014, +0.004]$
Roman-empire	Identity rung	$+0.011\,[+0.005, +0.017]$	$+0.001\,[-0.005, +0.006]$	$-0.010\,[-0.017, -0.004]^\dagger$
	Scalar (gate, sign)	$+0.002\,[-0.005, +0.009]$	$+0.000\,[-0.006, +0.007]$	$-0.001\,[-0.010, +0.007]$
	Sheaf-exclusive	$+0.007\,[+0.002, +0.013]$	$+0.024\,[+0.016, +0.032]$	$+0.017\,[+0.007, +0.026]^\dagger$
	Total	$+0.020\,[+0.013, +0.028]$	$+0.025\,[+0.017, +0.034]$	$+0.005\,[-0.004, +0.014]$

Table 16: Where the property-specific interaction lands depends on depth. Γ_k is the increment on the original data and $\Psi_{j,k}$ its change under a 25% within-label feature permutation; the identity row is the score difference at the identity rung, so each column of Ψ sums exactly to the total effect of the intervention on the dense model. Means and paired 95% t intervals over ten official splits (laptop GPU; original and intervened ladders share hardware and seeds). Bold: lower endpoint above the practical margin 0.01. $^\circ$: interval inside $[-0.01, 0.01]$. †: split-paired depth difference excludes zero.

		Two layers		Four layers		Depth difference
Dataset	Rung	Γ_k	$\Psi_{j,k}$	Γ_k	$\Psi_{j,k}$	$\Psi^{(4)} - \Psi^{(2)}$
Minesweeper	Identity ($\Psi_{j,0}$)	—	$+\mathbf{0.061}\,[+\mathbf{0.055}, +\mathbf{0.067}]$	—	$+\mathbf{0.031}\,[+\mathbf{0.028}, +\mathbf{0.034}]$	$-0.030\,[-0.036, -0.024]^\dagger$
	Gate	$+0.047$	$-0.009\,[-0.027, +0.010]$	$+0.126$	$+\mathbf{0.047}\,[+\mathbf{0.040}, +\mathbf{0.055}]$	$+0.056\,[+0.040, +0.073]^\dagger$
	Sign	$+0.013$	$+0.008\,[-0.002, +0.019]$	$+0.020$	$+0.011\,[+0.003, +0.019]$	$+0.002\,[-0.008, +0.013]$
	Rotation	$+0.033$	$+\mathbf{0.030}\,[+\mathbf{0.013}, +\mathbf{0.047}]$	$+0.007$	$+0.003\,[-0.003, +0.008]^\circ$	$-0.027\,[-0.048, -0.006]^\dagger$
	Diagonal	$+0.016$	$+0.009\,[-0.004, +0.023]$	$+0.006$	$+0.004\,[-0.000, +0.008]^\circ$	$-0.005\,[-0.020, +0.010]$
	Dense	$+0.003$	$-0.000\,[-0.004, +0.004]^\circ$	$+0.005$	$-0.002\,[-0.011, +0.006]$	$-0.002\,[-0.013, +0.009]$
	Total	—	$+0.099\,[+0.091, +0.107]$	—	$+0.094\,[+0.082, +0.106]$	$-0.005\,[-0.014, +0.004]$
Roman-empire	Identity ($\Psi_{j,0}$)	—	$+0.011\,[+0.005, +0.017]$	—	$+0.001\,[-0.005, +0.006]^\circ$	$-0.010\,[-0.017, -0.004]^\dagger$
	Gate	$+0.150$	$+0.005\,[-0.001, +0.012]$	$+0.350$	$+0.004\,[-0.003, +0.010]^\circ$	$-0.002\,[-0.010, +0.006]$
	Sign	-0.002	$-0.004\,[-0.007, -0.000]^\circ$	$+0.001$	$-0.003\,[-0.008, +0.002]^\circ$	$+0.000\,[-0.004, +0.005]$
	Rotation	$+0.003$	$+0.001\,[-0.003, +0.004]^\circ$	$+0.019$	$+0.009\,[-0.005, +0.022]$	$+0.008\,[-0.006, +0.022]$
	Diagonal	$+0.002$	$+0.003\,[-0.002, +0.007]^\circ$	$+0.023$	$+0.009\,[+0.004, +0.014]$	$+0.007\,[-0.001, +0.014]$
	Dense	$+0.012$	$+0.004\,[-0.001, +0.009]^\circ$	$+0.025$	$+0.006\,[-0.002, +0.014]$	$+0.002\,[-0.009, +0.012]$
	Total	—	$+0.020\,[+0.013, +0.028]$	—	$+0.025\,[+0.017, +0.034]$	$+0.005\,[-0.004, +0.014]$

Two causes are separable. First, the nesting protocol itself costs accuracy: at identical architecture and depth, the dense endpoint on Roman-empire rises from 0.7451 to 0.7926 when trained freely instead of warm-started under the parent training-loss cap with a shorter budget. Interpretability of Γ_k is not free, and we report its price. Second, depth: Roman-empire is a chain graph of diameter $6{,}824$, so a two-layer model sees two hops. Free training at four layers reaches 0.8238 on Roman-empire, above the neural sheaf diffusion reference, and 0.9183 on Minesweeper, 0.3 points below it; six layers regresses to 0.8106 on Roman-empire. These free-training numbers come from a small depth sweep (two seeds per configuration) that we used to choose the second depth; they are not audited ladders.

We then re-ran the full audited ladder at two and at four layers on A100 cluster hardware (PyTorch 2.0.1, whereas the first run used a laptop GPU with PyTorch 2.6), on Roman-empire, Minesweeper, Tolokers and Amazon-ratings, with 40 optimizer units per dataset and depth. Every other setting is identical between the two depths, and the restriction-map heads are untouched by the change. One Minesweeper array task at two layers aborted on the numerical audit described below and was re-run after the float64 adjudication was added; every cell of the table now has all 40 units.

The two-layer results replicate, except one ordering. The sheaf-exclusive magnitudes reproduce. Roman-empire gives $\eta_{\text{sheaf}} = 11.4\%$ against 11.2% in the first run. Minesweeper's sheaf-exclusive gain is $+0.0577\,[0.0485, 0.0670]$ (a t interval over the ten official split means, as throughout this section), an interval that contains the first run's $+0.0637$; the share is 53.0% against 58.5%,

and it moves more than the gain because it also depends on the gate and sign increments in its denominator. What does *not* replicate is that rotation exceeds gate on Minesweeper: rotation minus gate is $-0.0020\,[-0.0127, +0.0086]$, so the two are not distinguishable. The statement that rotation is the largest single Minesweeper increment therefore belongs to the first run only and we do not carry it forward.

Depth changes the allocation, and in opposite directions. Table 17 gives the paired contrast between depths.

Table 17: Paired depth contrast, $L{=}2$ minus $L{=}4$, over seeds present at both depths; intervals are t intervals over the ten official split means. The two tasks move in opposite directions on the sheaf-exclusive gain: it falls with depth on Minesweeper and rises on Roman-empire.

Dataset	n	Δ Sheaf (95% CI)	$\Delta\,\Gamma_1$ (95% CI)	$\Delta\,S(\text{dense})$ (95% CI)
Minesweeper	40	$+0.0368\,[+0.0227, +0.0509]$	$-0.0884\,[-0.0964, -0.0804]$	$+0.0002\,[-0.0036, +0.0041]$
Roman-empire	40	$-0.0503\,[-0.0551, -0.0455]$	$-0.2016\,[-0.2065, -0.1968]$	$-0.0312\,[-0.0368, -0.0256]$
Tolokers	40	$-0.0065\,[-0.0129, -0.0002]$	$+0.0075\,[+0.0018, +0.0131]$	$+0.0174\,[+0.0087, +0.0262]$
Amazon-ratings	40	$+0.0003\,[-0.0015, +0.0021]$	$+0.0013\,[-0.0017, +0.0043]$	$+0.0033\,[+0.0006, +0.0060]$

On Minesweeper the dense endpoint is unchanged (0.9049 to 0.9047; paired difference $+0.0002\,[-0.0036, +0.0041]$), yet the sheaf-exclusive gain falls from $+0.0577$ to $+0.0209$ and η_{sheaf} from 53.0% to 12.9%. At four layers identity propagation degrades from 0.796 to 0.742 and the gate rung, $+0.1280$, recovers it: the credit moves to the gate while the endpoint stays put. On Roman-empire the opposite occurs. The sheaf-exclusive gain rises from $+0.0188$ to $+0.0691$, with rotation, diagonal and dense contributing about $+0.02$ to $+0.03$ each, and the endpoint improves by 0.031 to 0.778. Identity propagation on this chain graph collapses to 0.361, so the gate increment, $+0.3473$, is largely repair of an aggregation rule that is harmful at depth four. Tolokers moves slightly in the same direction as Roman-empire (paired $-0.0065\,[-0.0129, -0.0002]$), and Amazon-ratings does not move ($+0.0003\,[-0.0015, +0.0021]$), where the sheaf-exclusive segment is at most 0.0015 at either depth.

Two conclusions follow. First, the split of credit between attenuation-like and sheaf-exclusive capabilities is not a property of the task: the same task and the same endpoint move from 53% to 13% under a hyperparameter that does not touch the restriction maps. Second, η_{sheaf} alone can conceal this. On Roman-empire it changes only from 11.4% to 16.6% while the absolute sheaf-exclusive gain grows 3.7-fold, because the gate increment in its denominator is inflated by the repair of a degraded identity baseline. We therefore report the absolute sheaf-exclusive gain together with η_{sheaf}.

Is a null increment an optimization failure? A rung whose increment is zero may have nothing to offer, or may simply not have been trained well. The fit logs separate three cases (Table 18). Each child is warm-started from its parent and the parent stays eligible, so when no trained checkpoint validates better the parent is retained and the increment is exactly zero. We call such a rung *fit-only* if the child nevertheless lowered training cross-entropy by at least 1% (a better fit was found and did not generalize) and *no-descent* if it never did within the budget (the null is indistinguishable from an optimization failure). The gate rung moved in all 315 units. On Amazon-ratings, whose sheaf-exclusive gain is practically zero, more than half of the rung fits above the gate retain the parent and *every one* of them is fit-only (median training cross-entropy reduction 4–10%), with tangent novelty above 0.84 on the sheaf-exclusive rungs: that null is a generalization null, not an optimization or a redundancy null. Roman-empire has no no-descent sheaf-exclusive rung at either depth. No-descent does occur at four layers on Minesweeper (19% of rung fits) and Tolokers (30%). Restricting to units with no no-descent sheaf-exclusive rung raises the sheaf-exclusive gain slightly and leaves every depth conclusion in place: on Minesweeper $+0.064$ and $+0.025$ against $+0.058$ and $+0.021$ on all units, with paired contrast $+0.034\,[0.020, 0.049]$ on the 19 seeds clean at both depths against $+0.037\,[0.023, 0.051]$; on Tolokers the four-layer gain rises from $+0.020$ to $+0.027$ on the 14 clean units, so that cell is a lower bound and its depth contrast ($-0.015\,[-0.028, -0.003]$ on clean units against -0.007) should be read as such. The clean subset is selected on an outcome of training and is reported as a sensitivity analysis, not as a corrected estimate.

Table 18: Anatomy of the rungs above the gate (sign to dense, four rung fits per unit; A100 runs). "Moved": a trained checkpoint was selected. "Fit-only": the parent was retained although the child lowered training cross-entropy by at least 1%, so the better fit never validated. "No-descent": the parent was retained and training cross-entropy never fell by 1% within the budget, so the null cannot be separated from an optimization failure. The last columns give the sheaf-exclusive gain on all units and on units with no no-descent sheaf-exclusive rung (95% t intervals over split means).

Dataset	L	Units	Moved (%)	Fit-only (%)	No-descent (%)	Sheaf gain, all units	Sheaf gain, clean units	Clean
Minesweeper	2	40	84	6	10	$+0.058\,[+0.048, +0.067]$	$+0.064\,[+0.056, +0.072]$	32
	4	40	72	9	19	$+0.021\,[+0.013, +0.029]$	$+0.025\,[+0.016, +0.033]$	25
Roman-empire	2	40	76	24	0	$+0.019\,[+0.016, +0.022]$	$+0.019\,[+0.016, +0.022]$	40
	4	40	84	14	1	$+0.069\,[+0.066, +0.072]$	$+0.069\,[+0.066, +0.072]$	40
Tolokers	2	40	82	17	1	$+0.014\,[+0.010, +0.017]$	$+0.014\,[+0.010, +0.017]$	39
	4	40	56	14	30	$+0.020\,[+0.015, +0.025]$	$+0.027\,[+0.017, +0.037]$	14
Amazon-ratings	2	40	46	54	0	$+0.002\,[-0.000, +0.003]$	$+0.002\,[-0.000, +0.003]$	40
	4	40	41	59	0	$+0.001\,[-0.000, +0.003]$	$+0.001\,[-0.000, +0.003]$	40

Table 19: The audited gain split by the algebra the restriction maps generate (Theorem 1): real scalars (gate and sign rungs), complex scalars (the rotation rung, whose blocks share one angle), two independent 2×2 blocks (rotated diagonal) and the full matrix algebra (dense). Increment with 95% t interval over split means, and its share of the total gain in percent (A100 runs).

			$\mathbb{R}$		$\mathbb{C}$		$M_2(\mathbb{R})^2$		$M_4(\mathbb{R})$	
Dataset	L	Units	gain	%	gain	%	gain	%	gain	%
Minesweeper	2	40	$+0.051\,[+0.040, +0.062]$	47	$+0.038\,[+0.030, +0.045]$	35	$+0.018\,[+0.011, +0.025]$	17	$+0.002\,[+0.001, +0.003]$	2
	4	40	$+0.142\,[+0.135, +0.149]$	87	$+0.009\,[+0.006, +0.012]$	6	$+0.006\,[+0.002, +0.009]$	3	$+0.006\,[+0.002, +0.010]$	4
Roman-empire	2	40	$+0.147\,[+0.143, +0.151]$	89	$+0.002\,[+0.000, +0.003]$	1	$+0.004\,[+0.002, +0.006]$	2	$+0.013\,[+0.012, +0.015]$	8
	4	40	$+0.348\,[+0.346, +0.351]$	83	$+0.020\,[+0.016, +0.024]$	5	$+0.023\,[+0.021, +0.026]$	6	$+0.026\,[+0.023, +0.029]$	6
Tolokers	2	40	$+0.045\,[+0.039, +0.051]$	77	$+0.009\,[+0.007, +0.010]$	15	$+0.004\,[+0.002, +0.006]$	6	$+0.001\,[-0.001, +0.003]$	2
	4	40	$+0.042\,[+0.035, +0.049]$	68	$+0.009\,[+0.006, +0.012]$	14	$+0.006\,[+0.004, +0.008]$	10	$+0.005\,[+0.002, +0.008]$	9
Amazon-ratings	2	40	$+0.008\,[+0.006, +0.010]$	84	$+0.001\,[-0.000, +0.003]$	13	$+0.000\,[-0.000, +0.001]$	2	$+0.000\,[-0.001, +0.001]$	0
	4	40	$+0.008\,[+0.006, +0.010]$	87	$+0.000\,[-0.001, +0.001]$	1	$+0.001\,[+0.000, +0.003]$	16	$-0.000\,[-0.001, +0.001]$	-3

Depth under both operators. Table 20 gives the ladder under both operators at both depths (ten seeds per cell, the same seeds in every cell, laptop GPU; no float32 embedding check exceeded the tolerance in the single-product runs). On Roman-empire the two operators agree at four layers up to $-0.010\,[-0.020, -0.000]$ in sheaf-exclusive gain; on Minesweeper they differ by $+0.076\,[0.060, 0.092]$. At four layers the no-descent rate above the gate is 25% under our operator and 8% under the released one on Minesweeper, which the sensitivity analysis of Table 18 shows is too small an effect to account for the gap. The comparison changes assembly and normalization together.

Table 20: Depth dependence under both operators: same ten official splits, seeds and hardware (laptop GPU). Sheaf-exclusive gain with paired 95% t intervals over splits; †: split-paired depth difference excludes zero.

Dataset	Operator	L	$S(\mathcal{M}_0)$	$S(\mathcal{M}_5)$	Γ_1	Sheaf-exclusive gain	η_{sheaf}
Minesweeper	averaged, row-norm.	2	0.795	0.906	$+0.047$	$+0.051\,[+0.033, +0.069]$	46%
		4	0.742	0.906	$+0.126$	$+0.017\,[+0.011, +0.023]$	10%
		4 minus 2				$-0.034\,[-0.055, -0.013]^\dagger$	
	single product, aug.-sym.	2	0.804	0.903	$+0.038$	$+0.058\,[+0.047, +0.070]$	59%
		4	0.745	0.912	$+0.067$	$+0.093\,[+0.080, +0.106]$	56%
		4 minus 2				$+0.035\,[+0.024, +0.045]^\dagger$	
Roman-empire	averaged, row-norm.	2	0.578	0.743	$+0.150$	$+0.018\,[+0.013, +0.022]$	11%
		4	0.357	0.776	$+0.350$	$+0.068\,[+0.060, +0.075]$	16%
		4 minus 2				$+0.050\,[+0.042, +0.058]^\dagger$	
	single product, aug.-sym.	2	0.712	0.762	$+0.033$	$+0.018\,[+0.013, +0.023]$	35%
		4	0.583	0.787	$+0.143$	$+0.058\,[+0.052, +0.064]$	28%
		4 minus 2				$+0.040\,[+0.034, +0.046]^\dagger$	

A third dataset under the released operator. Table 21 adds Tolokers. Under our operator its sheaf-exclusive gain is $+0.014$ at two layers and $+0.020$ at four (Table 1); under the released operator it is $+0.003\,[-0.001, 0.007]$ at two layers and $+0.008\,[-0.002, 0.017]$ at four, from eight and four seeds respectively, the remaining seeds still running on the cluster at submission. Tolokers is a task on which matrix-valued transport contributes little under either operator: the gain is small

under both, not resolved under the released one, and its sign does not change with depth, so the third dataset neither reproduces the Minesweeper reversal nor contradicts the Roman-empire rise.

Table 21: A third dataset under the released operator (single-product assembly, augmented symmetric normalization), same protocol as Table 20. Entries are means over n seeds (seed $\rightarrow$ official split; ROC AUC), Γ_k are adjacent increments, "sheaf" is $\Gamma_3 + \Gamma_4 + \Gamma_5$ with a 95% t interval over seeds and η its share of the identity-to-dense gain. Cells with fewer than ten seeds were still running on the cluster at submission and are reported as they stand.

Dataset	L	n	Identity	Dense	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Sheaf	η
Tolokers	2	8	0.7870	0.8320	+0.0414	+0.0006	+0.0007	+0.0018	+0.0007	$+0.0031\,[-0.0010, +0.0072]$	7%
Tolokers	4	4	0.7818	0.8435	+0.0447	+0.0095	+0.0033	+0.0013	+0.0030	$+0.0075\,[-0.0022, +0.0172]$	12%

The assembly $\times$ normalization grid on two further datasets. Tables 22 and 23 repeat the 2×2 grid of Section 5.2 on Tolokers and Amazon-ratings, ten seeds per cell with seeds shared across cells. The two datasets answer differently, and neither reproduces the Minesweeper picture. On Tolokers the normalization does move the depth difference of the sheaf-exclusive gain, but with the *opposite* sign: $-0.019\,[-0.028, -0.010]$ under the averaged assembly and $-0.001\,[-0.016, 0.013]$ under the single product, against $+0.044\,[0.007, 0.081]$ and $+0.061\,[0.035, 0.086]$ for the same contrast on Minesweeper, so on Tolokers it is the row-normalized cell whose gain grows with depth ($+0.012 \rightarrow +0.029$). On Amazon-ratings the sheaf-exclusive gain is at most $+0.003$ in every cell and no main effect is resolved beyond ± 0.004: with no gain to allocate, the grid has nothing to say. What replicates across the three datasets is the claim the grid was built to test, that the depth behaviour of the sheaf-exclusive gain is a property of the operator and not of the task alone; what does not replicate is the direction of the normalization effect, which on Minesweeper favours the augmented symmetric normalization at depth and on Tolokers the row normalization. The mechanism of Section 5.2, a scalar gate that row normalization allows to leave the contraction regime, explains a sign reversal on Minesweeper; it does not predict which side of the reversal a given task lands on, and we do not claim it does.

Table 22: Assembly $\times$ normalization on two further datasets, the same quantities as Table 2: the sheaf-exclusive gain $\Gamma_3 + \Gamma_4 + \Gamma_5$ at two and four layers, its seed-paired depth difference, and the gate increment Γ_1. Cells share seeds (seed $\rightarrow$ official split), so the depth differences are paired; †: interval excludes zero. Cells with fewer than ten seeds were still running on the cluster at submission.

Dataset	Assembly	Normalization	n	Gain, L=2	Gain, L=4	L=4 minus L=2	Γ_1, L=2	Γ_1, L=4
Tolokers	averaged	row	9	$+0.011\,[+0.008, +0.015]^\dagger$	$+0.029\,[+0.020, +0.039]^\dagger$	$+0.018\,[+0.009, +0.027]^\dagger$	+0.043	+0.030
Tolokers	averaged	aug.-sym.	10	$+0.013\,[+0.009, +0.017]^\dagger$	$+0.012\,[+0.007, +0.018]^\dagger$	$-0.001\,[-0.006, +0.005]$	+0.036	+0.042
Tolokers	single product	row	9	$+0.010\,[+0.004, +0.017]^\dagger$	$+0.017\,[+0.008, +0.025]^\dagger$	$+0.007\,[-0.005, +0.018]$	+0.042	+0.032
Tolokers	single product	aug.-sym.	9	$+0.003\,[-0.001, +0.006]$	$+0.008\,[+0.003, +0.013]^\dagger$	$+0.005\,[-0.002, +0.012]$	+0.042	+0.046
Amazon-ratings	averaged	row	10	$+0.001\,[-0.002, +0.004]$	$+0.002\,[-0.002, +0.006]$	$+0.001\,[-0.002, +0.005]$	+0.008	+0.007
Amazon-ratings	averaged	aug.-sym.	10	$+0.002\,[-0.001, +0.005]$	$+0.003\,[-0.000, +0.006]$	$+0.001\,[-0.002, +0.004]$	+0.008	+0.005
Amazon-ratings	single product	row	10	$+0.002\,[-0.001, +0.005]$	$+0.003\,[-0.000, +0.007]$	$+0.002\,[-0.002, +0.005]$	+0.007	+0.004
Amazon-ratings	single product	aug.-sym.	10	$+0.002\,[-0.000, +0.004]$	$+0.000\,[-0.002, +0.002]$	$-0.002\,[-0.003, -0.000]^\dagger$	+0.010	+0.008

Table 23: Main effects on the depth difference of the sheaf-exclusive gain, on the two further datasets of Table 22, paired over the seeds the two cells share. On Minesweeper the normalization effect is $+0.060$ and $+0.063$ under the two assemblies (Section 5.2); †: interval excludes zero.

Dataset	Effect	n	Change of the depth difference
Tolokers	normalization (averaged)	9	$-0.019\,[-0.028, -0.010]^\dagger$
Tolokers	normalization (product)	9	$-0.001\,[-0.016, +0.013]$
Tolokers	assembly (row)	9	$-0.011\,[-0.023, +0.001]$
Tolokers	assembly (aug.-sym.)	9	$+0.006\,[-0.003, +0.016]$
Amazon-ratings	normalization (averaged)	10	$+0.000\,[-0.004, +0.004]$
Amazon-ratings	normalization (product)	10	$-0.003\,[-0.007, +0.000]$
Amazon-ratings	assembly (row)	10	$+0.000\,[-0.005, +0.006]$
Amazon-ratings	assembly (aug.-sym.)	10	$-0.003\,[-0.006, +0.001]$

Assembly $\times$ normalization. Table 2 of the main text fills the two missing cells of the grid on Minesweeper (ten seeds per cell and depth, the same seeds everywhere). The effect of the normaliza-

Table 24: The two-layer ladder under two assemblies, same ten official splits, seeds and hardware (laptop GPU). "Averaged": one map per stored direction with row normalization, the layer of the main study. "Single product": $F_{u\trianglelefteq e}^{\top}F_{v\trianglelefteq e}$ with augmented symmetric normalization, the operator of released sheaf-diffusion code, to which Proposition 4 does not apply. Increments are means; intervals are paired 95% t intervals over splits. The averaged rows use these ten seeds (Minesweeper $\eta_{\mathrm{sheaf}} = 46\%$, against 53% over forty units in Table 1).

Dataset	Assembly	$S(\mathcal{M}_0)$	$S(\mathcal{M}_5)$	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Sheaf-exclusive gain	η_{sheaf}
Minesweeper	averaged, row-norm.	0.795	0.906	$+0.047$	$+0.013$	$+0.033$	$+0.016$	$+0.003$	$+0.051\,[+0.033, +0.069]$	46%
	single product, aug.-sym.	0.804	0.903	$+0.038$	$+0.002$	$+0.037$	$+0.008$	$+0.013$	$+0.058\,[+0.047, +0.070]$	59%
	product minus averaged		$-0.004\,[-0.007, +0.000]$			$+0.005\,[-0.026, +0.036]$			$+0.008\,[-0.015, +0.030]$	
Roman-empire	averaged, row-norm.	0.578	0.743	$+0.150$	-0.002	$+0.003$	$+0.002$	$+0.012$	$+0.018\,[+0.013, +0.022]$	11%
	single product, aug.-sym.	0.712	0.762	$+0.033$	-0.001	$+0.003$	$+0.004$	$+0.011$	$+0.018\,[+0.013, +0.023]$	35%
	product minus averaged		$+0.019\,[+0.013, +0.025]$			$+0.000\,[-0.004, +0.004]$			$+0.000\,[-0.006, +0.006]$	

tion on the depth trend itself, that is on the four-minus-two-layer difference, is $+0.044\,[0.007, 0.081]$ under the averaged assembly and $+0.061\,[0.035, 0.086]$ under the single product; that of the assembly is $+0.008\,[-0.010, 0.026]$ under row normalization and $+0.025\,[0.005, 0.044]$ under the augmented symmetric one. At two layers no main effect and no interaction is resolved. The single-product, row-normalized cell is the least well optimized: 20–23% of its rung fits above the gate are no-descent at both depths (against 5–25% elsewhere), and it produced 3 and 11 float32 embedding exceedances at two and four layers, the largest 2.5×10^{-3}, all of which passed the float64 adjudication (largest 6.3×10^{-12}). The other three cells produced none.

The intervention budget under the released operator. Table 25 repeats the budget of Eq. (2) on Minesweeper with the single-product assembly and the augmented symmetric normalization (ten seeds per depth, the seeds of the main-text figure; no float32 embedding check exceeded the tolerance). Two features are common to both operators: the total does not move with depth ($-0.005\,[-0.015, 0.004]$ under ours, $+0.003\,[-0.007, 0.013]$ under the released one), and the part already present at the identity rung halves ($-0.030\,[-0.036, -0.024]$ and $-0.029\,[-0.041, -0.017]$). What differs is who collects the difference. Under our operator the scalar block gains $+0.059\,[0.035, 0.083]$ and the sheaf-exclusive block loses $-0.034\,[-0.058, -0.010]$; under the released operator the sheaf-exclusive block gains $+0.019\,[0.003, 0.034]$, most of it at the diagonal rung ($+0.016\,[0.005, 0.028]$, nine of ten splits), and the scalar block $+0.013\,[0.001, 0.025]$. The operator's effect on the depth change is $+0.053\,[0.020, 0.085]$ for the sheaf-exclusive block and $-0.046\,[-0.068, -0.023]$ for the scalar block, the same direction as in Table 2. At two layers the two operators are not distinguishable on any block.

Table 25: The intervention budget of Eq. (2) on Minesweeper under both operators (25% within-label feature permutation; ten official splits per cell, the same seeds everywhere, laptop GPU). Paired 95% t intervals over splits; †: split-paired depth difference excludes zero.

Operator	Collected at	Two layers	Four layers	Four minus two
averaged, row-norm.	Identity rung	$+0.061\,[+0.055, +0.067]$	$+0.031\,[+0.028, +0.034]$	$-0.030\,[-0.036, -0.024]^{\dagger}$
	Scalar rungs	$-0.001\,[-0.022, +0.021]$	$+0.058\,[+0.052, +0.064]$	$+0.059\,[+0.035, +0.083]^{\dagger}$
	Sheaf-exclusive rungs	$+0.039\,[+0.019, +0.058]$	$+0.005\,[-0.004, +0.013]$	$-0.034\,[-0.058, -0.010]^{\dagger}$
	Total	$+0.099\,[+0.091, +0.107]$	$+0.094\,[+0.082, +0.106]$	$-0.005\,[-0.014, +0.004]$
single product, aug.-sym.	Identity rung	$+0.065\,[+0.053, +0.076]$	$+0.036\,[+0.031, +0.040]$	$-0.029\,[-0.040, -0.017]^{\dagger}$
	Scalar rungs	$+0.016\,[+0.006, +0.026]$	$+0.029\,[+0.022, +0.036]$	$+0.013\,[+0.001, +0.025]^{\dagger}$
	Sheaf-exclusive rungs	$+0.020\,[+0.008, +0.032]$	$+0.039\,[+0.024, +0.053]$	$+0.019\,[+0.003, +0.034]^{\dagger}$
	Total	$+0.101\,[+0.093, +0.108]$	$+0.104\,[+0.093, +0.114]$	$+0.003\,[-0.007, +0.013]$

A second intervention and a second width. Table 26 repeats the budget of Eq. (2) on Minesweeper with a different property removed: degree-preserving double-edge swaps that change 22% of the edges and leave the degree sequence, features, labels and splits untouched (all 200 float32 embedding checks passed). The dense model loses 0.082 at two layers and 0.077 at four. With depth the identity rung's part falls and the gate's rises by a similar amount, as under the feature intervention; the sheaf-exclusive block is unresolved at both depths, and the rotation rung carries nothing at two layers, unlike under the feature intervention. Table 27 repeats the two-layer ladder at hidden width 128 with the same seeds. On Minesweeper the gate increment halves and the rotation

increment nearly doubles, so both η_{sheaf} and the absolute sheaf-exclusive gain rise; on Roman-empire both fall slightly. Three float32 embedding checks exceeded the tolerance at width 128 on Minesweeper and passed the float64 adjudication. We did not tune either width, and the main tables use 64.

Table 26: A second intervention on Minesweeper: degree-preserving double-edge swaps that change 22% of the edges and leave degrees, features, labels and splits untouched. Entries split the effect on the dense model exactly as in Eq. (2); paired 95% t intervals over ten official splits (laptop GPU). †: split-paired depth difference excludes zero.

Collected at	Two layers	Four layers	Four minus two
Identity rung	$+0.049\,[+0.045, +0.053]$	$+0.023\,[+0.019, +0.028]$	$-0.026\,[-0.031, -0.021]^{\dagger}$
Gate	$+0.002\,[-0.006, +0.010]$	$+0.039\,[+0.028, +0.050]$	$+0.037\,[+0.028, +0.047]^{\dagger}$
Sign	$+0.017\,[+0.002, +0.032]$	$+0.008\,[-0.002, +0.018]$	$-0.009\,[-0.029, +0.012]$
Rotation	$+0.002\,[-0.015, +0.018]$	$+0.003\,[-0.011, +0.017]$	$+0.001\,[-0.019, +0.022]$
Diagonal	$+0.011\,[+0.001, +0.022]$	$+0.000\,[-0.006, +0.006]$	$-0.011\,[-0.024, +0.002]$
Dense	$+0.001\,[-0.002, +0.004]$	$+0.003\,[-0.004, +0.010]$	$+0.002\,[-0.007, +0.010]$
Scalar rungs	$+0.019\,[+0.002, +0.035]$	$+0.047\,[+0.028, +0.067]$	$+0.029\,[+0.004, +0.054]^{\dagger}$
Sheaf-exclusive rungs	$+0.014\,[-0.002, +0.031]$	$+0.006\,[-0.010, +0.023]$	$-0.008\,[-0.033, +0.017]$
Total	$+0.082\,[+0.077, +0.088]$	$+0.077\,[+0.070, +0.085]$	$-0.005\,[-0.012, +0.002]$

Table 27: Hidden width 128 against 64: two-layer ladder, same ten splits and seeds, laptop GPU. Paired 95% t intervals over splits; †: difference excludes zero.

Dataset	Width	$S(\mathcal{M}_0)$	$S(\mathcal{M}_5)$	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5	Sheaf-exclusive gain	η_{sheaf}
Minesweeper	64	0.795	0.906	+0.047	+0.013	+0.033	+0.016	+0.003	$+0.051\,[+0.033, +0.069]$	46%
	128	0.809	0.905	+0.022	+0.004	+0.060	+0.008	+0.002	$+0.070\,[+0.061, +0.079]$	73%
	128 minus 64								$+0.019\,[+0.003, +0.036]^{\dagger}$	
Roman-empire	64	0.578	0.743	+0.150	−0.002	+0.003	+0.002	+0.012	$+0.018\,[+0.013, +0.022]$	11%
	128	0.588	0.730	+0.128	+0.000	+0.001	+0.005	+0.007	$+0.013\,[+0.008, +0.018]$	9%
	128 minus 64								$-0.005\,[-0.010, -0.000]^{\dagger}$	

What is not established. The protocol cost persists at four layers: the audited dense endpoint is 0.778 on Roman-empire and 0.905 on Minesweeper, against 0.804 and 0.922 for the published reference, so our claims concern attribution at accuracy near, not at, the published reference. Graph value is depth-matched: the no-graph baseline was retrained at four layers (on a laptop GPU; it is an MLP and changes by less than 0.02 with depth). On Roman-empire identity propagation becomes far more harmful with depth, $G_{\mathrm{graph}}(\mathcal{M}_0) = -0.078\,[-0.081, -0.076]$ at two layers and $-0.295\,[-0.297, -0.293]$ at four, so about 85% of the four-layer gate increment (0.295 of 0.347) is repair of a harmful aggregation rule, against about 54% at two layers. Minesweeper remains graph-helpful at both depths ($+0.287$ and $+0.235$), as do Tolokers ($+0.038$, $+0.031$) and Amazon-ratings ($+0.056$, $+0.071$), so $k^{\dagger}$ is unchanged by depth on all four tasks. The property-specific interaction of Section 4.1 was re-measured at four layers on Minesweeper and Roman-empire (Tables 15 and 16; ten split-level units per cell, laptop GPU, original and intervened ladders sharing seeds and hardware; all 200 parent–child embedding checks passed in float32, maximum logit gap 8.1×10^{-5}). The rung carrying the interaction moves with depth on Minesweeper, from rotation to gate, so the two-layer fingerprint of Table 14 should be read as a statement about that depth and not about the rotation family. Finally, one array task aborted on the float32 embedding audit (a 5.7×10^{-4} parent–child logit gap against a 2×10^{-4} tolerance). We did not relax the tolerance; any exceedance is now recomputed in float64 and must fall below 10^{-9} to pass, so that a genuine nesting failure still aborts. Forcing every rung through this path gives gaps of 10^{-16} to 10^{-14}, and none of the 1,300 rung checks recorded since the change exceeded the float32 tolerance.

A.10 A PREREGISTERED SUBSTITUTION TEST: SPECTRAL FREEDOM AGAINST MATRIX TRANSPORT

The scalar rungs of the ladder propagate with one fixed polynomial of their operator, $h \mapsto h - \frac{1}{2}Lh$. Matrix-valued transport is then credited with everything those rungs cannot do, which includes responses they are barred from by that fixed polynomial rather than by their transport family. We

fixed a test of this before running it (the preregistration is in the supplementary material). The free arm replaces the update of every layer by $h \mapsto \sum_{k=0}^{K} \theta_k L^k h$ with $K = 3$ and θ learned per layer and shared by all channels, initialized at $(1, -\frac{1}{2}, 0, 0)$. Three properties make the arms comparable, all checked numerically: the fixed update is a point of the family, so every rung still contains its parent exactly and the warm start, the loss cap and the retention rule are unchanged; the θ are scalars, so the layer is a polynomial in the rung's own operator, with the same eigenvectors and a different eigenvalue response, which is spectral freedom and not transport; and within an arm the scalar and the matrix rungs have the same K, so the receptive field is matched where the contrast is taken. The preregistered quantity is the seed-paired interaction $D = G_{\text{free}} - G_{\text{restricted}}$ of the sheaf-exclusive gain $G = S(\mathcal{M}_5) - S(\mathcal{M}_2)$, with three readings fixed in advance: resolved negative means part of the gain credited to transport is a substitute for spectral freedom, an interval inside ± 0.01 means the substitution is not supported at this resolution, and resolved positive means the opposite of the hypothesis.

Table 28: Preregistered substitution test with its training-budget control. The restricted arm is the released-operator ladder of the main experiments; the free arm is the same ladder with the layer update $\sum_k \theta_k L^k h$, $K = 3$, θ learned per layer and initialized at the fixed update, so the arms share the rungs, the warm starts, the seeds and, within an arm, the receptive field. $G = S(\text{dense}) - S(\text{sign})$ is the sheaf-exclusive gain and D the seed-paired interaction, the quantity fixed in advance. The next two columns repeat both against the continued parent of each arm: $G_{\text{free}}^{\text{cont}}$ is what the free arm's matrix rungs buy beyond continued training, and D^{cont} the interaction computed from controlled increments on both sides. †: interval excludes zero; a row with one seed estimates no variance, so it is given as a point estimate and is an exploratory single run, not a test. The last column gives the identity endpoints, restricted/free.

Dataset	L	n	$G_{\text{restricted}}$	G_{free}	D	$G_{\text{free}}^{\text{cont}}$	D^{cont}	identity
Minesweeper	2	10	$+0.0584\,[+0.0473, +0.0695]^{\dagger}$	$+0.0036\,[+0.0003, +0.0070]^{\dagger}$	$-0.0548\,[-0.0668, -0.0428]^{\dagger}$	$+0.0037\,[+0.0002, +0.0072]^{\dagger}$	$-0.0356\,[-0.0441, -0.0271]^{\dagger}$	0.804/0.896
Minesweeper	4	10	$+0.0930\,[+0.0803, +0.1057]^{\dagger}$	$+0.0076\,[+0.0033, +0.0120]^{\dagger}$	$-0.0854\,[-0.0990, -0.0719]^{\dagger}$	$+0.0050\,[+0.0014, +0.0086]^{\dagger}$	$-0.0782\,[-0.0920, -0.0643]^{\dagger}$	0.745/0.895
Roman-empire	2	10	$+0.0179\,[+0.0132, +0.0225]^{\dagger}$	$+0.0237\,[+0.0193, +0.0282]^{\dagger}$	$+0.0059\,[-0.0016, +0.0133]$	$+0.0153\,[+0.0114, +0.0192]^{\dagger}$	$-0.0013\,[-0.0046, +0.0020]$	0.712/0.714
Roman-empire	4	10	$+0.0577\,[+0.0518, +0.0635]^{\dagger}$	$+0.0743\,[+0.0627, +0.0858]^{\dagger}$	$+0.0166\,[+0.0045, +0.0287]^{\dagger}$	$+0.0463\,[+0.0340, +0.0587]^{\dagger}$	$+0.0084\,[-0.0101, +0.0269]$	0.583/0.712
Tolokers	2	1	$+0.0100$	$+0.0070$	-0.0030	—	—	0.779/0.780

The two datasets answer differently, and both answers are informative. On Minesweeper the substitution reading holds and it is not a small effect: $D = -0.0548\,[-0.0668, -0.0428]$ at two layers and $-0.0854\,[-0.0990, -0.0719]$ at four, so of a sheaf-exclusive gain of $+0.0584$ and $+0.0930$ under the fixed update, 94% and 92% is available to a scalar ladder once its spectral response is free. What remains, $+0.0036$ and $+0.0076$, still survives the parent-continuation control, so it is not nothing; but the headline share on this dataset is mostly a statement about the scalar control's fixed polynomial. On Roman-empire the interaction goes the other way: $D = +0.0059\,[-0.0016, +0.0133]$ at two layers, neither resolved nor inside the practical margin, so the direction is undetermined there, and $+0.0166\,[+0.0045, +0.0287]$ at four, resolved positive before the training-budget control, with the gain growing from $+0.0577$ to $+0.0743$. That interval is $+0.0084\,[-0.0101, +0.0269]$ once both arms are measured against their own continued parent, so no direction is established on this dataset at either depth: it contains zero and it contains positive effects of practical size.

Two caveats limit what the test settles, the first of them preregistered. The identity endpoint moves a great deal in the free arm, from 0.804 to 0.896 on Minesweeper at two layers and from 0.745 to 0.895 at four, while the dense endpoint moves little (0.912 to 0.926): the free arm is a much stronger model at the bottom of the ladder and about the same at the top, which is exactly how a substitution shows up, but it also means the two arms are not the same task for the early rungs. A filter that is not a polynomial in the same operator could still place the boundary elsewhere, but within this family the degree is not what decides it. What the test does establish is that the attribution moves with a factor that has nothing to do with the restriction maps. The uncontrolled point estimates run in opposite directions on the two datasets; once the continuation budget is matched, only the negative interaction on Minesweeper is supported.

How far the substitution goes, and a third dataset. Two of the caveats above are answerable with more of the same experiment, and Table 29 answers them. The degree is swept over $K \in \{2, 3, 5\}$. On Minesweeper the share of the sheaf-exclusive gain that a free spectral response absorbs (point estimates of a ratio) is of comparable size at every degree tested and does not grow materially with K (Minesweeper at two layers ($K = 2$: 95%, $K = 3$: 94%, $K = 5$: 92%); Minesweeper at four layers

($K = 2$: 95%, $K = 3$: 92%, $K = 5$: 87%)), so the substitution is not an artifact of giving the scalar rungs an unboundedly rich filter: a short polynomial suffices. On Roman-empire the interaction stays positive at every degree (Roman-empire at two layers ($K = 2$: -21%, $K = 3$: -33%, $K = 5$: -7%); Roman-empire at four layers ($K = 2$: -29%, $K = 3$: -29%, $K = 5$: -16%), negative shares meaning the gain grew): the uncontrolled point estimates point the other way at every degree, but Table 30 shows none of them survives the continuation control, so no direction is established there. A third dataset has one run only: on Tolokers at two layers the restricted gain is $+0.0100$ and the free arm $+0.0070$, a difference of -0.0030. One observation estimates no variance, so we report the point estimate with no interval and no significance mark, and draw no conclusion from it; it records the direction we saw, nothing more. What does not change with either addition is the conclusion the test was run for: which fraction of a gain is credited to matrix-valued transport moves with the spectral freedom of the scalar control, a factor that involves no restriction map at all, and where it is resolved, on Minesweeper, the movement is a substitution.

Table 29: Degree sweep of the substitution test of Table 28. G at $K = 1$ is the sheaf-exclusive gain under the fixed update, and D at degree K is the seed-paired interaction $G_{\mathrm{free}}(K) - G_{\mathrm{restricted}}$; D at $K = 1$ is zero by construction and is omitted. Seed → official split, †: interval excludes zero. Cells hold ten seeds unless a count is given in parentheses; with one seed no variance can be estimated, so such a cell is a single exploratory run reported as a point estimate with no interval and no significance mark. Tolokers was added after the two datasets on which the test was preregistered, and its one run is exploratory.

Dataset	L	G at K=1	D at K=2	D at K=3	D at K=5
Minesweeper	2	$+0.0584\,[+0.0473, +0.0695]^{\dagger}$	$-0.0556\,[-0.0674, -0.0437]^{\dagger}$	$-0.0548\,[-0.0668, -0.0428]^{\dagger}$	$-0.0535\,[-0.0651, -0.0419]^{\dagger}$
Minesweeper	4	$+0.0930\,[+0.0803, +0.1057]^{\dagger}$	$-0.0882\,[-0.1013, -0.0751]^{\dagger}$	$-0.0854\,[-0.0990, -0.0719]^{\dagger}$	$-0.0808\,[-0.1031, -0.0586]^{\dagger}$ (5)
Roman-empire	2	$+0.0179\,[+0.0132, +0.0225]^{\dagger}$	$+0.0037\,[-0.0017, +0.0090]$	$+0.0059\,[-0.0016, +0.0133]$	$+0.0013\,[-0.0037, +0.0063]$
Roman-empire	4	$+0.0577\,[+0.0518, +0.0635]^{\dagger}$	$+0.0170\,[+0.0083, +0.0257]^{\dagger}$	$+0.0166\,[+0.0045, +0.0287]^{\dagger}$	$+0.0092\,[-0.0115, +0.0298]$ (5)
Tolokers	2	$+0.0100$ (1)	—	-0.0030 (1)	—

Which freedom is doing the work. A degree-K update with learned coefficients changes two things at once: a layer reaches K hops instead of one, and its spectral response is free. Table 30 separates them with controls run after the preregistered comparison, against the same restricted arm. Coefficient freedom alone, at the restricted arm's one-hop reach, produces a reduction of comparable magnitude on Minesweeper (the point estimates are close; we have not run an equivalence test between the one-hop and the three-hop interaction): $D = -0.0911\,[-0.1033, -0.0789]$ at four layers against $-0.0854\,[-0.0990, -0.0719]$ for three hops with learned coefficients, and $-0.0540\,[-0.0653, -0.0428]$ against $-0.0548\,[-0.0668, -0.0428]$ at two; both survive the continuation control ($-0.0822\,[-0.0936, -0.0707]$ and $-0.0349\,[-0.0436, -0.0262]$ at one hop). An expanded receptive field is therefore not necessary for the substitution, which is what we claim.

The converse — whether width alone would suffice — is not answered by the row marked *, and we flag it because the label invites the wrong reading. Freezing θ at the fixed update means freezing it at $(1, -\frac{1}{2}, 0, 0)$, and $\sum_k \theta_k L^k h$ is then $h - \frac{1}{2}Lh$ whatever the nominal degree: that arm computes exactly what the restricted arm computes, which `test_spectral_poly.py` checks by equality of outputs. Its interval, $-0.0013\,[-0.0120, +0.0095]$ on Minesweeper at four layers and $+0.0014\,[-0.0021, +0.0049]$ on Roman-empire at two, is the null control of the comparison: it detected no significant difference, which is what it should do, but an interval of that width is not a calibration guarantee for the pipeline, and it is not evidence about reach. A reach-only arm needs a fixed filter of genuine width: θ frozen at the expansion of $(I - \frac{1}{2}L)^3$, which propagates three hops with no free spectral parameter. Its response is that of three fixed updates composed rather than of one, so it does not hold the response fixed either; no arm in this family can widen a layer and leave the response untouched. We evaluated it in all four cells, ten seeds each. On Minesweeper it produces a substitution at both depths: $D = -0.0425\,[-0.0551, -0.0299]$ at two layers and $-0.0658\,[-0.0786, -0.0529]$ at four, $-0.0263\,[-0.0376, -0.0150]$ and $-0.0678\,[-0.0827, -0.0529]$ against the continued parents, 73% and 71% of the restricted gain; and a resolvedly smaller one than coefficient freedom at one hop on the same seeds (seed-paired differences $+0.0115\,[+0.0042, +0.0188]$ and $+0.0253\,[+0.0196, +0.0311]$). It gets there by a different route: the one-hop learned arm lifts the bottom of the ladder (identity $0.745 \to 0.895$ at four layers), whereas the fixed three-hop filter lowers it ($0.745 \to 0.695$) and instead lifts the scalar rungs' top, sign rising from 0.819 to 0.896 against

a dense rung at 0.923. On Roman-empire the arm resolves nothing ($+0.0032\,[-0.0023, +0.0088]$ and $+0.0055\,[-0.0038, +0.0148]$; $+0.0001$ and $+0.0056$ under the control, unresolved), and its endpoints say why the fixed filter is no help there: the identity rung collapses ($0.712 \to 0.405$ at two layers, $0.583 \to 0.287$ at four) while sign and dense stay where they were, so three fixed hops damage the bottom of that ladder without moving its top. We make no claim from the null.

On Roman-empire no arm produces an interaction that survives the training-budget control: one hop with learned coefficients gives $D = +0.0042\,[-0.0025, +0.0109]$ at four layers and $-0.0039\,[-0.0104, +0.0026]$ at two, three hops with learned coefficients $+0.0166\,[+0.0045, +0.0287]$ at four and $+0.0059\,[-0.0016, +0.0133]$ at two, and the first of those becomes $+0.0084\,[-0.0101, +0.0269]$ under the control. We therefore do not read Roman-empire as a case of complementarity, and we do not read it as an absence of effect either: no interaction was resolved after matching the continuation budget, in either direction. Taken together with the degree sweep, the picture is that a scalar control's fixed polynomial can hide a large part of what is later credited to matrix-valued transport, that either of the two things a higher degree changes — the coefficients or the reach — uncovers part of it, the coefficients more, and that whether it happens at all depends on the task.

Table 30: Which freedom produces the substitution. A degree-K update with learned coefficients changes two things at once: a layer reaches K hops instead of one, and its spectral response is free. The rows separate them against the same restricted arm. Degree 1 with learned coefficients has the reach of the restricted arm and a free response. Freezing θ at the expansion of $(I - \frac{1}{2}L)^3$ gives the converse: three hops with no free spectral parameter, the response of three fixed updates composed. * marks a null control rather than a reach control: freezing θ at the fixed update $(1, -\frac{1}{2}, 0, 0)$ leaves $\sum_k \theta_k L^k h = h - \frac{1}{2}Lh$, so that arm is the restricted model itself and its row measures the seed noise of the comparison, not an effect of reach. D is the seed-paired interaction of the sheaf-exclusive gain and D^{cont} the same after both arms are measured against their own continued parent; †: interval excludes zero. n is the number of seeds behind each row.

Dataset	L	hops	coefficients	n	D	D^{cont}
Minesweeper	2	1	learned	10	$-0.0540\,[-0.0653, -0.0428]^\dagger$	$-0.0349\,[-0.0436, -0.0262]^\dagger$
		3	fixed, $(I - \frac{1}{2}L)^3$	10	$-0.0425\,[-0.0551, -0.0299]^\dagger$	$-0.0263\,[-0.0376, -0.0150]^\dagger$
		3	learned	10	$-0.0548\,[-0.0668, -0.0428]^\dagger$	$-0.0356\,[-0.0441, -0.0271]^\dagger$
Minesweeper	4	1	learned	10	$-0.0911\,[-0.1033, -0.0789]^\dagger$	$-0.0822\,[-0.0936, -0.0707]^\dagger$
		3	fixed, $(I - \frac{1}{2}L)^3$	10	$-0.0658\,[-0.0786, -0.0529]^\dagger$	$-0.0678\,[-0.0827, -0.0529]^\dagger$
		3	learned	10	$-0.0854\,[-0.0990, -0.0719]^\dagger$	$-0.0782\,[-0.0920, -0.0643]^\dagger$
		1^*	fixed (zero tail)	10	$-0.0013\,[-0.0120, +0.0095]$	$-0.0175\,[-0.0391, +0.0041]$
Roman-empire	2	1	learned	10	$-0.0039\,[-0.0104, +0.0026]$	$-0.0043\,[-0.0123, +0.0038]$
		3	fixed, $(I - \frac{1}{2}L)^3$	10	$+0.0032\,[-0.0023, +0.0088]$	$+0.0001\,[-0.0063, +0.0065]$
		3	learned	10	$+0.0059\,[-0.0016, +0.0133]$	$-0.0013\,[-0.0046, +0.0020]$
		1^*	fixed (zero tail)	10	$+0.0014\,[-0.0021, +0.0049]$	$-0.0009\,[-0.0067, +0.0050]$
Roman-empire	4	1	learned	10	$+0.0042\,[-0.0025, +0.0109]$	$+0.0040\,[-0.0068, +0.0148]$
		3	fixed, $(I - \frac{1}{2}L)^3$	10	$+0.0055\,[-0.0038, +0.0148]$	$+0.0056\,[-0.0101, +0.0214]$
		3	learned	10	$+0.0166\,[+0.0045, +0.0287]^\dagger$	$+0.0084\,[-0.0101, +0.0269]$
		1^*	fixed (zero tail)	2	$+0.0100\,[-0.0427, +0.0627]$	$-0.0161\,[-0.2482, +0.2159]$

The budget identity. With $\Psi_{j,k}$ as in Section 5,

$$S(\mathcal{M}_5; \mathcal{D}) - S(\mathcal{M}_5; \mathcal{D}_{-j}) \;=\; \underbrace{S(\mathcal{M}_0; \mathcal{D}) - S(\mathcal{M}_0; \mathcal{D}_{-j})}_{\Psi_{j,0}} \;+\; \sum_{k=1}^{5} \Psi_{j,k}, \tag{2}$$

A.11 Parent-continuation control

Nesting and warm starts make every child start from the function its parent computes, but every rung also comes with more training: the child is optimized again, with a fresh optimizer, from the parent's checkpoint. Continued training alone can move a score (Table 5), so an increment Γ_k measured against the fitted parent mixes the two. The control is to train the parent again from the same checkpoint with no new parameters, under the same budget, optimizer reset, training-loss cap, random stream and validation-selection rule as the child, and to measure the child against

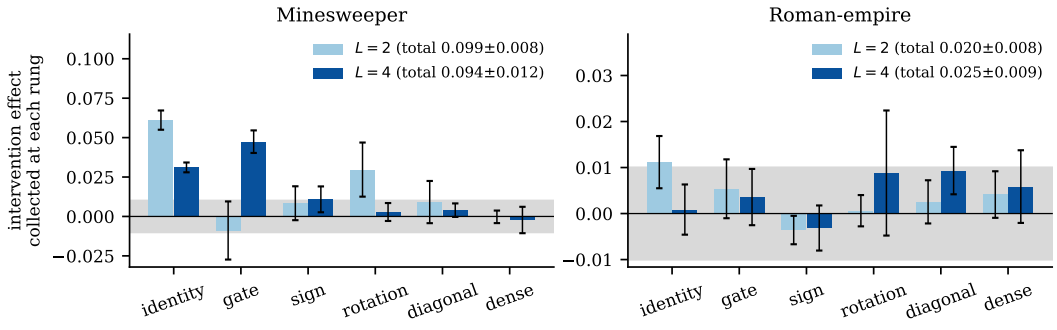


Figure 7: Where the effect of the intervention is collected. Each bar is one term of Eq. (2); whiskers are paired 95% t intervals over ten official splits and the gray band is the practical margin ± 0.01. The total barely moves with depth; the rung that collects it does (our operator; the released operator is in Table 25).

that continued parent (`run_continuation.py`, released operator and ours, ten seeds per cell).

The control costs part of every increment, and not the same part everywhere (Minesweeper at two layers under the ours operator, $+0.063$ becomes $+0.045\,[+0.026, +0.064]$; Minesweeper at four layers under the ours operator, $+0.019$ becomes $+0.010\,[+0.004, +0.017]$; Minesweeper at two layers under the released operator, $+0.055$ becomes $+0.039\,[+0.029, +0.049]$; Minesweeper at four layers under the released operator, $+0.101$ becomes $+0.083\,[+0.071, +0.096]$; Roman-empire at two layers under the ours operator, $+0.016$ becomes $+0.015\,[+0.012, +0.019]$; Roman-empire at four layers under the ours operator, $+0.071$ becomes $+0.048\,[+0.042, +0.055]$; Roman-empire at two layers under the released operator, $+0.020$ becomes $+0.017\,[+0.013, +0.020]$; Roman-empire at four layers under the released operator, $+0.060$ becomes $+0.038\,[+0.026, +0.050]$). The sheaf-exclusive gain survives it in every cell with the full ten seeds: the interval of the increment over the continued parent excludes zero throughout. Which rungs keep their increment varies by cell; the table gives each one. This is the estimand the ladder actually measures, and we report it as such: Γ_k is the increment over the fitted parent, the column "over continued" is the increment over the parent given the child's training budget, and the difference between them is the part of an increment that continued training alone would have produced. This is a paired decomposition of each increment, not a bound: the increment over the continued parent is itself negative in places, so the fitted parent's score is not a floor for what continued training reaches on the test split. Apart from the substitution test of Appendix A.10, whose arms are both recomputed against it, nothing else in the paper is recomputed against this control, and the shares of Section 5 are shares of Γ_k: the allocation of the gain under this training protocol, not a contribution with training removed. Nor should the corrected increments be summed and read as a decomposition of the identity-to-dense gain, since each is measured against its own continued parent, so the telescoping identity does not apply to them.

The depth reversal under the control. The most informative contrast the control can settle is whether the operators still disagree about depth once the training budget is matched. Table 32 takes the seed-paired difference of the sheaf-exclusive gain between four and two layers under each operator, from the increments over the fitted parent and from those over the continued parent. On Minesweeper the disagreement survives: the depth difference is $-0.034\,[-0.056, -0.012]$ under our operator and $+0.044\,[0.025, 0.063]$ under the released one, both resolved and of opposite sign, against -0.045 and $+0.046$ before the control. On Roman-empire the two operators agree before and after, with the gain rising with depth in both ($+0.033$ and $+0.021$ after the control). The reversal is therefore not an artifact of the training that comes with a deeper ladder.

B ARCHITECTURE TRANSFER: FULL ACCOUNT

Sections 3–A establish their claims on one two-layer normalized sheaf Laplacian. The natural objection is that both the normalization equivalence and the optimization sensitivity are artifacts of that architecture. We therefore repeated the relevant audits on a published directed sheaf hypergraph

Table 31: Parent-continuation control. At every transition the fitted parent is expanded into the child and trained ("ladder", the increment Γ_k of the main experiments) and, from the same checkpoint, trained again with no new parameters under the same budget, optimizer reset, training-loss cap, random stream and validation-selection rule ("continued"). The last column is the child's increment over the continued parent: what the new parameters bought beyond continued training. Seeds are units, seed $\rightarrow$ official split, 95% t intervals, † excluding zero.

Dataset	operator	L	n	rung	ladder Γ_k	continued parent	over continued
Minesweeper	ours	2	10	gate	$+0.0318\,[+0.0254, +0.0383]^\dagger$	$+0.0116\,[+0.0056, +0.0175]^\dagger$	$+0.0202\,[+0.0130, +0.0275]^\dagger$
Minesweeper	ours	2	10	sign	$+0.0118\,[+0.0026, +0.0209]^\dagger$	$+0.0031\,[-0.0025, +0.0086]$	$+0.0087\,[+0.0042, +0.0132]^\dagger$
Minesweeper	ours	2	10	rotation	$+0.0425\,[+0.0241, +0.0608]^\dagger$	$+0.0062\,[-0.0000, +0.0123]$	$+0.0363\,[+0.0166, +0.0559]^\dagger$
Minesweeper	ours	2	10	diagonal	$+0.0192\,[+0.0084, +0.0300]^\dagger$	$+0.0113\,[+0.0019, +0.0207]^\dagger$	$+0.0079\,[+0.0016, +0.0142]^\dagger$
Minesweeper	ours	2	10	dense	$+0.0014\,[-0.0003, +0.0030]$	$+0.0008\,[-0.0006, +0.0021]$	$+0.0006\,[-0.0008, +0.0019]$
Minesweeper	ours	2	10	sheaf-exclusive	$+0.0630\,[+0.0488, +0.0772]^\dagger$	$+0.0183\,[+0.0074, +0.0292]^\dagger$	$+0.0447\,[+0.0259, +0.0636]^\dagger$
Minesweeper	ours	4	10	gate	$+0.1283\,[+0.1180, +0.1386]^\dagger$	$+0.0026\,[+0.0007, +0.0045]^\dagger$	$+0.1257\,[+0.1147, +0.1367]^\dagger$
Minesweeper	ours	4	10	sign	$+0.0186\,[+0.0105, +0.0266]^\dagger$	$+0.0026\,[+0.0001, +0.0051]^\dagger$	$+0.0160\,[+0.0098, +0.0221]^\dagger$
Minesweeper	ours	4	10	rotation	$+0.0090\,[+0.0036, +0.0144]^\dagger$	$+0.0010\,[-0.0013, +0.0032]$	$+0.0080\,[+0.0029, +0.0131]^\dagger$
Minesweeper	ours	4	10	diagonal	$+0.0019\,[-0.0007, +0.0045]$	$+0.0019\,[-0.0005, +0.0043]$	$-0.0000\,[-0.0032, +0.0032]$
Minesweeper	ours	4	10	dense	$+0.0077\,[+0.0011, +0.0142]^\dagger$	$+0.0053\,[-0.0005, +0.0110]$	$+0.0024\,[-0.0023, +0.0071]$
Minesweeper	ours	4	10	sheaf-exclusive	$+0.0185\,[+0.0124, +0.0246]^\dagger$	$+0.0081\,[+0.0022, +0.0141]^\dagger$	$+0.0104\,[+0.0042, +0.0166]^\dagger$
Minesweeper	released	2	10	gate	$+0.0370\,[+0.0286, +0.0453]^\dagger$	$+0.0169\,[+0.0099, +0.0239]^\dagger$	$+0.0201\,[+0.0119, +0.0282]^\dagger$
Minesweeper	released	2	10	sign	$+0.0042\,[-0.0004, +0.0088]$	$+0.0063\,[+0.0025, +0.0101]^\dagger$	$-0.0021\,[-0.0063, +0.0022]$
Minesweeper	released	2	10	rotation	$+0.0250\,[+0.0178, +0.0321]^\dagger$	$+0.0022\,[-0.0006, +0.0050]$	$+0.0228\,[+0.0149, +0.0307]^\dagger$
Minesweeper	released	2	10	diagonal	$+0.0143\,[+0.0095, +0.0191]^\dagger$	$+0.0088\,[+0.0025, +0.0151]^\dagger$	$+0.0055\,[+0.0009, +0.0102]^\dagger$
Minesweeper	released	2	10	dense	$+0.0160\,[+0.0077, +0.0243]^\dagger$	$+0.0050\,[+0.0001, +0.0099]^\dagger$	$+0.0110\,[+0.0034, +0.0186]^\dagger$
Minesweeper	released	2	10	sheaf-exclusive	$+0.0553\,[+0.0478, +0.0628]^\dagger$	$+0.0160\,[+0.0067, +0.0253]^\dagger$	$+0.0393\,[+0.0293, +0.0493]^\dagger$
Minesweeper	released	4	10	gate	$+0.0670\,[+0.0603, +0.0737]^\dagger$	$+0.0028\,[+0.0008, +0.0049]^\dagger$	$+0.0642\,[+0.0563, +0.0720]^\dagger$
Minesweeper	released	4	10	sign	$+0.0011\,[-0.0004, +0.0027]$	$+0.0033\,[+0.0009, +0.0058]^\dagger$	$-0.0022\,[-0.0047, +0.0003]$
Minesweeper	released	4	10	rotation	$+0.0777\,[+0.0654, +0.0899]^\dagger$	$+0.0006\,[-0.0006, +0.0017]$	$+0.0771\,[+0.0650, +0.0891]^\dagger$
Minesweeper	released	4	10	diagonal	$+0.0135\,[+0.0063, +0.0206]^\dagger$	$+0.0116\,[+0.0049, +0.0182]^\dagger$	$+0.0019\,[+0.0000, +0.0038]^\dagger$
Minesweeper	released	4	10	dense	$+0.0100\,[+0.0053, +0.0148]^\dagger$	$+0.0058\,[+0.0029, +0.0088]^\dagger$	$+0.0042\,[+0.0010, +0.0074]^\dagger$
Minesweeper	released	4	10	sheaf-exclusive	$+0.1012\,[+0.0909, +0.1114]^\dagger$	$+0.0180\,[+0.0093, +0.0267]^\dagger$	$+0.0832\,[+0.0705, +0.0958]^\dagger$
Roman-empire	ours	2	10	gate	$+0.1435\,[+0.1385, +0.1486]^\dagger$	$-0.0001\,[-0.0019, +0.0017]$	$+0.1436\,[+0.1383, +0.1489]^\dagger$
Roman-empire	ours	2	10	sign	$+0.0011\,[-0.0020, +0.0041]$	$+0.0007\,[-0.0027, +0.0041]$	$+0.0004\,[-0.0027, +0.0034]$
Roman-empire	ours	2	10	rotation	$+0.0026\,[-0.0006, +0.0058]$	$-0.0001\,[-0.0004, +0.0002]$	$+0.0027\,[-0.0007, +0.0061]$
Roman-empire	ours	2	10	diagonal	$+0.0032\,[+0.0004, +0.0060]^\dagger$	$+0.0001\,[-0.0005, +0.0007]$	$+0.0031\,[+0.0004, +0.0058]^\dagger$
Roman-empire	ours	2	10	dense	$+0.0104\,[+0.0062, +0.0146]^\dagger$	$+0.0009\,[-0.0011, +0.0029]$	$+0.0095\,[+0.0063, +0.0127]^\dagger$
Roman-empire	ours	2	10	sheaf-exclusive	$+0.0162\,[+0.0119, +0.0205]^\dagger$	$+0.0009\,[-0.0012, +0.0030]$	$+0.0153\,[+0.0116, +0.0190]^\dagger$
Roman-empire	ours	4	10	gate	$+0.3503\,[+0.3455, +0.3551]^\dagger$	$+0.0028\,[+0.0006, +0.0050]^\dagger$	$+0.3475\,[+0.3430, +0.3519]^\dagger$
Roman-empire	ours	4	10	sign	$+0.0011\,[-0.0012, +0.0034]$	$+0.0002\,[-0.0004, +0.0009]$	$+0.0008\,[-0.0016, +0.0033]$
Roman-empire	ours	4	10	rotation	$+0.0177\,[+0.0096, +0.0258]^\dagger$	$+0.0004\,[-0.0016, +0.0023]$	$+0.0173\,[+0.0093, +0.0253]^\dagger$
Roman-empire	ours	4	10	diagonal	$+0.0263\,[+0.0221, +0.0305]^\dagger$	$+0.0112\,[+0.0052, +0.0173]^\dagger$	$+0.0151\,[+0.0082, +0.0219]^\dagger$
Roman-empire	ours	4	10	dense	$+0.0275\,[+0.0214, +0.0337]^\dagger$	$+0.0115\,[+0.0062, +0.0167]^\dagger$	$+0.0161\,[+0.0119, +0.0202]^\dagger$
Roman-empire	ours	4	10	sheaf-exclusive	$+0.0715\,[+0.0670, +0.0759]^\dagger$	$+0.0230\,[+0.0173, +0.0288]^\dagger$	$+0.0484\,[+0.0417, +0.0551]^\dagger$
Roman-empire	released	2	10	gate	$+0.0305\,[+0.0256, +0.0353]^\dagger$	$-0.0011\,[-0.0028, +0.0005]$	$+0.0316\,[+0.0263, +0.0369]^\dagger$
Roman-empire	released	2	10	sign	$+0.0003\,[-0.0013, +0.0019]$	$-0.0002\,[-0.0013, +0.0009]$	$+0.0005\,[-0.0014, +0.0023]$
Roman-empire	released	2	10	rotation	$+0.0020\,[+0.0002, +0.0038]^\dagger$	$+0.0001\,[-0.0009, +0.0011]$	$+0.0019\,[-0.0000, +0.0037]$
Roman-empire	released	2	10	diagonal	$+0.0025\,[+0.0005, +0.0045]^\dagger$	$+0.0014\,[-0.0013, +0.0040]$	$+0.0011\,[-0.0024, +0.0047]$
Roman-empire	released	2	10	dense	$+0.0150\,[+0.0122, +0.0179]^\dagger$	$+0.0015\,[-0.0006, +0.0036]$	$+0.0136\,[+0.0099, +0.0172]^\dagger$
Roman-empire	released	2	10	sheaf-exclusive	$+0.0195\,[+0.0174, +0.0216]^\dagger$	$+0.0030\,[-0.0007, +0.0066]$	$+0.0166\,[+0.0132, +0.0199]^\dagger$
Roman-empire	released	4	10	gate	$+0.1431\,[+0.1377, +0.1485]^\dagger$	$+0.0018\,[+0.0001, +0.0035]^\dagger$	$+0.1413\,[+0.1348, +0.1478]^\dagger$
Roman-empire	released	4	10	sign	$+0.0031\,[-0.0006, +0.0067]$	$+0.0002\,[-0.0003, +0.0007]$	$+0.0028\,[-0.0005, +0.0061]$
Roman-empire	released	4	10	rotation	$+0.0149\,[+0.0046, +0.0252]^\dagger$	$+0.0000\,[+0.0000, +0.0000]$	$+0.0149\,[+0.0046, +0.0252]^\dagger$
Roman-empire	released	4	10	diagonal	$+0.0191\,[+0.0145, +0.0237]^\dagger$	$+0.0123\,[+0.0065, +0.0180]^\dagger$	$+0.0068\,[+0.0020, +0.0116]^\dagger$
Roman-empire	released	4	10	dense	$+0.0258\,[+0.0217, +0.0299]^\dagger$	$+0.0096\,[+0.0053, +0.0139]^\dagger$	$+0.0162\,[+0.0141, +0.0183]^\dagger$
Roman-empire	released	4	10	sheaf-exclusive	$+0.0598\,[+0.0508, +0.0689]^\dagger$	$+0.0219\,[+0.0160, +0.0278]^\dagger$	$+0.0379\,[+0.0264, +0.0495]^\dagger$

Table 32: Does the depth reversal survive the parent-continuation control? Each entry is the seed-paired difference of the sheaf-exclusive gain between four and two layers, under the two operators, computed from the increments over the fitted parent ("ladder") and from the increments over the continued parent. Ten seeds per cell, seed $\rightarrow$ official split; †: interval excludes zero.

Dataset	Increment measured against	Our operator	Released operator
Minesweeper	ladder Γ	$-0.0445\,[-0.0611, -0.0279]^\dagger$	$+0.0459\,[+0.0350, +0.0567]^\dagger$
Minesweeper	over continued parent	$-0.0344\,[-0.0563, -0.0124]^\dagger$	$+0.0439\,[+0.0245, +0.0633]^\dagger$
Roman-empire	ladder Γ	$+0.0553\,[+0.0478, +0.0627]^\dagger$	$+0.0403\,[+0.0307, +0.0499]^\dagger$
Roman-empire	over continued parent	$+0.0332\,[+0.0271, +0.0392]^\dagger$	$+0.0214\,[+0.0095, +0.0332]^\dagger$

model, using the authors' released implementation and their own best configurations, changing as little as one flag per comparison.

B.1 Learned restriction maps converge to near-constant transport

Proposition 1 says a layerwise constant restriction scalar is absorbed by normalization into an effective propagation strength. It does not say that a *learned* map will go there. Whether it does is an empirical question, and on this architecture it does.

We ran a seven-arm comparison crossing the hyperedge context aggregation (sum, mean, merged-mean, role-mean) with the restriction parameterization (the upstream $\sigma(\cdot)$ map, and an identity-centered $1 + \frac{1}{2}\tanh(g)$ map that contains $F = I$ exactly), plus a trivial $F = I$ control: ten seeds per arm, 70 runs in total, one A100 each, the upstream early-stopping rule, all tasks completed, per-seed data, split and shared-layer initialization fingerprints verified, and machine validators passing on all ten formal summaries. The evaluation quantity is test accuracy at the best validation epoch, not the best test accuracy seen during training.

Table 33: Seven-arm restriction-map comparison on the published directed sheaf hypergraph model (ten seeds per arm, one A100 each). Test accuracy at the best validation epoch; "collapse" counts runs whose best validation accuracy is below 50%.

Context	Map parameterization	Test acc. (%)	Collapse
sum (upstream)	$\sigma(\cdot)$	64.29 ± 23.86	3/10
mean	$\sigma(\cdot)$	70.65 ± 22.13	2/10
merged-mean	$\sigma(\cdot)$	64.29 ± 26.34	3/10
role-mean	$\sigma(\cdot)$	75.14 ± 16.01	1/10
merged-mean	$1 + \frac{1}{2}\tanh(\cdot)$	$\mathbf{80.20 \pm 2.53}$	0/10
role-mean	$1 + \frac{1}{2}\tanh(\cdot)$	79.27 ± 2.88	0/10
—	trivial $F = I$ (control)	79.35 ± 3.33	0/10

Two results follow. First, the identity-centered parameterization removes the failure mode entirely ($0/10$ collapse against $3/10$ upstream), confirming that the instability is a property of the map's parameterization rather than of learned transport as such. Second, and more importantly, once that pathology is removed the best learned arm exceeds the trivial $F = I$ control by only $+0.85$ points, paired 95% CI $[-0.71, +2.41]$, exact sign-flip $p = 0.309$, family-wise Holm $p = 1.000$; role separation is -0.93 points ($p = 0.266$, Holm 1.000). No improvement over identity transport can be claimed.

The mechanism diagnosis is what connects this to Proposition 1. At each seed's best validation epoch the fitted restriction values are almost elementwise constant: $10/10$ role-mean seeds sit at 1.5, and $9/10$ merged-mean seeds at 1.5 with the remaining one at 0.5; the $\sigma(\cdot)$ arms saturate to all-ones on successful trajectories and all-zeros on failed ones. The stabilized arms therefore converge to a globally constant transport with a positive lower bound, retaining no meaningful incidence-level variation — and for stalk dimension one under a normalized Laplacian such a common scale is exactly the quantity Proposition 1 shows normalization absorbs.

We state the epistemic status precisely. These runs are ours, not a third-party replication; what is independent of this paper is the architecture, its released code, and its published configurations. The finding is that an audit on an independently published implementation *is consistent with* the prediction of Proposition 1, not that an outside group confirmed it.

B.2 Optimization can reverse the sign of a capacity effect

Section A showed that non-nested optimization moves a measured capacity effect by up to 0.043 in either direction. On this architecture the same mechanism does something stronger: it changes the effect's *sign*.

We crossed the trivial-versus-learned restriction switch with the magnetic phase in a 2×2 design on a ten-class directed benchmark, ten runs per cell. The configurations differ from the published

best configuration in exactly two tokens; at stalk dimension one the special-head flag *is* the trivial-versus-learned switch, so no upstream code was modified and both arms carry equal stalk dimension and equal parameter count.

Table 34: Trivial-versus-learned restriction maps crossed with the magnetic phase on a ten-class directed benchmark (ten runs per cell, test accuracy in %). The lower block splits each arm by whether the run collapsed (best validation accuracy below 50%).

	$q = 0$	$q = 0.25$
trivial $F \equiv I$	80.08 ± 2.68	78.18 ± 2.08
learned F	50.49 ± 25.18	66.56 ± 18.03

Per-run breakdown: collapsed runs / mean of healthy runs / mean of collapsed runs			
trivial, $q = 0$	$0/10$	80.16	—
learned, $q = 0$	$6/10$	81.78	29.96
learned, $q = 0.25$	$3/10$	78.20	30.50

The standard deviations of 25.18 and 18.03 are themselves the alarm. Per run the learned arms are *bimodal* rather than uniformly shifted: six of ten runs collapse to roughly 30% on a ten-class task (chance is 10%, so this is partial rather than total collapse) while the remainder reach 80–83%. Consequently

$$\underbrace{\Delta_{\text{sheaf}} = -29.59}_{\text{unconditional}} \qquad \text{versus} \qquad \underbrace{81.78 > 80.16}_{\text{conditional on not collapsing}} \quad .$$

Read literally, the unconditional contrast says learned transport is catastrophic and the conditional contrast says it helps. Both are computed from the same 20 runs.

Neither reading is correct on its own, and Section B.1 settles which is closer. A trajectory diagnosis shows the $\sigma(\cdot)$ parameterization drives the sheaf into an all-zero or all-one attractor at the very start of training, which is why the failure is bimodal rather than gradual and why an identity-containing parameterization removes it. When that parameterization is adopted and the collapse disappears (Section B.1), the conditional advantage does not survive ($+0.85$, Holm $p = 1.000$). The survivor-conditional $+1.62$ was therefore a selection effect, exactly as conditioning on an outcome should lead one to suspect.

One comparison in this design needs no conditioning at all, because both arms collapse zero times out of ten: on trivial transport, switching the magnetic phase *off* gains $+1.90$ points, with validation agreeing in direction. We report it as a bounded single-dataset observation.

The methodological consequence is sharper than Standard 2 as stated in Section 4. Optimization protocol does not merely scale a measured capacity effect; it can determine whether that effect is reported as -29.59 or as positive. Any capacity comparison that does not report its collapse rate, and does not state whether it conditions on convergence, is uninterpretable at this magnitude.

B.3 Published family labels are not a chain

Both preceding subsections concern learned maps. The third transfer result concerns how published work *labels* capacity. The released implementation of this model family ships three restriction-map families — diagonal, orthogonal, and general — but the published best configurations use only the diagonal one. That triple is precisely the list Standard 3 argues is not a chain: diagonal and orthogonal are incomparable branches, and both are contained in general. The same endpoint is therefore reachable from two incomparable baselines, so the credit assigned to "the sheaf part" depends on which branch is called the baseline.

We take the authors' own best configuration for each of ten real datasets and change exactly one token, the family. Everything else — stalk dimension, depth, width, dropout, learning rate, magnetic charge, residual, activation, and the number of runs — is held at its published value; we verified token-by-token that no other flag differs in any of the ten configurations. This is a matched-capacity family comparison, not an each-family-at-its-own-optimum leaderboard: the latter confounds family with tuning effort, which is what Standard 3 objects to in the first place.

Table 35: Restriction-map family comparison on the published directed sheaf hypergraph model. Each row uses the authors' best configuration for that dataset with exactly one token changed (the family); test accuracy (%), mean ± std over ten runs. The last two columns are the increment to the common endpoint (general) from each of the two incomparable branches. "n/a": the published configuration has stalk dimension one, where no orthogonal family exists; "fail": cuSPARSE ran out of resources under the matched configuration.

Dataset	Diag	Ortho	General	Gen.−Diag	Gen.−Ortho
Telegram	88.08 ± 1.88	80.19 ± 4.47	86.92 ± 5.07	-1.16	$+6.73$
Cora	87.50 ± 1.23	84.33 ± 0.75	86.41 ± 1.09	-1.09	$+2.08$
Citeseer	77.23 ± 1.28	74.52 ± 1.54	74.57 ± 1.22	-2.66	$+0.05$
Chameleon	44.76 ± 3.52	43.90 ± 3.05	43.48 ± 4.30	-1.28	-0.42
Cornell	77.57 ± 3.21	75.95 ± 2.82	77.03 ± 2.77	-0.54	$+1.08$
Texas	80.54 ± 5.90	79.73 ± 4.05	79.19 ± 7.26	-1.35	-0.54
Wisconsin	85.29 ± 5.21	86.08 ± 3.45	86.86 ± 2.64	$+1.57$	$+0.78$
Email-Enron (d=1)	72.16 ± 4.20	n/a	66.22 ± 5.95	-5.94	—
Email-EU (d=1)	76.48 ± 2.94	n/a	59.07 ± 23.50	-17.41	—
Squirrel	42.80 ± 2.31	fail	fail	—	—

Table 35 shows the path dependence directly. On Telegram the increment to the common endpoint is -1.16 points from the diagonal branch and $+6.73$ from the orthogonal branch; on Cora it is -1.09 against $+2.08$, and on Citeseer -2.66 against $+0.05$. Whether "upgrading to general maps helps" therefore has no single answer on these datasets: its sign depends on which of two incomparable branches is called the baseline. The three WebKB datasets have standard deviations of 3–7 points and resolve nothing. These are differences of ten-run means read from the released training script's summaries; we did not have per-run pairing for this comparison and report them descriptively.

Two further observations. At stalk dimension one the diagonal and general families are nominally the same scalar class, yet the general arm is 5.9 and 17.4 points lower on the two $d = 1$ datasets, with a standard deviation of 23.5 on Email-EU — the bimodal collapse signature of Section B.2, reappearing under a different flag. We have not verified that the two families' $d = 1$ parameterizations are identical, so we report this as nominal rather than exact equivalence. And the orthogonal family does not exist at $d = 1$ at all: a family label can name an empty set under a published configuration.

Limitation. Only the diagonal family was tuned by the original authors; orthogonal and general inherit its hyperparameters. This biases *against* them, so the diagonal family beating the general one on six of seven datasets is not evidence that more capacity hurts. The comparison that is free of this confound is general against orthogonal, both untuned, and there the general family is at least as good on five of seven datasets.

The same pattern in a second released code base. Table 36 repeats the family comparison in the released neural-sheaf-diffusion code (Bodnar et al., 2022), again changing only the family flag. The repository ships a hyperparameter sweep definition but no selected configurations, so an each-family-at-its-own-optimum comparison cannot be reproduced from the release; we therefore use one fixed, untuned configuration for all three families, and the accuracies are below the published ones, far below on the homophilous citation graphs, so only within-row comparisons are meaningful. Across nine datasets the increment to the general family from the diagonal branch is never positive: it is negative on eight and zero on Cornell, and resolved on four (Texas -4.87, Wisconsin -2.74, Film -0.61, Citeseer -5.31 points). From the bundle branch it is positive on six of nine and resolved on two (Citeseer $+5.40$, Pubmed $+0.70$), and negative, never resolved, on the other three. As for the hypergraph model, whether "more general maps help" depends on which incomparable branch is taken as the baseline.

B.4 Scope of Proposition 4: the collapse is implementation-contingent

Proposition 4 is a statement about a particular way of assembling an undirected edge from two directed incidences. We checked whether it applies to the reference orthogonal sheaf model in the literature, using that model's own released orthogonal transform rather than a reimplementation.

Table 36: Restriction-map families in the released neural-sheaf-diffusion code. Within a dataset only the family flag changes (stalk dimension 2, two layers, hidden width 20, 200 epochs, the ten standard folds; one fixed configuration, not tuned, so accuracies are below the published ones). Test accuracy, mean $\pm$ standard deviation over folds; the last two columns are fold-paired increments to the general family with 95% t intervals, † excluding zero. On the homophilous citation graphs the untuned configuration is far below the published accuracies; the comparison is within a row.

Dataset	Diagonal	Bundle (orthogonal)	General	General − Diagonal	General − Bundle
Texas	78.9 ± 3.8	75.9 ± 5.6	74.1 ± 5.3	$-4.87\,[-7.41, -2.32]^\dagger$	$-1.89\,[-7.59, +3.80]$
Wisconsin	76.9 ± 4.4	67.3 ± 11.0	74.1 ± 5.7	$-2.74\,[-4.96, -0.53]^\dagger$	$+6.86\,[-0.94, +14.66]$
Cornell	65.4 ± 6.5	62.4 ± 5.3	65.4 ± 5.9	$+0.00\,[-3.41, +3.41]$	$+2.97\,[-2.30, +8.24]$
Film	35.3 ± 0.9	34.0 ± 1.3	34.7 ± 0.8	$-0.61\,[-1.20, -0.02]^\dagger$	$+0.68\,[-0.23, +1.59]$
Squirrel	24.4 ± 1.6	25.0 ± 1.2	24.0 ± 1.8	$-0.46\,[-2.45, +1.53]$	$-1.03\,[-2.39, +0.33]$
Chameleon	35.1 ± 3.1	36.6 ± 3.3	34.0 ± 2.5	$-1.10\,[-3.06, +0.87]$	$-2.55\,[-5.13, +0.04]$
Citeseer	60.8 ± 3.9	50.1 ± 5.0	55.5 ± 4.9	$-5.31\,[-7.09, -3.53]^\dagger$	$+5.40\,[+3.01, +7.78]^\dagger$
Pubmed	88.2 ± 0.4	87.4 ± 0.5	88.1 ± 0.5	$-0.15\,[-0.46, +0.17]$	$+0.70\,[+0.30, +1.11]^\dagger$
Cora	55.8 ± 9.7	47.8 ± 8.3	52.6 ± 8.2	$-3.18\,[-10.07, +3.72]$	$+4.83\,[-2.09, +11.75]$

It does not, and the reason is instructive. That implementation learns a map per directed incidence but forms a *single* coupling per undirected edge, $-F_{u\trianglelefteq e}^\top F_{v\trianglelefteq e}$, retaining only the lower-triangular half and filling the transposed block by transposition. A product of orthogonal matrices is orthogonal, so the coupling is exactly norm preserving. Our own block-$SO(2)$ implementation instead *averages* two independently parameterized couplings, which is what produces $\cos(\psi)Q(\phi)$.

Table 37: Singular values of the effective neighbor coupling built from identical random angles with the reference implementation's own orthogonal transform. A range of $[1, 1]$ is norm preserving.

Orthogonal map	d	single product $F_u^\top F_v$	average of two couplings
Householder	3	[1.0000, 1.0000]	[0.0868, 1.0000]
Householder	4	[1.0000, 1.0000]	[0.0158, 0.9997]
Euler	3	[1.0000, 1.0000]	[0.0819, 1.0000]

The $d = 2$ closed form matches $\cos(\psi)Q(\phi)$ to 1.5×10^{-7}, with implied attenuation $|\cos\psi| \in [0.036, 0.969]$ over the sampled angles. We also read the assembly code of two further released sheaf models, the directed sheaf hypergraph network used above and a sheaf-based recommender; both form the single product. In the three released codebases we inspected, none uses the averaging construction. A fourth, the bundle model, has not released code.

Two consequences. First, the corollary to Proposition 4 must be stated for the averaging construction only; a blanket claim that $SO(d)$-restricted models can attenuate would be false for the single-product construction. Second, and more usefully, *whether a declared orthogonal structure group survives to the realized operator depends on an implementation detail that no paper states*. Two released codebases in the same literature make opposite choices, and neither documents the choice as consequential. That is a sharper instance of this paper's thesis than the broad claim would have been: the nominal-versus-effective gap is not merely a property of the declared family, it is contingent on assembly details that are currently invisible in published descriptions.

C Domain transfer: a staged comparison in recommendation

Every experiment so far is node classification. We close with a comparison from a different task family, social recommendation, where the same nominal ordering of restriction-map complexity is available: no social graph, raw scalar transport, gated scalar, gated diagonal, and gated matrix transport, with homophily scores computed from training interactions only.

Two features of this comparison matter.

First, the no-graph arm is not merely worse but near-degenerate (0.000605 against 0.048944), so this task sits deep in the regime where graph structure is essential rather than merely helpful. Second, within that regime, nominally increasing the complexity of the restriction maps makes NDCG@20

Table 38: Staged comparison in social recommendation (LastFM; five seeds, 500 epochs, 30 runs; mean over seeds). Variants were trained independently and are *not* an audited nested ladder.

Variant	NDCG@20	Recall@20	NDCG@100	Recall@100
no social graph	0.000605	0.001806	0.001599	0.007435
raw scalar	0.046657	0.103771	0.068112	0.223155
gated scalar	**0.048944**	**0.108869**	**0.071024**	**0.231970**
gated diagonal	0.042119	0.092300	0.066163	0.225704
gated matrix	0.035337	0.081997	0.056656	0.201487

monotonically *worse*: gated diagonal loses 14.0% relative to gated scalar, and gated matrix loses 27.8%. On all four metrics gated scalar is best and gated matrix is worst among the graph variants; the middle of the ordering is metric dependent (at cutoff 100 gated diagonal is slightly above raw scalar). Whatever the extra coordinate-mixing capacity is doing, it is not being converted into predictive value here.

This is a staged architecture comparison, not an audited capability ladder. The five variants were trained independently. They were not constructed to contain one another exactly as function classes, no child was warm-started from its fitted parent, no parent checkpoint was retained as an eligible validation candidate, and no empirical-risk monotonicity was audited. By Standard 2 and Standard 3 of Section 4, the differences above are therefore *not* marginal capacity effects, and we do not report them alongside the audited Γ_k of Section A or combine them into η_{sheaf}.

That restriction is the reason to include the comparison rather than a reason to omit it. Section B.2 showed that on a directed architecture the same class of unaudited comparison reported a capacity effect of -29.59 whose sign reversed under conditioning, and that the apparent advantage vanished once the optimization pathology was removed. The recommendation comparison has exactly the structure that failure mode requires: independently trained variants along a nominal complexity ordering, with no collapse rate reported and no nesting audit. We therefore read it as evidence about the *reporting convention*, not about relational capacity: a conventional scalar/diagonal/matrix endpoint comparison cannot distinguish a genuine capacity effect from an optimization or path effect, in either direction, and this holds in a domain far from the node-classification benchmarks where the convention originated.

A coarse alternative explanation, excluded. In the bundled recommendation endpoint of the bundled comparison, the sheaf-specific parameters account for $21{,}440$ of $645{,}184$ total, or 3.3%; the remainder is embedding capacity. A reader might attribute an endpoint difference to the sheaf model simply having far more parameters, and that explanation is unavailable here. We claim nothing stronger: a parameter share rules out a crude capacity-count story, and does not establish that the difference originates in sheaf transport.

What the three settings share. Taken with Sections A and B, the same principle recurs in three settings that share no architecture, no task, and no benchmark family: a two-layer normalized sheaf Laplacian on heterophilous node classification, a directed sheaf hypergraph network, and gated social recommendation. In all three, nominal relational complexity fails to translate into effective useful capability — through normalization absorbing a nominal distinction, through optimization determining the sign of a measured effect, and through added map complexity monotonically reducing predictive value. We regard this recurrence as stronger evidence for the paper's thesis than any single regularity across benchmarks would be.

D PROOFS

Proof of Theorem 1. *Blocks.* In the averaged, row-normalized layer the block acting from h_u on node v is $-(R_{uv} + R_{vu}^{\top})/(2d_v)$ and the diagonal block is $(2d_v)^{-1}\sum_{u\sim v}(I + R_{vu}^{\top}R_{vu})$; both are sums of products of elements of $\mathcal{S}$, their transposes and I. In the single-product layer the unnormalized blocks are $\sum_{e\ni v} F_{v\trianglelefteq e}^{\top}F_{v\trianglelefteq e}$ and $-F_{v\trianglelefteq e}^{\top}F_{u\trianglelefteq e}$, and the normalizer is $B_v^{-1/2}$ with $B_v =$

$I + \sum_{e \ni v} F^\top F \in \mathcal{A}$ symmetric positive definite. If p is a real polynomial with $p(\lambda) = \lambda^{-1/2}$ on the finite spectrum of B_v, then $B_v^{-1/2} = p(B_v) \in \mathcal{A}$. Hence $\tilde{L}_\mathcal{F} \in M_n(\mathcal{A})$.

Decomposition. $\mathcal{A}$ is closed under transpose, so the orthogonal complement of an $\mathcal{A}$-invariant subspace W is invariant: for $a \in \mathcal{A}$, $w \in W$, $w' \in W^\perp$, $\langle aw', w\rangle = \langle w', a^\top w\rangle = 0$. Therefore $\mathbb{R}^d$ is an orthogonal direct sum of irreducible $\mathcal{A}$-modules. Collect isomorphic ones into isotypic components $V_i \cong U_i^{\oplus m_i}$; different components are orthogonal, because the orthogonal projection onto a component commutes with $\mathcal{A}$ and there is no nonzero intertwiner between non-isomorphic irreducibles. By Schur's lemma $\mathrm{End}_\mathcal{A}(U_i)$ is a real division algebra, hence $\mathbb{R}$, $\mathbb{C}$ or $\mathbb{H}$ by Frobenius' theorem, and $\mathcal{A}$ acts on U_i as $M_{n_i}(\mathbb{D}_i)$ (Wedderburn–Artin; Lam, 2001). An intertwiner T between two copies of U_i satisfies $T^\top T \in \mathrm{End}_\mathcal{A}(U_i)$, because the commutant of a transpose-closed algebra is transpose-closed. $T^\top T$ is symmetric positive definite; if λ is one of its eigenvalues then $T^\top T - \lambda I$ is a singular element of a division algebra, hence zero. So $T/\sqrt{\lambda}$ is an isometry, and orthonormal bases of the copies can be chosen that are carried to each other by these isometries. Assembling them gives $P \in O(d)$ with $P^\top a P = \bigoplus_i \rho_i(a) \otimes I_{m_i}$ for every $a \in \mathcal{A}$, where ρ_i is the action on U_i. Applying this to each block of $\tilde{L}_\mathcal{F} \in M_n(\mathcal{A})$ and reordering coordinates gives the claim. □

Proof of Proposition 3. In either assembly the unnormalized operator is $L = \delta^\top \delta \succeq 0$, a sum over undirected edges of terms $\|F_{u \trianglelefteq e} x_u - F_{v \trianglelefteq e} x_v\|^2$ (for the averaged assembly, over stored directions with $F = R_{uv}$ on the source and I on the target). Since $\|a-b\|^2 \le 2\|a\|^2 + 2\|b\|^2$, summing edge by edge gives $x^\top L x \le 2\, x^\top D_F x$, where D_F is the block diagonal of L. With $B = D_F + I \succ D_F$, the normalized $\tilde{L} = B^{-1/2} L B^{-1/2}$ is symmetric and satisfies $0 \preceq \tilde{L} \preceq 2B^{-1/2} D_F B^{-1/2} \prec 2I$, so $P = I - \frac{1}{2}\tilde{L}$ has spectrum in $(0, 1]$. For row normalization with a constant scalar c, Proposition 1 gives the eigenvalues $a(c)[(1 - \tau_{\text{eff}}) + \tau_{\text{eff}}\mu]$ over the spectrum $\mu \in [-1, 1]$ of $D^{-1}A$; on a graph with μ near -1 this is negative as soon as $\tau_{\text{eff}} > \frac{1}{2}$, i.e. $c > 1$, and the diagonal $(3 - c^2)/4$ is negative for $c > \sqrt{3}$. On random graphs with random dense maps the symmetric bound held in all 400 draws (eigenvalues in $[0.09, 1.00]$), and row-normalized constant scalars gave smallest eigenvalues -0.30 at $c = 1.5$ and -0.77 at $c = 1.9$. □

Table 39: What the fitted gate rung realizes on Minesweeper (five seeds per row; the ladder is trained exactly as in the main experiments and stopped after the sign rung). Per seed and layer we read the learned per-direction scalars and the spectrum of $P = I - \frac{1}{2}\tilde{L}$; ranges are over seeds. "Off": layers with $c < 0.05$ on 95% of edges. Proposition 3 requires the last row to stay in $(0, 1]$.

Normalization	L	Layers off	Layers with $\lambda_{\min} < 0$	Largest median c	Nodes with negative self-weight	Smallest eigenvalue	Largest eigenvalue
row	2	0–0	1–2	0.65 – −1.17	2–11%	−1.277 to −0.679	0.984
row	4	2–3	1–2	1.42 – −1.77	22–84%	−14.588 to −1.557	0.913
augmented symmetric	4	0–1	0–0	2.48 – −12.11	0–0%	0.001 to 0.025	1.000

Proof of Corollary 1. (i) $\mathcal{A} = \mathbb{R}I$; the entries of M follow by collecting the coefficients of h_v and h_u in the blocks above. (ii) $I_{d/2} \otimes R(\theta) = \cos\theta\, I + \sin\theta\, J$ with $J = I_{d/2} \otimes R(\pi/2)$, $J^2 = -I$, $J^\top = -J$, so $\mathcal{A} = \mathrm{span}\{I, J\} \cong \mathbb{C}$ with transpose acting as conjugation; identifying J with multiplication by i turns the blocks of (i) into their complex counterparts, and for the single product $(L)_{vu} = -\bar{z}_{vu} z_{uv} = \overline{(L)_{uv}}$. (iii) Diagonal matrices commute and $\mathcal{A} \subseteq \mathbb{R}^d$. (iv) $I_{d/2} \otimes R(\theta)$ and D are both block diagonal with 2×2 blocks, so $\mathcal{A} \subseteq M_2(\mathbb{R})^{d/2}$, with equality for generic θ and D. (v) A generic matrix together with its transpose generates $\mathbb{R}^{d \times d}$. □

Lemma 1 (Layer normalization with $\epsilon > 0$)**.** *Let* $\mathrm{LN}_\epsilon(x) = (x - \mu\mathbf{1})/\sqrt{\sigma^2 + \epsilon}$ *with* μ, σ^2 *the mean and variance of the entries of* x*,* $\sigma^2 > 0$*. For every* $a > 0$*,*

$$\frac{\|\mathrm{LN}_\epsilon(ax) - \mathrm{LN}_\epsilon(x)\|}{\|\mathrm{LN}_\epsilon(x)\|} \;\le\; \frac{\epsilon\,|a^{-2} - 1|}{2\sigma^2}.$$

In Proposition 1, $a(c) \in (\frac{3}{4}, 1]$ *on* $(0, \sqrt{3})$*, so* $|a^{-2} - 1| \le \frac{7}{9}$ *and the bound is* $\frac{7}{18}\epsilon/\sigma^2$*. The learned affine map of layer normalization is applied afterwards and is common to both models.*

Proof. $\mathrm{LN}_\epsilon(ax) = (x - \mu\mathbf{1})/\sqrt{\sigma^2 + \epsilon a^{-2}} = r\,\mathrm{LN}_\epsilon(x)$ with $r^2 = (\sigma^2 + \epsilon)/(\sigma^2 + \epsilon a^{-2})$, so the left side is $|r - 1| = |r^2 - 1|/(1 + r)$. Here $|r^2 - 1| = \epsilon|a^{-2} - 1|/(\sigma^2 + \epsilon a^{-2})$, and $(\sigma^2 + \epsilon a^{-2})(1 + r) = (\sigma^2 + \epsilon a^{-2}) + \sqrt{(\sigma^2 + \epsilon)(\sigma^2 + \epsilon a^{-2})} \ge 2\sigma^2$. □

On 2000 random vectors and scales the ratio of the actual error to the bound never exceeded one and came within 10^{-4} of it, so the constant cannot be improved.

The interaction alone does not identify necessity. Let $y = \beta p + \gamma s + \xi$ with p, s, ξ independent and centered; rung $k-1$ regresses y on p, rung k on (p, s), and risks are population mean squared errors, so $\Gamma_k(\mathcal{D}) = \gamma^2 \mathrm{Var}(s)$. *World A (necessity):* s carries property j and the intervention replaces it by an independent copy, so $\Gamma_k(\mathcal{D}_{-j}) = 0$. *World B (side effect):* s is a nuisance channel unrelated to property j, and the intervention, meant to remove property j, also resamples s; again $\Gamma_k(\mathcal{D}_{-j}) = 0$. The pairs $(\Gamma_k(\mathcal{D}), \Gamma_k(\mathcal{D}_{-j}))$, and hence $\Psi_{j,k}$, coincide, yet rung k depends on property j only in world A: what separates them is a statement about the intervention (gate G2), not about the increments. *World C (collapse):* as A, but the intervention also resamples p. The increments are unchanged while the parent's risk after intervention rises to $\mathrm{Var}(y)$, so $\mathrm{Ret} = 0$, against $\mathrm{Ret} = \beta^2 \mathrm{Var}(p)/\mathrm{Var}(y) > 0$ in world A: this is what gate G3 measures, and it is the pattern of the last row of Table 12.

E AVERAGED BIDIRECTIONAL COUPLINGS

Proposition 4 (Averaging leaves the group)**.** *Let the two stored directions of an undirected edge carry orthogonal maps $g, h \in O(d)$ in the layer of Section 2. The neighbor coupling contributed by the pair is $\frac{1}{2}(g + h^\top)$, and its singular values are $|\cos(\varphi_i/2)|$, where $e^{i\varphi_i}$ are the eigenvalues of hg. It is an isometry if and only if $h = g^\top$; otherwise it contracts, and it annihilates a direction whenever hg has eigenvalue -1. For block rotations sharing one angle, $g = Q(\theta_{uv})$, $h = Q(\theta_{vu})$, this reads $\frac{1}{2}\big[Q(\theta_{uv}) + Q(\theta_{vu})^\top\big] = \cos(\psi)\, Q(\phi)$ with $\psi = \frac{1}{2}(\theta_{uv} + \theta_{vu})$, $\phi = \frac{1}{2}(\theta_{uv} - \theta_{vu})$: the symmetric angular component is a signed scalar attenuation and only the antisymmetric component is a reciprocal rotation.*

Proof. $\frac{1}{2}(g + h^\top) = h^\top \frac{1}{2}(I + hg)$ and $h^\top$ is orthogonal, so the singular values are those of $\frac{1}{2}(I + w)$ with $w = hg \in O(d)$. An orthogonal matrix is normal with eigenvalues $e^{i\varphi_i}$, hence $\frac{1}{2}(I + w)$ is normal with eigenvalues $\frac{1}{2}(1 + e^{i\varphi_i})$ of modulus $|\cos(\varphi_i/2)|$, and the singular values of a normal matrix are the moduli of its eigenvalues. All of them equal one iff $w = I$. For the block rotations, $Q(\theta_{vu})^\top = Q(-\theta_{vu})$ and $\frac{1}{2}[Q(\phi+\psi) + Q(\phi-\psi)] = Q(\phi)\frac{1}{2}[Q(\psi) + Q(-\psi)] = \cos(\psi) Q(\phi)$. □

Put differently, the average of two elements of a compact group lies in its convex hull and in general outside the group, whereas their product stays inside. The singular-value formula was checked on 1000 random pairs in $O(d)$, $d \in \{2, 3, 4, 5, 8\}$, to 10^{-15}.

Scope. The collapse is a property of the *assembly*, not of orthogonal maps as such. It arises when each stored direction is treated as its own edge, as in our implementation, which is a natural way to write a sheaf layer on a bidirectional edge list. It does *not* arise when a single coupling $F_{u \trianglelefteq e}^\top F_{v \trianglelefteq e}$ is formed per undirected edge and the transposed block is filled by transposition, because a product of orthogonal matrices is orthogonal. We inspected the assembly code of three released sheaf models (Bodnar et al., 2022; Mule et al., 2026; Purificato et al., 2025): all three form the single product, and with the orthogonal transform of Bodnar et al. (2022) the product's singular values are $[1, 1]$ while the averaged coupling's are $[0.09, 1]$ (Table 37). So the released models we checked are not affected; an implementation like ours is. The block formula $-F_{v \trianglelefteq e}^\top F_{u \trianglelefteq e}$ printed in these papers does specify the single product, but none of the ten papers we read (Appendix G) says how the two stored directions are assembled in code, and a per-direction implementation departs from the formula silently.

Proposition 4, measured. Splitting our block rotation into its symmetric part A (attenuation) and antisymmetric part R (reciprocal rotation) in a strict I, A, R, AR lattice, attenuation alone gains $+0.147$ and rotation alone $+0.155$ on Roman-empire, but rotation *after* attenuation retains only $+0.025$. On Minesweeper rotation after attenuation retains $+0.062$ while attenuation after rotation adds only $+0.005$. After controlling both learned strength and scalar gating in a full 2^3 lattice, rotation still adds $+0.023$ and $+0.057$ (Figures 2 and 3).

Order dependence. Table 40 gives rotation's increment in every context of the 2^3 lattice. On Cora the total gain is within 0.01 of zero and no share is defined. Because every cell of the lattice was fitted, the scores $v(S)$ can be inverted: the Harsanyi dividends $d(S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} v(T)$ (Table 41) are the unique decomposition of the lattice into additive contributions of each coalition, the Shapley value being their equal split. The order dependence of rotation's credit is then an identity, $[v(TGQ) - v(TG)] - [v(Q) - v(I)] = d(TQ) + d(GQ) + d(TGQ)$, and its sign is the sign of the interaction dividends. On Roman-empire every singleton dividend is about $+0.13$ to $+0.16$, every pairwise dividend about -0.13 and the triple dividend $+0.13$ (all resolved): any one of the three capabilities delivers essentially the whole gain and the other two add nothing on top of it, which is the strongest form of substitutability and the reason rotation's share can be anything between 13% and 94%. On Minesweeper the singletons are $+0.02$, $+0.04$ and $+0.08$, the pairwise dividends -0.01 to -0.02 and the triple $+0.007$: partial substitutes, with rotation the largest stand-alone capability, and a range of shares that is correspondingly narrower (53% to 79%). Negative interaction dividends are exactly what submodularity of the lattice asserts, and they identify the substitutes: a rotation added first is credited with what a gate or a propagation strength would otherwise have done.

Table 40: How much credit rotation receives depends on when it is added. Full 2^3 lattice over learned propagation strength T, edge gate G and block rotation Q (two layers, averaged assembly, every cell fitted for every unit); entries are the increment from adding Q in each context, the Shapley value over the six orders, the total gain $S(TGQ) - S(I)$, and the range of rotation's share between being added last and first. 95% t intervals over split means.

Dataset	Units	Q first	Q after T	Q after G	Q last	Shapley	Total	Share of Q
Roman-empire	20	$+0.159\,[+0.155, +0.164]$	$+0.031\,[+0.028, +0.035]$	$+0.023\,[+0.020, +0.026]$	$+0.023\,[+0.019, +0.026]$	$+0.070\,[+0.067, +0.073]$	$+0.169$	13–94%
Minesweeper	20	$+0.084\,[+0.071, +0.097]$	$+0.073\,[+0.067, +0.080]$	$+0.061\,[+0.045, +0.077]$	$+0.057\,[+0.041, +0.073]$	$+0.069\,[+0.062, +0.077]$	$+0.107$	53–79%
Cora	20	$+0.001\,[-0.001, +0.002]$	$+0.001\,[-0.003, +0.006]$	$-0.000\,[-0.003, +0.002]$	$-0.003\,[-0.008, +0.001]$	$-0.001\,[-0.003, +0.001]$	-0.005	—

Table 41: Harsanyi dividends of the 2^3 lattice of Table 40 (two layers, averaged assembly): $d(S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} v(T)$ over the cells v, with 95% t intervals over the ten official split means and † excluding zero. Singleton dividends are the increments of each capability added alone; the interaction dividends are the second and third differences. The order dependence of rotation's credit is exactly $d(TQ) + d(GQ) + d(TGQ)$; negative interaction dividends make the capabilities substitutes.

Dataset	$d(T)$	$d(G)$	$d(Q)$	$d(TG)$	$d(TQ)$	$d(GQ)$	$d(TGQ)$
Roman-empire	$+0.129\,[+0.125, +0.132]^\dagger$	$+0.146\,[+0.142, +0.150]^\dagger$	$+0.159\,[+0.155, +0.164]^\dagger$	$-0.128\,[-0.132, -0.124]^\dagger$	$-0.128\,[-0.131, -0.124]^\dagger$	$-0.136\,[-0.140, -0.132]^\dagger$	$+0.127\,[+0.122, +0.132]^\dagger$
Minesweeper	$+0.023\,[+0.014, +0.033]^\dagger$	$+0.044\,[+0.025, +0.063]^\dagger$	$+0.084\,[+0.071, +0.097]^\dagger$	$-0.017\,[-0.027, -0.008]^\dagger$	$-0.011\,[-0.025, +0.003]$	$-0.023\,[-0.048, +0.001]$	$+0.007\,[-0.007, +0.021]$

F STALK DIMENSION, EDGE GAUGE AND NODE GAUGE

Three further consequences of reading a layer through the algebra its maps generate. Each is verified numerically by a script in the supplementary code.

Below the rotated-diagonal rung, stalk dimension is width.

Proposition 5 (Stalk collapse)**.** *Let the layer of Section 2 have an even stalk dimension d and c channels, so that the hidden vector of a node has width dc, with an elementwise nonlinearity, layer normalization over the hidden vector and node-wise weights acting on the hidden vector. On rungs* 0–3 *every restriction map is aI_d or $a\,(I_{d/2} \otimes R(\theta))$ and acts identically on the $d/2$ coordinate planes of the stalk. Re-reading each hidden vector as $\frac{d}{2}c$ channels of a two-dimensional stalk therefore gives a network of the same rung, with the same parameters, that computes the same function, under either assembly and either normalization; rungs* 0–2 *reduce further to a one-dimensional stalk with dc channels. On rung* 4 *the identity holds if and only if the diagonal head is tied across planes, and the dense rung is the first that uses the stalk irreducibly.*

Proof. On these rungs the propagation operator is $M \otimes I_d$ or $M_{\mathbb{C}} \otimes I_{d/2}$ (Corollary 1(i)–(ii)); under the re-indexing the identity factor of the Kronecker product becomes a channel index, on which the operator acts trivially. The normalization on these rungs is a per-node scalar (row) or a per-coordinate scalar equal across coordinates (augmented symmetric, since $F^\top F \propto I_d$), layer normalization acts on the whole hidden vector and the nonlinearity elementwise, so all commute

with the re-indexing. On rung 4 the scaling $D = \mathrm{diag}(\delta_1, \ldots, \delta_d)$ differs between planes unless tied. The identity is checked to machine precision, including its failure on rung 4 for an untied head, by `test_stalk_collapse.py`. □

A comparison of "stalk dimension" between sheaf models that use scalar or rotation maps is therefore a comparison of widths, and the first rung at which d is a capacity of its own is the dense one, which never carries more than 9% of the audited gain (Table 19).

What the operator can identify: the edge gauge.

Proposition 6 (Identifiable dimension per edge)**.** *Under the single-product assembly an edge $e = \{u, v\}$ enters the sheaf Laplacian only through the Gram blocks $(F_{u\trianglelefteq e}^\top F_{u\trianglelefteq e}, F_{v\trianglelefteq e}^\top F_{v\trianglelefteq e}, F_{u\trianglelefteq e}^\top F_{v\trianglelefteq e})$, which are invariant under $(F_u, F_v) \mapsto (QF_u, QF_v)$ for $Q \in O(d)$ acting on the edge stalk, and generically this is the whole fibre. The number of parameters of an edge that the operator can identify is therefore the parameter count minus the dimension of the gauge that preserves the rung's family: gate and sign, 2 of 2 (the fibre is $\{\pm 1\}$: only $\mathrm{sign}(c_u c_v)$ is identifiable, not the signs of c_u and c_v); rotation, 3 of 4 (only the difference of the two angles); rotated diagonal, $2d + 1$ of $2d + 2$; dense, $2d^2 - d(d-1)/2$ of $2d^2$, i.e. 26 of 32 at $d = 4$. Under the averaged assembly the two stored directions enter separately and every parameter is identifiable.*

Proof. Invariance is immediate from $(QF)^\top(QF) = F^\top F$. For generic (F_u, F_v) of full rank the Gram blocks determine (F_u, F_v) up to a common orthogonal factor, which is the classical statement that a Gram matrix determines a configuration up to isometry, applied to the $2d$ columns of $[F_u\ F_v]$. Restricting Q to the rung's family gives the listed dimensions. The dimension of the image of the parameterization is the generic rank of its Jacobian, which `test_fiber_rank.py` computes in double precision at random points and finds equal to each count; for the averaged assembly it finds the full parameter count. □

This is complementary to the $d^2 - d$ unidentified degrees of freedom of Liu (2026), which concern what the labels leave undetermined: the edge gauge is what the operator itself cannot distinguish, whatever the labels. That the averaged assembly identifies everything is another face of Proposition 4: averaging two independently parameterized directions leaves the orbit structure that the product respects.

A gauge-trivial field is a node-wise conjugation.

Proposition 7 (Node gauge)**.** *Under the single-product assembly, the sign rung couples u and v through $s_e = \mathrm{sign}(c_u c_v)$ and the rotation rung through $R(\psi_e)$ with $\psi_e = \theta_{v\trianglelefteq e} - \theta_{u\trianglelefteq e}$. If the signed graph is balanced in the sense of Harary, $s_e = \sigma_u \sigma_v$ for some $\sigma \in \{\pm 1\}^V$, the sign-rung operator equals $\Sigma\, L_{|c|}\, \Sigma$ with $\Sigma = \mathrm{blockdiag}(\sigma_v I_d)$ and $L_{|c|}$ the gate-rung operator of the magnitudes; if the angle field has trivial holonomy on every cycle, $\psi_e = \phi_v - \phi_u$ for a node potential ϕ, the rotation-rung operator equals $\Phi\, L_a\, \Phi^\top$ with $\Phi = \mathrm{blockdiag}(I_{d/2} \otimes R(\phi_v))$ and L_a the scalar operator of the magnitudes. Both conditions hold if and only if the auxiliary normalized signed, respectively magnetic, Laplacian built from the field alone, with entries $w_e s_e$, respectively $w_e e^{i\psi_e}$, and degree normalization, has eigenvalue zero on every connected component. This auxiliary operator, which Table 42 reports, is not the propagation operator: the kernel of $L = \delta^\top \delta$ also depends on the incidence magnitudes (with $F_{u\trianglelefteq e} = 2$ and $F_{v\trianglelefteq e} = 1$ on every edge of a triangle, all signs positive and no holonomy, L has spectrum $\{1, 7, 7\}$). The conjugation commutes with neither layer normalization nor the nonlinearity, so, as in Theorem 1, the statement concerns the operator.*

Proof. Write $F_{u\trianglelefteq e} = |c_u|\sigma_u I_d$; then $F_u^\top F_v = |c_u||c_v|\sigma_u\sigma_v I_d$ and $F_u^\top F_u = c_u^2 I_d$, so every block of L acquires the factor $\sigma_u\sigma_v$, which is conjugation by Σ; the diagonal blocks and hence the normalization are unchanged. For rotations, $F_u^\top F_v = a_u a_v R(\phi_v - \phi_u) = a_u a_v R(\phi_u)^\top R(\phi_v)$ on each plane, which is conjugation of the scalar block by Φ. The spectral characterizations are the classical ones for signed graphs (Zaslavsky, 1982) and for magnetic Laplacians (Fanuel et al., 2017): the quadratic form of the normalized operator vanishes on a nonzero vector if and only if a switching, respectively a gauge, makes all couplings positive. □

What is left after both gauges: the moduli of a rung. Propositions 6 and 7 quotient the map parameters by the edge gauge and the node gauge. What remains, at the operator level and up to the magnitudes, is a classical object. For the sign rung it is the set of switching classes of signed graphs, $H^1(G;\mathbb{Z}/2) \cong (\mathbb{Z}/2)^{\beta_1}$ with $\beta_1 = |E| - |V| + 1$ the cycle rank (Zaslavsky, 1982); for the rotation rung it is the set of flat $U(1)$-bundles up to isomorphism, i.e. holonomy representations $\pi_1(G) \to U(1)$ of the free group of rank β_1, the torus T^{β_1}, whose dimension agrees with Proposition 6 ($|E|$ identifiable angle differences minus $|V| - 1$ node rotations); for the rotated-diagonal rung it is $(T^{\beta_1})^{d/2}$, and for the orthogonal part of the dense rung the character variety $\mathrm{Hom}(\pi_1(G), O(d))/O(d)$. The ladder is thus, at the operator level, a chain of moduli spaces $\{*\} \subset H^1(G;\mathbb{Z}/2) \subset T^{\beta_1} \subset (T^{\beta_1})^{d/2} \subset \mathrm{Hom}(\pi_1(G), O(d))/O(d)$, the spectrum of the magnetic Laplacian being a function on the torus (Fanuel et al., 2017), and the holonomy statistics of Table 42 locate the fitted field on it: a gauge-trivial field sits at the origin. The same quotient explains why the averaged assembly credits differently: it does not quotient by the edge gauge at all (Proposition 6), so two fields that are one point of these moduli spaces can be two different operators for it.

Proposition 7 gives a necessary condition for a sign or rotation increment to be a capability of the operator rather than of the nonlinearity's interaction with a change of basis: the fitted field must be frustrated, respectively carry holonomy. Both are measurable on the fitted models. `probe_gauge.py` trains the ladder as in the main experiments under the released operator and reports, per layer, the spectral gap of the signed and magnetic Laplacians, the fraction of fundamental cycles that are unbalanced or carry holonomy, the residual of the best node potential, and the test score of the fitted network when its field is replaced by the nearest gauge-trivial one, without retraining.

Table 42: Gauge structure of the fitted sign and rotation rungs under the released operator (single-product assembly, augmented symmetric normalization; n seeds, seed → official split). $\lambda_{\min}$ is the smallest eigenvalue of the normalized signed (sign rung) or magnetic (rotation rung) Laplacian of the fitted field, zero if and only if the field is gauge-trivial; "cycles" is the percentage of fundamental cycles that are unbalanced, respectively carry holonomy above 0.1 rad; "residual" is the weight fraction that the best node potential cannot align. Per-layer quantities are ranges over layers of seed means. "field" is the fraction of negative edges (sign rung) or the median $|\psi_e|$ in radians (rotation rung). Γ is the rung's increment; "drop" is the loss of test score when the fitted field is replaced by its nearest gauge-trivial field, and "drop$_0$" when it is replaced by the constant field (all positive, respectively all zero), both without retraining.

Dataset	L	rung	n	field	$\lambda_{\min}$	cycles (%)	residual	Γ	drop	drop$_0$
Minesweeper	2	sign	5	0.00–0.00	0.000–0.000	0–4	0.000–0.000	$+0.0034$	-0.0000	$+0.0001$
Minesweeper	2	rotation	5	0.11–0.28	0.002–0.003	86–88	0.004–0.005	$+0.0236$	$+0.0086$	$+0.0416$
Minesweeper	4	sign	5	0.00–0.12	-0.000–0.001	0–27	0.000–0.014	$+0.0085$	$+0.0037$	$+0.0162$
Minesweeper	4	rotation	5	0.07–0.25	0.001–0.002	46–82	0.001–0.004	$+0.0467$	-0.0001	$+0.0841$
Roman-empire	2	sign	5	0.00–0.00	0.000–0.000	0–0	0.000–0.000	$+0.0010$	$+0.0000$	$+0.0000$
Roman-empire	2	rotation	5	0.09–0.15	0.000–0.001	10–58	0.010–0.083	$+0.0015$	$+0.0053$	$+0.0032$
Roman-empire	4	sign	5	0.00–0.09	-0.000–0.000	0–7	0.000–0.009	-0.0003	$+0.0001$	-0.0002
Roman-empire	4	rotation	5	0.21–0.72	0.000–0.001	52–70	0.064–0.207	$+0.0232$	$+0.0137$	$+0.0372$

Table 42 reads the fitted fields. The sign rung is gauge-trivial wherever it was fitted: at two layers no layer on either dataset has a negative edge, at four layers at most 12% of the edges are negative, $\lambda_{\min} \leq 0.001$ throughout and the residual is at most 0.014; replacing the field by the balanced field nearest to it moves the score by at most 0.004, in line with increments Γ_2 of at most $+0.009$. Signed propagation is available to the ladder and is not used: by Proposition 7 the fitted sign rung is, at the operator level, the gate rung in a switched basis. The rotation rung differs in one respect and not in another. Its fitted angle field is far from gauge-trivial, with holonomy above 0.1 rad on 46–88% of the fundamental cycles on Minesweeper and 10–70% on Roman-empire. Yet on Minesweeper the network does not use that holonomy: replacing the field by its nearest holonomy-free field costs $+0.009$ at two layers and nothing at four, against increments Γ_3 of $+0.024$ and $+0.047$, whereas removing the field altogether (angle zero, which is the sign rung's operator under the rotation rung's weights) costs $+0.042$ and $+0.084$. What the rotation rung contributes on Minesweeper is the gauge part of its field, a node potential ϕ: at the operator level a conjugation of the scalar rung that adds nothing, which the layer normalization and the nonlinearity, neither of which commutes with it, turn into a usable node-wise change of basis. On Roman-empire at four layers the two drops are $+0.014$

and $+0.037$ against $\Gamma_3 = +0.023$, so about half of the increment goes with the holonomy and the rest with the potential. This is the operator-level counterpart of the algebra split of Table 19: the part of the sheaf-exclusive gain that Theorem 1 labels commutative is, on Minesweeper, also gauge-trivial, and the question of Grover & Bourgerie (2026), whether sheaf networks use holonomy, has different answers on the two datasets. The drops are measured without retraining, so they bound what the field carries in the fitted network, not what a network retrained under the constraint would recover.

The third division algebra: a quaternion rung. Theorem 1 allows $D_i \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$, and the ladder realizes $\mathbb{R}$ (gate, sign) and $\mathbb{C}$ (rotation) but not $\mathbb{H}$. The missing rung is $F = a\,(I_{d/4} \otimes L_q)$ with q a unit quaternion and $L_q \in SU(2) \subset SO(4)$ its left-multiplication matrix on $\mathbb{H} = \mathbb{R}^4$. Parameterized as $q = \exp(\theta\,\mathbf{i} + v_j\,\mathbf{j} + v_k\,\mathbf{k})$, with θ from the rotation head and (v_j, v_k) from a zero-initialized head, it contains the rotation rung exactly ($v = 0$ gives $\cos\theta\, I + \sin\theta\, L_{\mathbf{i}} = I_{d/2} \otimes R(\theta)$ in the model's plane convention), so the warm start is a function-preserving embedding, and its maps generate $\mathcal{A} \cong \mathbb{H}$: non-commutative, yet a division algebra. It is incomparable with the rotated-diagonal rung ($SU(2) \cdot \mathbb{R}$ and $\{QD\}$ contain neither the other), so it is a side branch of the chain rather than a new rung, and by the argument of Proposition 5 it is the first family whose minimal stalk dimension is four ($\mathbb{R} \to 1$, $\mathbb{C} \to 2$, $\mathbb{H} \to 4$, $M_k(\mathbb{R}) \to k$), before the dense rung. `test_quaternion_rung.py` checks the quaternion relations of the generators, the exact embedding under both operators, the generated dimension (4, against 2 for rotations) and the collapse of a stalk-8 quaternion network to stalk 4. `run_quaternion_branch.py` trains the branch with the main driver; its increment $\Gamma_{\mathbb{H}}$ is the value of non-commutativity without leaving a division algebra, to be compared with the non-commutative gain of the chain (Table 19).

Table 43: The quaternion branch under the released operator: identity $\subset$ gate $\subset$ sign $\subset$ rotation $\subset$ quaternion, trained with the main driver (n seeds, seed $\to$ official split, 95% t intervals over seeds, † excluding zero). Γ_3 is the rotation increment, $\Gamma_{\mathbb{H}}$ the increment of the quaternion rung over rotation, and "total" the identity-to-quaternion gain. The last columns read the fitted quaternion maps: the mean fraction of their Frobenius energy inside $\mathbb{C} = \mathrm{span}\{I, L_i\}$ and inside $\mathbb{H}$, and the numerical dimension of the algebra a sample of them generates (tolerance 10^{-2}; 2 for $\mathbb{C}$, 4 for $\mathbb{H}$).

Dataset	L	n	Γ_3	$\Gamma_{\mathbb{H}}$	total	energy in $\mathbb{C}$	in $\mathbb{H}$	generated dim
Minesweeper	2	10	$+0.0347\,[+0.0226, +0.0467]^\dagger$	$+0.0082\,[+0.0051, +0.0113]^\dagger$	$+0.0830\,[+0.0685, +0.0975]^\dagger$	0.952	1.000	3.8
Minesweeper	4	10	$+0.0747\,[+0.0647, +0.0846]^\dagger$	$+0.0167\,[+0.0119, +0.0216]^\dagger$	$+0.1553\,[+0.1454, +0.1652]^\dagger$	0.939	1.000	3.9
Roman-empire	2	10	$+0.0019\,[-0.0001, +0.0038]$	$+0.0060\,[+0.0033, +0.0087]^\dagger$	$+0.0390\,[+0.0344, +0.0436]^\dagger$	0.960	1.000	3.7
Roman-empire	4	10	$+0.0161\,[+0.0070, +0.0253]^\dagger$	$+0.0242\,[+0.0158, +0.0326]^\dagger$	$+0.1915\,[+0.1818, +0.2011]^\dagger$	0.875	1.000	4.0

Table 43 shows that the quaternion rung adds a resolved increment on both datasets at both depths: $+0.008$ and $+0.006$ at two layers, $+0.017$ and $+0.024$ at four, positive in 38 of 40 units and larger at depth on both. On Roman-empire at two layers it is three times the rotation increment, which is not resolved there ($+0.002$), and at four layers it exceeds it ($+0.024$ against $+0.016$). Read with Table 42: on Minesweeper the holonomy of the rotation rung's field is not used, yet the non-commutative extension of the same field is, so the value appears with non-commutativity rather than with the phase. The fitted quaternions stay near the complex plane, with 88–96% of their energy in $\mathbb{C}$ on average, less at four layers, and generate an algebra of numerical dimension 4, that is $\mathbb{H}$, as designed. Because the branch is incomparable with the rotated-diagonal rung, its increment is not a share of the chain's; it is a second measurement of what the non-commutative part of Table 19 buys, obtained with three angle parameters per incidence.

Which algebra the fitted maps generate. Theorem 1 classifies nominal families. The converse question is empirical: which algebra do the maps a trained rung actually produces generate? `probe_dense_algebra.py` reads the fitted per-incidence maps of the rotated-diagonal and dense rungs and reports, for each layer, the fraction of their Frobenius energy inside the nested subalgebras $\mathbb{R}I \subset \mathbb{C} \subset \mathbb{H}$ and $\mathbb{C} \subset M_2(\mathbb{R})^2 \subset M_4(\mathbb{R})$ (each a linear subspace, so the projections are orthogonal), and the numerical dimension of the algebra generated by a sample of the maps, computed as the rank, at a tolerance, of the span of all words of length at most two in the maps and their transposes. A dense rung whose maps keep most of their energy inside $M_2(\mathbb{R})^2$, or whose generated dimension stays well below 16, has a nominal capability it did not realize, which is the hidden equivalence produced by training rather than by the operator.

Table 44: Which algebra the fitted maps of the rotated-diagonal and dense rungs generate (released operator, seed means). Energy: the mean fraction of a map's Frobenius energy inside the subalgebras $\mathbb{R}I \subset \mathbb{C} \subset \mathbb{H}$ and $\mathbb{C} \subset M_2(\mathbb{R})^2$ (block-diagonal in the two planes; the rotated-diagonal rung lies in it exactly). "dim" is the numerical dimension of the algebra generated by a sample of the maps (tolerance 10^{-2}, at most 16) and PR the participation ratio of the singular values of the span of their words, a soft dimension.

Dataset	L	rung	layer	$\mathbb{R}$	$\mathbb{C}$	$\mathbb{H}$	$M_2(\mathbb{R})^2$	dim	PR
Minesweeper	2	diagonal	1	0.337	0.910	0.910	1.000	8.0	4.22
Minesweeper	2	diagonal	2	0.290	0.940	0.940	1.000	6.3	3.53
Minesweeper	2	dense	1	0.590	0.838	0.854	0.960	16.0	6.05
Minesweeper	2	dense	2	0.350	0.707	0.743	0.873	16.0	8.28
Roman-empire	2	diagonal	1	0.318	0.867	0.867	1.000	8.0	4.34
Roman-empire	2	diagonal	2	0.258	0.962	0.962	1.000	8.0	3.32
Roman-empire	2	dense	1	0.351	0.670	0.709	0.866	16.0	11.14
Roman-empire	2	dense	2	0.625	0.776	0.823	0.882	16.0	9.14

Table 44 answers the converse question. The rotated-diagonal rung lies in $M_2(\mathbb{R})^2$ by construction and keeps 87–96% of its energy in $\mathbb{C}$, generating an algebra of dimension 6–8. The dense rung's maps generate all of $M_4(\mathbb{R})$ in every layer and every seed (dimension 16 at tolerance 10^{-2}): training produces no hidden equivalence here, the nominal capability is realized. It is realized sparingly, however: 87–96% of the energy stays block-diagonal and 67–84% in $\mathbb{C}$, and the participation ratio of the generated algebra is 6–11 of 16. This is the map-level face of the dense rung never carrying more than 9% of the audited gain: the layer uses the full algebra, in small amounts.

A first-order descent diagnostic before the child is trained. A cheap companion to the increment is the gradient of the training loss at the fitted parent with respect to the zero-initialized head of the child: the unnormalized score of the nested model, without the Fisher normalization and nuisance adjustment that would make it a test statistic, which costs one backward pass and no training. A nonzero score means a first-order descent direction exists, so a null increment with nonzero score is a generalization or optimization outcome (*fit-only* in the anatomy of Table 18), not absence of capacity; a vanishing score means the parent is stationary in the child's parameters and *no-descent* is the expected outcome. `probe_gauge.py` records, for every transition, the norm of the score, the norm of the gradient with respect to the already active parameters for scale, and the best one-step decrease of the training loss along the score.

Table 45 gives the score at every transition. It is never zero: at every fitted parent, on both datasets and both depths, the gradient with respect to the child's zero-initialized head has norm between 0.02 and 2.1, and one gradient step lowers the training loss by between 0.6×10^{-3} and 141×10^{-3}. A first-order descent direction therefore always exists, so no null increment of the ladder is a stationary parent: the null increments are *fit-only* in the sense of Table 18, a better fit that did not validate. The score is also not a predictor of the increment. On Minesweeper at two layers the gate→sign transition has a large score (0.61) and an increment of $+0.003$, while identity→gate has the smallest score (0.02) and an increment of $+0.043$. The score measures how much the child can fit at the parent; the increment measures what that fit is worth on held-out data; the two are decoupled here, which is one more reason to attribute along an audited chain rather than by any quantity computed from the training loss.

G What published sheaf-network papers report

Related work. Sheaf Laplacians as a learning primitive (Hansen & Ghrist, 2019; Hansen & Gebhart, 2020) led to neural sheaf diffusion (Bodnar et al., 2022) and to many variants that compare restriction-map families as a design choice (Barbero et al., 2022a; Duta et al., 2023; Mule et al., 2026; Hernandez Caralt et al., 2024; Zaghen, 2024; Braithwaite et al., 2024; Purificato et al., 2025; Bamberger et al., 2025); we ask what such comparisons identify, and Appendix G audits what ten of them report. Recent work intervenes on the model instead, asking whether learnable Laplacians are needed, whether holonomy is used, why depth loses signal, how to filter it spectrally, and what the labels leave unidentified (Hernandez Caralt et al., 2026; Grover & Bourgerie, 2026; Bourgerie et al.,

Table 45: First-order descent diagnostic at every transition, computed at the fitted parent before the child is trained (released operator, seed means). $\|g_{\mathrm{new}}\|$ is the norm of the gradient of the training loss with respect to the zero-initialized head of the child (the unnormalized score of the nested model), $\|g_{\mathrm{act}}\|$ that with respect to the parameters already active (the parent is a validation-selected checkpoint, not a training minimum), $\Delta\mathrm{CE}$ the best one-step decrease of the training loss along that gradient ($\times 10^3$), and Γ the held-out increment the trained child then obtained.

Dataset	L	transition	$\|g_{\mathrm{new}}\|$	$\|g_{\mathrm{act}}\|$	$\Delta\mathrm{CE}$	Γ
Minesweeper	2	identity→gate	0.023	0.255	0.57	+0.0434
Minesweeper	2	gate→sign	0.608	0.865	3.56	+0.0034
Minesweeper	2	sign→rotation	0.961	0.981	3.74	+0.0236
Minesweeper	2	rotation→diagonal	0.204	0.818	8.18	—
Minesweeper	4	identity→gate	0.028	0.161	0.83	+0.0649
Minesweeper	4	gate→sign	0.587	0.259	1.77	+0.0085
Minesweeper	4	sign→rotation	0.120	0.416	2.10	+0.0467
Minesweeper	4	rotation→diagonal	0.131	0.374	4.19	—
Roman-empire	2	identity→gate	0.080	0.600	4.92	+0.0311
Roman-empire	2	gate→sign	0.538	0.545	2.48	+0.0010
Roman-empire	2	sign→rotation	0.259	0.678	1.91	+0.0015
Roman-empire	2	rotation→diagonal	0.326	0.700	6.22	—
Roman-empire	4	identity→gate	0.432	1.159	141.38	+0.1407
Roman-empire	4	gate→sign	2.076	0.892	6.94	−0.0003
Roman-empire	4	sign→rotation	0.554	1.901	6.99	+0.0232
Roman-empire	4	rotation→diagonal	0.692	1.416	18.77	—

2026; Borgi et al., 2025; Liu, 2026); Theorem 1 is complementary, being about which operators a family can realize. Propagation strength is a known axis in non-sheaf GNNs (Gasteiger et al., 2019; Chen et al., 2020), complex edge weights are classical (Zhang et al., 2021; Singer & Wu, 2012), and our evaluation concerns follow Platonov et al. (2023); Luan et al. (2022) and Ferrari Dacrema et al. (2019).

Table 46 records, for ten papers, the items of the reporting list in Section 7 that can be read off the paper text. We read the latest public version of each paper (for Braithwaite et al. (2024), the first version, which is the one we cite) and did not consult code, so an entry "not stated" may be settled by a repository. Four papers compare learned restriction-map families on real benchmarks; in three of them the families include diagonal and orthogonal maps, which are not nested in each other, and one of the four reports trivial-sheaf equivalents of the same model, implemented separately (one of the two re-run). Three papers include an MLP on real benchmarks, in each case with numbers taken from earlier work, so its depth, width and training protocol are those of the earlier study; a fourth includes non-graph recommenders, and none runs the same model with the graph removed. Three state the degree normalization used in the experiments and a fourth adopts that of earlier work. No paper reports a divergence or collapse rate for its own sheaf model; out-of-memory or instability events are marked in four, in two of them only for a baseline.

H Numerical checks and implementation notes

Nuisance audit of the feature permutation (gate G2). The within-label permutation is an oracle intervention: the true labels define the strata, and the permutation is drawn over all nodes of a stratum, across the train, validation and test splits. By construction it preserves the graph and the class-conditional multiset of feature vectors; G2 asks what else it changes, which `nuisance_audit.py` measures (Table 47). Two side effects are visible. First, wherever node degree is predictable from the features, the permutation removes that dependence along with the autocorrelation: on Tolokers the R^2 of log-degree on the leading principal components falls from 0.31 to 0.08 at the screening fraction, so the positive Tolokers interaction of Table 13 is an effect of removing feature autocorrelation *and* degree–feature dependence together. The screen's own G2 is therefore not met on Tolokers, and we apply the rule rather than reason around it: the cell is a refusal for any property-specific claim, and what the interval supports is that the gate rung's value depends on this joint intervention. The interaction is reported because a refusal is an output of the protocol, not a reason to hide the mea-

Table 46: What ten sheaf-network papers report, as stated in the paper text (latest public version; first version of Braithwaite et al. (2024), journal version of Purificato et al. (2025); code was not consulted; "not stated" means we did not find it, and an entry may be settled by code we did not read). The table records reporting, not correctness: every listed practice was standard at the time. "Trivial-sheaf control" asks for an identity- or scalar-map version of the *same* model on the real benchmarks.

Paper	Restriction-map families compared	Trivial-sheaf control (real data)	Graph-free baseline (real data)	Normalization used in the experiments	Failed runs
Hansen & Gebhart (2020)	one hand-crafted signed scalar sheaf (not learned)	no real data (GCN on synthetic graphs, own run)	none (no real data)	stated ($I - L_{\mathcal{F}}/d_{\max}$)	none mentioned
Bodnar et al. (2022)	diagonal, orthogonal, general	no (symmetric vs. general at $d{=}1$ on synthetic data)	MLP, numbers from earlier work	stated (augmented symmetric, appendix)	none reported; SVD gradient issue noted
Barbero et al. (2022a)	pre-computed vs. random orthogonal (learned families quoted)	no	MLP, numbers from the literature	normalized form by notation, not restated for experiments	none mentioned
Barbero et al. (2022b)	orthogonal only	no (notes GAT is the $d{=}1$ case)	none	attention-normalized; no degree normalization stated	out-of-memory (own model at 64 layers) and baseline instability marked
Duta et al. (2023)	diagonal, low-rank, general	yes (HyperGNN/HyperGCN as trivial-sheaf equivalents; one re-run)	none	searched as a hyperparameter; selected value not stated	instability of a baseline noted (synthetic)
Zaghen (2024)	diagonal, orthogonal	no	MLP, numbers from earlier work	stated	an underperforming variant disclosed
Hernandez Caralt et al. (2024)	learned vs. identity and diagonal (synthetic data only)	synthetic data only	none on the benchmark	that of earlier work (appendix)	exploding gradients noted as a limitation
Braithwaite et al. (2024)	diagonal, orthogonal, general	no (type-agnostic sheaf model as control)	none	background definition only	out-of-memory marked (a baseline)
Bamberger et al. (2025)	orthogonal only	no (notes GCN is the trivial bundle)	none (MLP on synthetic tasks, own run)	random-walk form defined; per-experiment choice not stated	out-of-memory marked (NSD baseline)
Purificato et al. (2025)	none (stalk-dimension ablation)	partly (one-dimensional stalks; dataset not stated)	non-graph recommenders only (MostPop, ItemKNN)	background definition only	none mentioned

surement; on Roman-empire, Minesweeper, Amazon-ratings and Questions the dependence is small before and after. Second, because the permutation crosses splits, per-split feature means move: by at most 0.05 standard deviations on the five large datasets, but by 0.5 standard deviations within splits and 2.6 within (label, split) cells on Cornell, whose 183 nodes make split composition the dominant effect; Cornell is refused by the screen on other grounds already. The audit does not certify G2, and no finite list of statistics can; it bounds what a reader must accept when reading a positive interaction as necessity.

Table 47: Nuisance audit of the within-label feature permutation (gate G2), at the fractions used in the paper, mean over three permutation seeds. Moran's I is the intended target. Degree–feature dependence is $|\mathrm{corr}(\log \deg, \mathrm{PC}_1)|$ and the R^2 of $\log \deg$ on the first sixteen principal components of the original features. "Split shift" is the largest change of a feature mean inside any train, validation or test split, in units of the feature's standard deviation; "class-cell shift" the same within any (label, split) cell, which the permutation changes only through split composition. The labels serve as strata only, so the intervention is an oracle one, and the permutation crosses splits.

Dataset	frac.	Moran's I	$\lvert\mathrm{corr}(\log \deg, \mathrm{PC}_1)\rvert$	$R^2(\log \deg)$	split shift	class-cell shift
Roman-empire	0.25	$0.038 \to 0.019$	$0.062 \to 0.051$	$0.056 \to 0.046$	0.03	0.31
Roman-empire	0.5	$0.038 \to 0.003$	$0.062 \to 0.036$	$0.056 \to 0.040$	0.04	0.57
Minesweeper	0.25	$0.046 \to 0.028$	$0.001 \to 0.007$	$0.006 \to 0.003$	0.03	0.08
Minesweeper	0.5	$0.046 \to 0.011$	$0.001 \to 0.006$	$0.006 \to 0.002$	0.04	0.13
Tolokers	0.25	$0.155 \to 0.095$	$0.219 \to 0.173$	$0.310 \to 0.180$	0.03	0.06
Tolokers	0.5	$0.155 \to 0.047$	$0.219 \to 0.125$	$0.310 \to 0.083$	0.05	0.07
Amazon-ratings	0.25	$0.168 \to 0.095$	$0.014 \to 0.011$	$0.002 \to 0.001$	0.03	0.13
Amazon-ratings	0.5	$0.168 \to 0.043$	$0.014 \to 0.005$	$0.002 \to 0.001$	0.04	0.19
Questions	0.25	$0.015 \to 0.008$	$0.018 \to 0.012$	$0.051 \to 0.029$	0.02	0.14
Questions	0.5	$0.015 \to 0.004$	$0.018 \to 0.010$	$0.051 \to 0.014$	0.03	0.20
Cornell	0.25	$0.040 \to 0.023$	$0.097 \to 0.106$	$0.217 \to 0.212$	0.48	2.56
Cornell	0.5	$0.040 \to 0.016$	$0.097 \to 0.116$	$0.217 \to 0.201$	0.59	2.73

Proposition 2 against released code. We instantiated the released diagonal Laplacian builder of Bodnar et al. (2022) with stalk dimension one, augmented normalization, and constant maps $c \in \{0.3, 1.0, 2.5\}$ on an 8-cycle, K_4, and an irregular 7-node graph, and compared the assembled operator with $s(sD+I)^{-1/2}(D-A)(sD+I)^{-1/2}$. The maximum absolute deviation is 7.5×10^{-9}. On the two regular graphs the operator equals $\frac{sk}{sk+1}(I - A/k)$ to within 1.2×10^{-7}, e.g. $\tau_{\text{eff}} = 0.1525,\ 0.6667,\ 0.9259$ on the 8-cycle. The congruence $S^{1/2} L_{\text{sym}} S^{1/2}$ of Proposition 2 equals that expression to 4×10^{-16} in double precision on random irregular graphs (degrees 2 to 10, $s \in \{0.05, 0.7, 1, 9\}$).

Theorem 1 against the implemented layers. On a random irregular 30-node graph we froze signed per-direction scalars drawn from a standard normal, applied each layer to every basis vector to obtain its dense operator, and compared it with $M \otimes I$ for the M of the proposition. The maximum absolute deviation is 4.4×10^{-16} (averaged assembly) and 3.3×10^{-16} (single product). With random block rotations in place of the scalars, the relative Frobenius distance to the least-squares nearest operator of the form $M \otimes I$ is 0.19 and 0.26.

The single-product ladder. For Section 5 we re-implemented the ladder with the single-product assembly and augmented symmetric normalization of released sheaf-diffusion code, changing nothing else. With maps up to the rotated-diagonal rung $F^\top F$ is diagonal and the block normalizer is elementwise; on the dense rung we use a coupled Newton–Schulz iteration for the inverse square root, which is differentiable at repeated eigenvalues where back-propagation through an eigendecomposition is not. In double precision the layer reproduces the operator of Proposition 2 to 1.6×10^{-15}, each rung loaded with its parent's parameters reproduces the parent's logits to 10^{-14}, the Newton–Schulz root agrees with an eigendecomposition to 2×10^{-14} at condition numbers up to 500, and the assembled coupling of two rotations has all singular values equal to one where the averaged coupling's range from 0.01 to 1.

Known-truth generator for SND. Each of 40 draws samples a random graph on 800 nodes with mean degree 8, a node signal, and random edge signs; in-neighbor aggregation gives an unsigned

and a signed channel, and the intervention permutes the signs. The parent class regresses on the node's own signal and the unsigned channel; the nested child adds one feature. Risk is held-out MSE of a ridge probe ($\lambda = 10^{-3}$, half the nodes for fitting), so Γ values are not bounded by one. The cells are: *necessity* (target depends on the signed channel, which is the added feature), *irrelevant* (target ignores it), *generic capacity* (added feature is an intervention-invariant nuisance channel that the target uses), and *collapse* (as necessity, but the intervention also corrupts a non-target parent input, a known-invalid intervention that must be refused).

Deviations from released code. For Mule et al. (2026) no source line was changed; comparisons alter one or two command-line flags. For Bodnar et al. (2022) the numerical checks use the Laplacian builder and orthogonal transform, and the family comparison of Table 36 runs the released training script with one fixed configuration; one line of a dataset loader was adapted to a newer graph-learning library that stores a three-tuple where the original code expected two, and the dataset files, which the release does not download, were obtained through that library.