SHEAFORMER: SHEAF DIFFUSION AS A STRUCTURED MULTI-HEAD OPERATOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Sheaf Neural Networks (SNNs) enrich graph diffusion with vector-valued stalks connected by learned restriction maps, and have been extended to directed hypergraphs by encoding orientation through a complex phase. We show that sheaf diffusion admits an exact stream-major decomposition whose diagonal specialization has the additive structure of the Transformer's multi-head attention: stalk coordinates act as heads, sheaf-induced node operators take the place of softmax attention matrices, and stalk and channel maps become structured value and output maps. We introduce SheaFormer, an SNN whose multi-head structure is inherited from the sheaf itself: it places sheaf propagation inside a residual block with a complex whitening of the propagated branch and generates role-aware restriction maps from the head and the tail of each hyperedge as bilinear node–context compatibilities. Whereas the phase encoding of the Directional Sheaf Hypergraph Network (DSHN) discards orientation at zero charge, the role-aware restrictions retain it and, in a single layer, distinguish orientations of a graph that no single DSHN layer with diagonal restriction maps distinguishes for any charge. Across twelve directed hypergraph benchmarks, the gain of the residual block grows as homophily decreases, while the role-aware restrictions improve eleven of the twelve benchmarks. SheaFormer attains the highest average accuracy, 2.03 points above the strongest baseline, and the highest F1 on two real-world molecular-reaction hypergraphs.

1 INTRODUCTION AND PREVIOUS WORKS

Graph Neural Networks (GNNs) and their hypergraph generalizations (HNNs) are key in modern deep learning due to their capability of propagating information by diffusion over the underlying structure, but are known to suffer from oversmoothing and to struggle in heterophilic settings, where connected nodes tend to exhibit dissimilar features or labels (Luan et al., 2024).

Sheaf Neural Networks (SNNs) (Hansen & Gebhart, 2020) address both limitations by extending graph diffusion to *sheaf diffusion*, which equips nodes and edges with vector spaces (*stalks*) related by linear *restriction maps*: Neural Sheaf Diffusion (NSD) (Bodnar et al., 2022) mitigates oversmoothing and performs well under heterophily. SNNs have been extended to directed graphs by the Directed Sheaf Neural Network (DSNN) (Fiorini et al., 2026a), to hypergraphs by the Sheaf Hypergraph Network (SHN) (Duta et al., 2023), and to directed hypergraphs by the Directional Sheaf Hypergraph Network (DSHN) (Mule et al., 2026). Both DSNN and DSHN encode direction through complex-valued incidence restrictions whose phase depends on whether a node belongs to the head or the tail of a hyperedge.

Since the seminal work of Vaswani et al. (2017), multi-head self-attention (MHA) has motivated many attention-based graph architectures, from GAT (Veličković et al., 2018) and GATv2 (Brody et al., 2022) to Graph Transformers (Ying et al., 2021; Rampášek et al., 2022; Shirzad et al., 2023), to hypergraph Transformers such as HGraphormer (Qu et al., 2026), and, in the sheaf setting, Sheaf Attention Networks (Barbero et al., 2022b; Fiorini et al., 2026b), which augment sheaf transport with GAT-style scalar attention. A related line of work has highlighted a connection between sheaf propagation and MHA: Bourgerie et al. (2026) interpret matrix-valued sheaf interactions as counterparts of attention coefficients and the stalk dimension as loosely analogous to multiple heads, Hajij et al. (2025) place SNNs and Transformers in a common copresheaf framework, D'Elia (2026)

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.

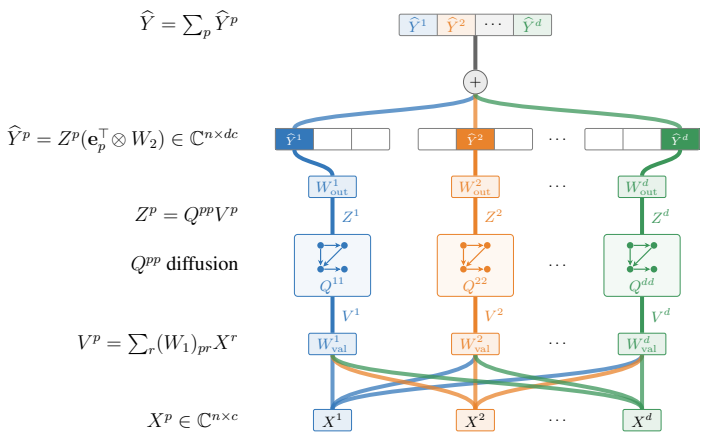


Figure 1: A diagonal sheaf layer has the additive multi-head form of MHA, with topology-constrained sheaf propagation in place of softmax attention—see Theorem 1.

gives a sheaf-theoretic reading of the Transformer block, and, most closely, Lin et al. (2026) prove an exact identity in the Transformer-to-geometry direction, representing MHA as an edge-dependent connection propagation.

Following a complementary direction that starts from the sheaf layer, we establish an exact stream-major decomposition for the complete linear sheaf layer (Theorem 1), whose diagonal specialization is the additive per-head form of MHA (Elhage et al., 2021) with sheaf propagation in place of attention. To our knowledge, no prior work derives this decomposition from the standard sheaf diffusion layer or identifies its stalk and channel transformations with value and output maps. Next, we propose *SheaFormer*, an SNN featuring a natural MHA structure already present in the sheaf operator itself (rather than being added to it). It consists of a residual block and role-aware restriction maps, whose contributions we evaluate separately. Our main contributions are as follows:

1. Starting from the standard sheaf diffusion layer, we derive an exact stream-major decomposition showing that, for diagonal sheaves, stalk coordinates act as the heads of an additive multi-head operator: sheaf-induced node operators occupy the position of attention matrices, while the stalk and channel transformations induce structured value and output maps.
2. Building on this correspondence, we introduce *SheaFormer*, a sheaf architecture with residual propagation, complex whitening of the propagated branch, and role-aware restriction maps generated from separate head and tail hyperedge contexts through bilinear compatibilities; we prove that these restrictions encode orientation even at zero charge, and that the complex whitening can move propagated features outside the harmonic space.
3. Across twelve directed hypergraph benchmarks, we show that the residual block and the proposed restriction maps provide complementary improvements, with SheaFormer gaining 3.00 accuracy points over DSHN on average and attaining the highest F1 on two molecular-reaction hypergraphs.

2 Preliminaries

Directed sheaf hypergraph diffusion. Let $\mathcal{H} = (V, E)$ be a directed hypergraph, with $|V| = n$ and $|E| = m$, where each hyperedge e has tail and head sets $T(e), H(e) \subseteq V$; a directed edge $u \to v$ places u in the tail and v in the head of the corresponding hyperedge. Let d be the stalk dimension and, for each incidence $i \triangleleft e$, let $F_{i\triangleleft e} \in \mathbb{R}^{d\times d}$ be a real restriction map. Following Mule et al. (2026), directionality is introduced through a complex phase where $q \in \mathbb{R}$ is a magnetic charge parameter (for $q = 0$, the phase carries no orientation): $\vec{F}_{i\triangleleft e} = S^{(q)}_{i\triangleleft e} F_{i\triangleleft e}$, where $S^{(q)}_{i\triangleleft e} = 1$ if $i \in H(e)$, $S^{(q)}_{i\triangleleft e} = e^{-2\pi \mathbf{i} q}$ if $i \in T(e)$, and $S^{(q)}_{i\triangleleft e} = 0$ if $i \notin e$. The *directed sheaf boundary operator* $B^{(q)} \in \mathbb{C}^{md\times nd}$ is the block matrix with $B_{ei} = \delta_e^{-1/2} \vec{F}_{i\triangleleft e}$ for $i \in e$ and zero otherwise, with $\delta_e = |e|$ the cardinality of hyperedge e. The associated *signless sheaf operator* and *directed*

sheaf Laplacian are defined as follows with $D_{\text{nds}} = \text{diag}(D_1, \dots, D_n)$:

$$Q_{\vec{\mathcal{F}}} = B^{(q)*}B^{(q)}, \qquad L_{\vec{\mathcal{F}}} = D_{\text{nds}} - Q_{\vec{\mathcal{F}}}, \qquad D_i = \sum_{e \ni i} F_{i \lhd e}^{\top} F_{i \lhd e}. \tag{1}$$

Sheaf neural networks. An SNN layer is obtained by combining a sheaf propagation operator $\Omega^{(t)} \in \mathbb{C}^{nd \times nd}$ with two learnable transformations $W_1^{(t)} \in \mathbb{C}^{d \times d}$ and $W_2^{(t)} \in \mathbb{C}^{c \times c'}$, which mix, respectively, stalk and feature coordinates. Letting $X^{(t)} \in \mathbb{C}^{nd \times c}$, with c feature channels, these architectures take the form

$$Z^{(t)} = \Omega^{(t)}(I_n \otimes W_1^{(t)})X^{(t)}W_2^{(t)} \in \mathbb{C}^{nd \times c'}, \qquad X^{(t+1)} = \Psi^{(t)}\left(X^{(t)}, Z^{(t)}\right), \tag{2}$$

where $\Psi^{(t)}$ applies the nonlinearity and, depending on the architecture, combines $Z^{(t)}$ with $X^{(t)}$. This encompasses NSD (Bodnar et al., 2022), DSNN (Fiorini et al., 2026a), SHN (Duta et al., 2023), and DSHN (Mule et al., 2026), which differ in whether $\Omega^{(t)}$ is a diffusion or Laplacian operator and in the choice of $\Psi^{(t)}$. In particular, DSHN employs $Q_{\vec{\mathcal{F}}_N^{(t)}} = I_{nd} - L_{\vec{\mathcal{F}}_N^{(t)}}$ with $L_{\vec{\mathcal{F}}N} := D_{\text{nds}}^{-1/2} L_{\vec{\mathcal{F}}} D_{\text{nds}}^{-1/2}$, defined whenever $D_{\text{nds}} \succ 0$.

Restriction maps. For graphs, restriction maps are constructed from the representations of the two incident nodes. For hypergraphs, each restriction map is constructed by Φ, an (optionally) learnable function, typically an MLP, producing a full $d \times d$ matrix, its diagonal, or the parameters of an orthogonal matrix, together with an aggregation function AGG of the incident nodes, where

$$\mathbf{h}_e = \text{AGG}\{X_j : j \in e\} \qquad F_{i \lhd e} = \Phi(X_i \| \mathbf{h}_e) \in \mathbb{R}^{d \times d}. \tag{3}$$

Complex-valued inputs to Φ are first mapped to the real domain through $\text{unwind}(x) = [\Re(x) \| \Im(x)]$. This encompasses NSD, SHN, DSHN, and DSNN up to architecture-specific details (Appendix B).

Transformer multi-head attention. A Transformer (Vaswani et al., 2017) represents n tokens by a matrix $X \in \mathbb{R}^{n \times d_{\text{model}}}$, where $d_{\text{model}} = Hd_h$, with H and d_h being respectively the number of heads and the head dimension. For each of its H attention heads, head h relies on the matrices $W_{\text{qry}}^h, W_{\text{key}}^h, W_{\text{val}}^h \in \mathbb{R}^{d_{\text{model}} \times d_h}$ to define the *attention matrix* $A^h(X)$ and the *multi-head self-attention* operator MHA:

$$A^h(X) = \text{softmax}\left(\frac{XW_{\text{qry}}^h (XW_{\text{key}}^h)^{\top}}{\sqrt{d_h}}\right), \qquad \text{MHA}(X) = \sum_{h=1}^{H} A^h(X) X W_{\text{val}}^h W_{\text{out}}^h. \tag{4}$$

Following Elhage et al. (2021), W_{val}^h maps the input into head h's value space, reading all d_{model} coordinates, and $W_{\text{out}}^h \in \mathbb{R}^{d_h \times d_{\text{model}}}$ maps back into the model space, writing into all of them. Moreover, up to the factor $1/\sqrt{d_h}$, the argument of the softmax is the bilinear form $XW_{\text{qry}}^h (W_{\text{key}}^h)^{\top} X^{\top}$, whose (i, j) entry measures the compatibility of tokens i and j through the query–key matrix $W_{\text{qry}}^h (W_{\text{key}}^h)^{\top} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, of rank at most d_h.

3 Sheaf Diffusion as a Structured Multi-Head Operator

Focusing on the most general case of a directed hypergraph cellular sheaf proposed in Mule et al. (2026), in this section we introduce a stream-major view exposing the sheaf operator as a bank of node operators indexed by pairs of stalk coordinates and use it to restate the corresponding diffusion layer in the additive multi-head form of the Transformer's MHA operator, with sheaf-generated node operators in place of attention matrices. All proofs are in Appendix I.

The stream-major view. Let Ω be an operator acting on a hypergraph signal with n nodes and stalk dimension d, such as $Q_{\vec{\mathcal{F}}}$, and consider the step $Z = \Omega X$ of Eq. 2 with $W_1^{(t)} = I_d$ and $W_2^{(t)} = I_c$. We regard $X \in \mathbb{C}^{nd \times c}$ as the node-major matricization of a tensor $\mathcal{X} \in \mathbb{C}^{n \times d \times c}$ indexed by node, stalk coordinate, and channel, and write $X_i := \mathcal{X}_{i::} \in \mathbb{C}^{d \times c}$ for the stalk at node i and $X^p := \mathcal{X}_{:p:} \in \mathbb{C}^{n \times c}$ for the p-th *stream*, i.e., the p-th stalk coordinate of all nodes. In the node-major view, $\Omega = [\Omega_{ij}]_{i,j \in V}$ has blocks $\Omega_{ij} \in \mathbb{C}^{d \times d}$ and $Z_i = \sum_j \Omega_{ij} X_j$: each block maps the stalk at node j into the stalk at node i. The *stream-major view* reorganizes the same tensor by stalk coordinate through a permutation matrix Π (a perfect shuffle, Van Loan, 2000):

$\Pi X = ((X^1)^\top, \dots, (X^d)^\top)^\top$ stacks the streams, and $\widehat{\Omega} = \Pi\Omega\Pi^\top = [\Omega^{ps}]_{p,s\in[d]}$ is a $d \times d$ block matrix of *stream blocks* $\Omega^{ps} \in \mathbb{C}^{n\times n}$ indexed by pairs of stalk coordinates, so that $\Pi Z = \widehat{\Omega}\,\Pi X$:

$$Z^p = \sum_{s=1}^{d} \Omega^{ps} X^s, \qquad p \in [d]. \tag{5}$$

While Ω_{ij} fixes a node pair and acts across stalk coordinates, $\Omega^{ps} \in \mathbb{C}^{n\times n}$ fixes a pair of stalk coordinates and acts across nodes: Ω^{pp} propagates within stream p, and Ω^{ps}, $p \neq s$, transfers information from stream s to stream p, so that each stream pair has its own $n \times n$ node operator, whose support is inherited from the topology: $\Omega^{ps}_{ij} = 0$ whenever Ω does not couple i and j. Finally, we write $\widehat{X} := [X^1 \ \cdots \ X^d] \in \mathbb{C}^{n\times dc}$ for the *stream representation* of the signal, whose i-th row is $\mathrm{rvec}(X_i)$, the row-wise vectorization (Henderson & Searle, 1981) of the stalk at node i: $\widehat{X}$ has the shape of a Transformer signal, with d as the number of heads and c as the per-head dimension. A hat on a signal always denotes this $n \times dc$ representation.

Toward the diagonal case. For the diffusion operator of Mule et al. (2026), Eq. 1 gives the factorization $Q_{\vec{\mathcal{F}}} = B^{(q)*}B^{(q)}$ through the directed sheaf boundary operator $B^{(q)} \in \mathbb{C}^{md\times nd}$, whose rows are indexed by hyperedge–stalk pairs. Applying to $B^{(q)}$ a permutation Π_E of the hyperedge–stalk coordinates analogous to Π, we define $\widehat{B} := \Pi_E B^{(q)} \Pi^\top = [B^{rs}]_{r,s\in[d]}$, $B^{rs} \in \mathbb{C}^{m\times n}$, with $(B^{rs})_{ei} = \delta_e^{-1/2} S^{(q)}_{i\lhd e}(F_{i\lhd e})_{rs}$. Each B^{rs} maps the node signal in stream s to the hyperedge signal in stream r through the (r,s) entries of the restriction maps, and $\widehat{Q_{\vec{\mathcal{F}}}} = \widehat{B}^*\widehat{B}$. Taking the (p,s) block gives the stream blocks of $Q_{\vec{\mathcal{F}}}$ as $Q^{ps} = \sum_{r=1}^{d} (B^{rp})^* B^{rs}$: in general, the d^2 stream operators Q^{ps} are jointly generated by the same family of incidence operators B^{rs}, so that a full sheaf induces a coupled family of node operators $Q^{ps} \in \mathbb{C}^{n\times n}$ sharing the same topology. In the diagonal case, the incidence operators B^{rs} with $r \neq s$ vanish: the streams decouple and each is governed by its own scalar sheaf Laplacian, so that the sheaf operator is a bank of independent heads:

Proposition 1. *With diagonal restriction maps $F_{i\lhd e} = \mathrm{diag}(F^1_{i\lhd e}, \dots, F^d_{i\lhd e})$, $F^p_{i\lhd e} \in \mathbb{R}$, in the stream-major view the directed sheaf hypergraph Laplacian decomposes as the direct sum*

$$\widehat{L_{\vec{\mathcal{F}}}} = \bigoplus_{p=1}^{d} L^p, \qquad L^p = D^p - (B^p)^* B^p \in \mathbb{C}^{n\times n}, \tag{6}$$

where $B^p := B^{pp}$ is the p-th diagonal block of $\widehat{B}$, with $(B^p)_{ei} = \delta_e^{-1/2} S^{(q)}_{i\lhd e} F^p_{i\lhd e}$, and $D^p_{ii} = \sum_{e\ni i} (F^p_{i\lhd e})^2$; likewise $\widehat{Q_{\vec{\mathcal{F}}}} = \bigoplus_{p=1}^{d} (B^p)^* B^p$, and analogously for the normalized operators whenever every $D^p_{ii} > 0$.

Each L^p is thus a one-dimensional directed sheaf hypergraph Laplacian on the same hypergraph, with the same directional phases $S^{(q)}_{i\lhd e}$ but its own scalar restriction maps $F^p_{i\lhd e}$: diagonal sheaves admit a *multi-head Laplacian* interpretation in which all streams share topology and phases while each learns its own node Laplacian, as heads in MHA learn different weights over the same support (Veličković et al., 2018; Brody et al., 2022). We call a stream p of a diagonal sheaf, governed by its one-dimensional operator L^p, a *scalar stream*.

Layer-map correspondence. We now consider the linear part (pre-activation) of the layer defined in Mule et al. (2026), associated with the generic propagation operator Ω introduced above. Suppressing the layer index for readability, we consider

$$Z = \Omega(I_n \otimes W_1) X W_2, \tag{7}$$

where $Z \in \mathbb{C}^{nd\times c'}$ denotes the resulting linear pre-activation. The following theorem shows the central relationship between Eq. 7 and the Transformer MHA of Eq. 4:

Theorem 1. *Letting $\widehat{\Omega} = \Pi\Omega\Pi^\top = [\Omega^{ps}]_{p,s\in[d]}$ be the stream-major view of Ω and letting $\widehat{X} \in \mathbb{C}^{n\times dc}$ and $\widehat{Z} \in \mathbb{C}^{n\times dc'}$ be the stream representations of input and output of Eq. 7, we have $\widehat{Z} = \sum_{p=1}^{d}\sum_{s=1}^{d} \Omega^{ps}\widehat{X} W^s_{\mathrm{val}} W^p_{\mathrm{out}}$ with $W^s_{\mathrm{val}} = (W_1^\top \mathbf{e}_s) \otimes I_c \in \mathbb{C}^{dc\times c}$ and $W^p_{\mathrm{out}} = \mathbf{e}_p^\top \otimes W_2 \in \mathbb{C}^{c\times dc'}$. If $\widehat{\Omega}$ is block diagonal in the stream index, so that $\Omega^{ps} = 0$ for $p \neq s$, this reduces to*

$$\widehat{Z} = \sum_{p=1}^{d} \Omega^{pp}\widehat{X} W^p_{\mathrm{val}} W^p_{\mathrm{out}}. \tag{8}$$

In particular, this holds for $\Omega = Q_{\vec{\mathcal{F}}}$ whenever all restriction maps are diagonal, by Proposition 1.

In the diagonal case the off-diagonal stream operators vanish and Eq. 8 has the same additive form as MHA's equation: nodes play the role of tokens, the d stalk coordinates play the role of heads, and Ω^{pp} occupies the position of the attention matrix $A^p(X)$ (Figure 1). The operator identity is exact, while the value and output maps and the node operators form a structured subclass of those of MHA. The value map $V^p := \widehat{X} W^p_{\mathrm{val}} = \sum_{r=1}^{d} (W_1)_{pr} X^r \in \mathbb{C}^{n \times c}$ mixes streams only through the scalar coefficients of W_1, while the same W_2 is used for every output stream. Whereas $A^p(X)$ is a dense row-stochastic operator obtained from query and key projections, Ω^{pp} is, by Proposition 1, the operator of a one-dimensional directed sheaf, supported on the underlying topology and built from restriction maps that depend on the current representation.

4 SHEAFORMER

The multi-head form of sheaf diffusion suggests two complementary modifications, which define *SheaFormer*: placing the bank of sheaf propagation operators inside a residual block with a normalized branch, so that diffusion updates the current representation rather than being the entire layer, and generating the restriction maps from stream-specific node–context bilinear compatibilities. The latter mirrors how MHA builds its attention matrices: by Theorem 1, Q^{pp} occupies the position of $A^p(X)$, which the p-th head computes from the query–key bilinear form of Eq. 4, whereas Q^{pp} is determined by the restriction-map coefficients $F^p_{i \lhd e}$ of stream p (Proposition 1); we therefore compute each $F^p_{i \lhd e}$ as a bilinear compatibility between the incident node, in the role of the query, and a representation of the hyperedge, in the role of the key.

A residual block for sheaf propagation. In DSHN, sheaf propagation *is* the layer,

$$X^{(t+1)} = \sigma\Big(Q^{(t)}_{\vec{\mathcal{F}}_N} (I_n \otimes W_1) X^{(t)} W_2 \Big). \tag{9}$$

By Theorem 1, for diagonal restriction maps its pre-activation is the multi-head sum of Eq. 8. In a Transformer, multi-head attention appears inside a residual sublayer rather than replacing the representation outright. SheaFormer adopts this residual placement, not the full Transformer block: it has no feed-forward sublayer, normalizes the propagated branch before the residual addition with the *complex whitening* batch normalization (Trabelsi et al., 2018) used in DSHN (Appendix D), and applies σ after the addition, as in the residual unit of He et al. (2016). For $\widehat{X} \in \mathbb{C}^{n \times dc}$, a SheaFormer layer computes

$$\widehat{X}^{(t+1)} = \sigma\left(\widehat{X}^{(t)} + \mathrm{Drop}\left[\mathrm{Norm}\left(\sum_{p=1}^{d} Q^{pp}_N\big(\widehat{X}^{(t)}\big)\, \widehat{X}^{(t)}\, W^p_{\mathrm{val}} W^p_{\mathrm{out}} \right) \right] \right). \tag{10}$$

Here $Q^{pp}_N(\widehat{X}^{(t)})$ is the p-th stream block of the normalized signless operator and $W^p_{\mathrm{val}}, W^p_{\mathrm{out}}$ are precisely the structured value and output maps of Theorem 1, so the propagated branch is the original sheaf layer expressed in stream coordinates.[1] Norm is the whitening batch normalization, applied once after diffusion: it centers each channel, whitens its real and imaginary parts by the inverse square root of their 2×2 covariance, and applies a learned 2×2 scale and a learned shift (Appendix D). Finally, Drop is dropout on the update and $\sigma(z) = z\,\mathbf{1}[\Re z \geq 0]$ is the complex ReLU.[2]

Complex whitening and the harmonic space. The harmonic space $\ker L$ is the fixed-point space of the diffusion operator $Q = I - L$, so diffusion applied to a harmonic signal will return that signal unchanged; since Norm acts after diffusion, it can still move such an update out of $\ker L$ before the residual addition. We now show that, beyond the learned stalk transformation W_1, through which neural sheaf diffusion already leaves $\ker L$ (Bodnar et al., 2022, Proposition 17) and which SheaFormer retains, the complex whitening adds two such mechanisms, absent from complex-linear propagation: centering and the conjugate-linear part of the learned 2×2 whitening, which acts on each channel as $z \mapsto \alpha z + \eta \overline{z}$ with $\alpha, \eta \in \mathbb{C}$ (Lemma 1).

[1] We share the restriction-map parameters across layers, so their number is independent of depth.

[2] The last layer omits both Drop and σ, and its output is normalized and mapped to the real domain through unwind before classification.

Theorem 2. *Let $L = L^* \succeq 0$ and $\mathcal{K} := \ker L$, with $x \in \mathcal{K}$ one channel of the current representation, $v \in \mathcal{K}$ the same channel of the propagated update in Eq. 10 before normalization, $m(v) := \frac{1}{nd}\mathbf{1}^* v$ its mean and $Pv = v - m(v)\mathbf{1}$ its centered version, where $P := I_{nd} - \frac{1}{nd}\mathbf{1}\mathbf{1}^*$ is the centering projection. With zero learned shift, the complex whitening acts as $\mathrm{Norm}(v) = \alpha(v)Pv + \eta(v)\overline{Pv}$ for coefficients $\alpha(v), \eta(v) \in \mathbb{C}$ (Lemma 1). For $y := x + \mathrm{Norm}(v)$: **(1)** if $\eta(v) = 0$ and $\alpha(v) \neq 0$, then $y \notin \mathcal{K}$ if and only if $m(v) \neq 0$ and $\mathbf{1} \notin \mathcal{K}$; **(2)** if $m(v) = 0$, then $y \notin \mathcal{K}$ if and only if $\eta(v) \neq 0$ and $\overline{v} \notin \mathcal{K}$.*

Role-aware bilinear restriction maps. The second step toward SheaFormer changes the *restriction generator*, i.e., the map Φ of Eq. 3 that generates the restriction maps. In DSHN, each restriction-map coefficient is computed by an MLP from a node representation and an orientation-independent hyperedge representation as in Eq. 3. We replace this generator by a role-aware bilinear compatibility: each stream projects the incident node and a role-aware representation of the hyperedge into a common c-dimensional space, and their bilinear compatibility becomes the corresponding restriction-map coefficient. Node i carries $X_i \in \mathbb{C}^{d\times c}$, whose stream representation is the i-th row $\widehat{X}_i = \mathrm{rvec}(X_i) \in \mathbb{C}^{1\times dc}$ of $\widehat{X}$. For an incidence $i \triangleleft e$, we represent the hyperedge from the viewpoint of i by keeping its head and tail contexts separate:

$$\widehat{H}^H_{i\triangleleft e} = \frac{1}{k_e}\sum_{j\in H(e)\setminus\{i\}} \widehat{X}_j; \qquad \widehat{H}^T_{i\triangleleft e} = \frac{1}{k_e}\sum_{j\in T(e)\setminus\{i\}} \widehat{X}_j, \tag{11}$$

with $k_e := \max\{1, \delta_e - 1\}$. With $\mathbf{u}_i := \mathrm{unwind}(\widehat{X}_i)$ and $\mathbf{h}_{i\triangleleft e} := [\mathrm{unwind}(\widehat{H}^H_{i\triangleleft e}) \| \,\mathrm{unwind}(\widehat{H}^T_{i\triangleleft e})]$, for each stream p the node and hyperedge context are then constructed through learned projections, $\mathbf{x}^p_i = \mathbf{u}_i W^p_{\mathrm{node}}$, and $\mathbf{h}^p_{i\triangleleft e} = \mathbf{h}_{i\triangleleft e} W^p_{\mathrm{edge}}$, where $W^p_{\mathrm{node}} \in \mathbb{R}^{2dc\times c}$ and $W^p_{\mathrm{edge}} \in \mathbb{R}^{4dc\times c}$ play the roles of query and key projections. Their inner product, scaled by $1/\sqrt{c}$, defines the scalar p-th restriction-map coefficient,

$$F^p_{i\triangleleft e} = \frac{\mathbf{x}^p_i(\mathbf{h}^p_{i\triangleleft e})^\top}{\sqrt{c}} = \frac{\mathbf{u}_i W^p_{\mathrm{node}}(W^p_{\mathrm{edge}})^\top \mathbf{h}^\top_{i\triangleleft e}}{\sqrt{c}}, \quad F_{i\triangleleft e} = \mathrm{diag}\left(F^1_{i\triangleleft e}, \ldots, F^d_{i\triangleleft e}\right) \in \mathbb{R}^{d\times d}. \tag{12}$$

The coefficient thus has the form of an entry of the argument of the softmax in Eq. 4: as the query–key matrix $W^h_{\mathrm{qry}}(W^h_{\mathrm{key}})^\top$ couples tokens i and j in the (i,j) entry of $XW^h_{\mathrm{qry}}(W^h_{\mathrm{key}})^\top X^\top/\sqrt{d_h}$, the matrix $W^p_{\mathrm{node}}(W^p_{\mathrm{edge}})^\top$ couples node i with the context of hyperedge e in Eq. 12, with the node in the role of the query and the hyperedge in the role of the key. We call $M^p := W^p_{\mathrm{node}}(W^p_{\mathrm{edge}})^\top/\sqrt{c} \in \mathbb{R}^{2dc\times 4dc}$ the *compatibility map* of stream p: it is shared by all incidences and has rank at most c. By Proposition 1, the resulting operator remains block diagonal in the stream index, with $Q^{pp} = (B^p)^* B^p$ and $(B^p)_{ei} = S^{(q)}_{i\triangleleft e} F^p_{i\triangleleft e}/\sqrt{\delta_e}$. Thus, the construction preserves the sheaf operator: the phase still encodes the head–tail role of the incident node, while the learned real restriction-map coefficient now also depends on how that node relates to the head and tail of the hyperedge; Appendix C details the resulting contribution of a single hyperedge to Q^{pp}.

Directional expressivity of the restriction maps. In DSHN, the real factor of the restriction maps is orientation-independent and direction enters only through the charge-dependent phase $S^{(q)}_{i\triangleleft e}$, which records the role of i itself. The result below shows that, while DSHN at zero charge $q = 0$ loses any orientation-related information, SheaFormer still produces different node representations:

Theorem 3. *For any hypergraph $\mathcal{H}$ with node features and orientation-independent hyperedge features and any two orientations $\vec{\mathcal{H}}_1$, $\vec{\mathcal{H}}_2$ of $\mathcal{H}$, DSHN with $q = 0$ produces identical node representations on $\vec{\mathcal{H}}_1$ and $\vec{\mathcal{H}}_2$ at every depth and for every parameter choice, so no permutation-invariant readout distinguishes them. In contrast, there exist $(\mathcal{H}, \vec{\mathcal{H}}_1, \vec{\mathcal{H}}_2)$ and SheaFormer parameters for which a single SheaFormer layer with $q = 0$ produces different representations on $\vec{\mathcal{H}}_1$ and $\vec{\mathcal{H}}_2$.*

Under identical features, the following theorem shows that the normalized operators of a scalar stream of SheaFormer strictly contain those of a scalar stream of DSHN with diagonal restriction maps, and separate orientations that DSHN cannot separate in one layer for any charge:

Theorem 4. *For simple directed graphs without isolated nodes, with identical nonzero node features and identical edge features, every normalized scalar diffusion operator realizable by a scalar stream of DSHN with diagonal restriction maps is realizable by a scalar stream of SheaFormer, while there*

are graphs where the converse fails. Moreover, two orientations of the same undirected graph exist whose one-layer representations under DSHN with diagonal restriction maps coincide for every charge and parameter choice, whereas one SheaFormer layer at $q = 0$ separates them for some parameter choice.

5 EXPERIMENTS

SheaFormer differs from DSHN in two separable components, the residual block of Eq. 10 and the restriction generator of Eq. 12. We evaluate these two components separately, with capacity-matched controls for the generator, before comparing the full model with state-of-the-art baselines. We address four questions: **(Q1)** How much does the residual block contribute? **(Q2)** How much do SheaFormer's restriction maps contribute? **(Q3)** How much of this gain does generator capacity alone explain? **(Q4)** How does SheaFormer compare with state-of-the-art methods?

Table 1: Decomposition of SheaFormer over the twelve benchmarks (accuracy points). Each row after the baselines introduces one controlled change. Δ_{arch} is reported relative to DSHN / DSHN-Light, respectively. Increments are means of per-dataset differences, rounded independently.

Model	Controlled change	Avg. acc.	Effect
DSHN	–	70.48	–
DSHN-Light	–	71.45	–
SheaFormer (DSHN restr.)	Residual block	71.56	$\Delta_{\mathrm{arch}} = +1.08/+0.11$
SheaFormer (MLP role-blind)	Generator capacity	72.19	$\Delta_{\mathrm{capacity}} = +0.63$
SheaFormer (MLP role-aware)	Role awareness	72.92	$\Delta_{\mathrm{role}} = +0.72$
SheaFormer	Bilinear compatibility	**73.48**	$\Delta_{\mathrm{bilinear}} = +0.56$

Setup. We evaluate on two groups of directed hypergraph benchmarks from Mule et al. (2026). The first comprises twelve node-classification benchmarks, ordered by clique-expansion homophily (CEHom), the node homophily of Pei et al. (2020) computed on the loop-free clique expansion. Each is built from a directed graph: every node u with at least one outgoing edge defines one hyperedge with tail $\{u\}$ and head its out-neighborhood $\mathcal{N}^+(u) \setminus \{u\}$. The second comprises two molecular-reaction hypergraphs, Molecular-1 and Molecular-2, where each reaction is a directed hyperedge whose tail is the set of reagents and whose head is the set of products; the task is hyperedge classification, with the final node representations summed over each hyperedge before the classifier. For the twelve node-classification benchmarks, we report mean and standard deviation over 10 runs under the split protocol of Mule et al. (2026), using their predefined splits where available and ten random $50/25/25$ splits otherwise. The molecular datasets use ten random $50/25/25$ splits of the hyperedges and are evaluated with the F1 score of Mule et al. (2026), the harmonic mean of macro-averaged precision and macro-averaged recall. Each run trains for at most 500 epochs with early stopping, and the hyperparameters of each model are selected by mean validation accuracy (F1 on the molecular datasets) over 150 Optuna trials per dataset. Further details are in Appendices E and F. The decomposition compares six models: the baselines **DSHN** and **DSHN-Light**, the latter without backpropagation through the operator construction;[3] **SheaFormer (DSHN restr.)**, the SheaFormer layer of Eq. 10 with the restriction generator of DSHN, an MLP on the node representation and the summed hyperedge representation $\mathbf{h}_e$ of Eq. 3; two MLP controls, **SheaFormer (MLP role-aware)** and **SheaFormer (MLP role-blind)**, defined below; and **SheaFormer**, the SheaFormer layer with role-aware bilinear restrictions. The role-aware MLP replaces the bilinear compatibility of Eq. 12 by a two-layer MLP on the same input $\mathbf{z}_{i\triangleleft e} := [\mathbf{u}_i \| \mathbf{h}_{i\triangleleft e}] \in \mathbb{R}^{1\times 6dc}$, i.e., the node and its head and tail contexts of Eq. 11, which computes $(F^1_{i\triangleleft e}, \dots, F^d_{i\triangleleft e}) = \mathrm{ReLU}(\mathbf{z}_{i\triangleleft e} W_a) W_b$ with $W_a \in \mathbb{R}^{6dc\times dc}$ and $W_b \in \mathbb{R}^{dc\times d}$ shared across layers. Since W_a has the shape of the d stacked projections W^p_{node} and W^p_{edge}, the two generators read the same input with the same number of parameters, up to the d^2c entries of W_b, and differ only in how the projections are combined (Appendix F). The role-blind MLP is the same network, in which $\mathbf{h}_{i\triangleleft e}$ is replaced by $[\mathrm{unwind}(\widehat{H}_{i\triangleleft e}) \| \mathrm{unwind}(\widehat{H}_{i\triangleleft e})]$, with $\widehat{H}_{i\triangleleft e} := \widehat{H}^H_{i\triangleleft e} + \widehat{H}^T_{i\triangleleft e} = k_e^{-1} \sum_{j\in e\setminus\{i\}} \widehat{X}_j$ the *role-blind context*, the mean of the other members

[3] All baseline numbers were obtained by re-running the public implementation of Mule et al. (2026); HGraphormer (Qu et al., 2026), absent from that suite, was adapted from the authors' public implementation and tuned under the same protocol over its authors' search space.

Table 2: Node classification accuracy (%, mean ± std). **First**, second and third best per dataset, ties on the mean broken by the lower standard deviation. Avg. is the mean over the twelve datasets; references in Appendix E.

Model CEHom	Roman-empire 0.0502	Squirrel 0.1889	Chameleon 0.2574	email-EU 0.2581	Telegram 0.2775	Cornell 0.3139
HGNN	59.95 ± 0.30	33.96 ± 1.81	37.25 ± 3.69	48.54 ± 2.73	54.81 ± 3.87	51.62 ± 7.09
HNHN	72.15 ± 0.26	36.30 ± 1.41	33.62 ± 3.83	29.07 ± 3.19	47.50 ± 13.55	50.81 ± 10.17
UniGCNII	78.77 ± 0.51	37.48 ± 2.08	38.95 ± 4.14	44.21 ± 2.58	55.38 ± 7.13	69.73 ± 5.38
LEGCN	65.59 ± 0.49	38.59 ± 2.12	39.61 ± 4.75	31.62 ± 3.96	40.77 ± 8.80	72.16 ± 6.74
HyperND	68.21 ± 0.51	38.96 ± 2.03	41.97 ± 3.96	30.77 ± 2.37	42.12 ± 4.42	74.05 ± 2.48
AllDeepSets	79.81 ± 0.52	40.23 ± 1.64	41.30 ± 4.09	34.86 ± 5.05	43.27 ± 8.86	75.14 ± 3.38
AllSetTransformer	81.80 ± 0.55	39.53 ± 1.40	40.31 ± 3.25	44.33 ± 8.94	66.73 ± 6.32	75.14 ± 2.65
ED-HNN	88.87 ± 0.32	39.92 ± 2.10	44.94 ± 2.19	63.97 ± 3.44	55.00 ± 6.67	74.05 ± 4.86
SheafHyperGNN	78.33 ± 0.49	41.97 ± 2.24	37.62 ± 2.31	46.56 ± 4.24	67.31 ± 4.63	73.78 ± 4.02
PhenomNN	71.87 ± 0.49	38.67 ± 2.24	43.64 ± 3.22	37.09 ± 2.82	49.81 ± 6.05	76.49 ± 3.64
HGraphormer	80.45 ± 0.90	41.22 ± 2.26	39.75 ± 4.88	48.50 ± 3.36	70.77 ± 7.07	74.86 ± 3.83
GeDi-HNN	81.65 ± 0.48	40.60 ± 1.40	43.00 ± 4.22	47.49 ± 3.52	77.88 ± 5.72	**77.84 ± 2.65**
DHGNN	21.62 ± 1.48	36.07 ± 2.31	40.05 ± 2.57	62.39 ± 2.59	79.81 ± 5.59	44.86 ± 7.47
DHGNN (w/ emb.)	76.75 ± 0.50	38.78 ± 1.63	41.26 ± 2.07	33.00 ± 3.26	78.65 ± 4.25	74.59 ± 5.01
DSHN	87.80 ± 0.62	41.97 ± 2.52	45.67 ± 3.42	73.68 ± 3.18	78.65 ± 4.59	72.70 ± 2.55
DSHN-Light	87.74 ± 0.55	43.29 ± 2.54	44.73 ± 3.25	75.43 ± 3.16	81.15 ± 2.69	74.59 ± 4.22
SheaFormer	**91.07 ± 0.42**	**45.22 ± 2.29**	**47.46 ± 4.10**	**81.54 ± 2.42**	**84.42 ± 4.75**	77.30 ± 4.07

Model CEHom	email-Enron 0.3173	Wisconsin 0.3423	Amazon-ratings 0.3711	Texas 0.3949	Citeseer 0.7155	Cora 0.7810	Avg. –
HGNN	48.38 ± 9.47	56.27 ± 5.95	45.72 ± 0.68	56.49 ± 6.99	76.88 ± 1.24	**87.84 ± 0.79**	54.81
HNHN	39.19 ± 7.76	58.63 ± 3.45	41.42 ± 0.47	65.14 ± 6.45	70.66 ± 1.10	76.35 ± 2.10	51.74
UniGCNII	47.57 ± 8.73	85.88 ± 3.70	49.08 ± 0.47	79.73 ± 3.68	**77.22 ± 1.11**	87.03 ± 0.86	62.59
LEGCN	31.08 ± 8.98	83.53 ± 3.06	47.18 ± 0.49	75.95 ± 8.06	72.51 ± 1.09	74.76 ± 1.56	56.11
HyperND	31.35 ± 3.86	85.88 ± 4.54	46.58 ± 0.77	82.70 ± 5.69	74.90 ± 1.46	77.95 ± 1.33	57.95
AllDeepSets	35.68 ± 7.53	**88.24 ± 4.64**	50.43 ± 0.88	79.73 ± 7.48	75.98 ± 1.40	86.59 ± 0.68	60.94
AllSetTransformer	51.62 ± 8.15	85.69 ± 3.51	49.75 ± 1.22	81.62 ± 3.78	75.07 ± 1.07	86.54 ± 0.94	64.84
ED-HNN	46.76 ± 8.64	82.16 ± 4.15	47.73 ± 0.75	78.65 ± 5.98	75.85 ± 1.17	86.38 ± 0.62	65.36
SheafHyperGNN	60.00 ± 4.80	83.33 ± 4.58	46.86 ± 0.95	75.95 ± 6.33	76.21 ± 1.41	87.34 ± 1.31	64.61
PhenomNN	46.49 ± 8.09	84.71 ± 4.19	48.37 ± 0.43	80.00 ± 4.05	77.09 ± 1.20	87.37 ± 1.10	61.80
HGraphormer	58.11 ± 4.46	85.49 ± 5.33	47.31 ± 0.69	78.65 ± 5.76	76.49 ± 1.39	87.47 ± 1.20	65.76
GeDi-HNN	45.68 ± 5.85	84.71 ± 3.14	48.31 ± 0.47	81.35 ± 3.30	74.99 ± 1.08	82.85 ± 1.27	65.53
DHGNN	62.70 ± 7.03	58.82 ± 5.88	**53.61 ± 0.64**	58.92 ± 7.43	67.16 ± 1.61	83.44 ± 1.00	55.79
DHGNN (w/ emb.)	50.27 ± 6.42	86.08 ± 4.59	52.14 ± 0.49	81.62 ± 6.02	75.74 ± 0.99	85.88 ± 0.91	64.56
DSHN	67.30 ± 3.91	86.47 ± 5.15	48.78 ± 0.40	79.19 ± 5.55	76.14 ± 1.15	87.42 ± 0.76	70.48
DSHN-Light	65.14 ± 4.75	85.88 ± 5.46	50.76 ± 0.55	**84.59 ± 4.02**	76.74 ± 1.08	87.39 ± 0.90	71.45
SheaFormer	**72.97 ± 6.24**	86.67 ± 2.89	49.16 ± 0.62	82.97 ± 5.41	76.17 ± 1.17	86.82 ± 0.58	**73.48**

Table 3: Hyperedge classification F1 on the molecular-reaction hypergraphs (%, mean ± std; harmonic mean of macro precision and macro recall), against the three strongest baselines per dataset; all methods in Table 8 (Appendix F). Colors as in Table 2; OOM: out of memory.

Dataset	ED-HNN	PhenomNN	GeDi-HNN	DSHN	DSHN-Light	SheaFormer
Molecular-1	82.01 ± 0.55	63.14 ± 1.59	82.36 ± 0.58	OOM	83.49 ± 2.37	**87.59 ± 0.52**
Molecular-2	85.83 ± 2.24	85.94 ± 1.93	84.60 ± 2.94	86.07 ± 4.46	85.95 ± 3.81	**87.76 ± 2.67**

of e, which ignores which of them are heads and which are tails. Table 1 summarizes this controlled experiment, and Tables 5 and 7 (Appendix E) give the per-dataset values and the Spearman correlations ρ with CEHom and their 95% percentile-bootstrap intervals over datasets.

Q1. Keeping the restriction generator identical to DSHN's, the gain of adopting the SheaFormer layer of Eq. 10 as opposed to DSHN's layer of Eq. 9, $\Delta_{\text{arch}} := \text{Acc}(\text{SheaFormer (DSHN restr.)}) - \text{Acc}(\text{DSHN(-Light)})$, is $+1.08$ and $+0.11$ relative to DSHN and DSHN-Light, respectively, and this gain is strongly negatively associated with homophily: $\rho(\text{CEHom}, \Delta_{\text{arch}}) = -0.74$ (95% CI $[-0.95, -0.30]$) against DSHN and -0.84 ($[-0.97, -0.50]$) against DSHN-Light (Table 7). This gain therefore increases as CEHom decreases and is small or negative on the most homophilic benchmarks (Table 5). Since this ablation changes the residual formulation together with the placement of normalization and dropout, Δ_{arch} measures the effect of adopting the SheaFormer layer.

Q2. Replacing the DSHN generator with the proposed role-aware bilinear restrictions within the SheaFormer layer, $\Delta_{\text{restr}} := \text{Acc}(\text{SheaFormer}) - \text{Acc}(\text{SheaFormer (DSHN restr.)})$, improves accuracy on eleven of the twelve benchmarks, by $+1.92$ points on average, with no detectable association with CEHom ($\rho = +0.10$, 95% CI $[-0.50, +0.67]$; Table 7). Theorems 3 and 4 establish that exposing the head and tail roles to the restriction-map coefficients enables directional distinctions unavailable to the DSHN restriction maps, which depend on orientation only through the phase.

On Squirrel, email-EU, email-Enron and Cora, the selected charge is $q = 0$ for both SheaFormer (DSHN restr.) and SheaFormer. The corresponding values of Δ_{restr}, up to $+5.67$ points on email-EU and $+6.48$ on email-Enron (Table 5), are therefore obtained without the phase, in agreement with Theorem 3.

Q3. To separate capacity from the structural changes in our construction, we decompose $\Delta_{\mathrm{restr}} = \Delta_{\mathrm{capacity}} + \Delta_{\mathrm{role}} + \Delta_{\mathrm{bilinear}}$ along the two MLP controls, with $\Delta_{\mathrm{capacity}} := \mathrm{Acc}(\text{MLP role-blind}) - \mathrm{Acc}(\text{DSHN restr.})$, $\Delta_{\mathrm{role}} := \mathrm{Acc}(\text{MLP role-aware}) - \mathrm{Acc}(\text{MLP role-blind})$ and $\Delta_{\mathrm{bilinear}} := \mathrm{Acc}(\text{SheaFormer}) - \mathrm{Acc}(\text{MLP role-aware})$, where each name in parentheses denotes the corresponding SheaFormer variant. Both controls have substantially more capacity than the DSHN generator, with a parameter count growing as $\mathcal{O}(c^2 d^2)$ rather than $\mathcal{O}(c^2 + cd)$ (Appendix G). Holding the MLP architecture, parameter count and hyperparameter search fixed, exposing the head and tail contexts separately adds $+0.72$ (Δ_{role}), and replacing this MLP by the bilinear compatibility adds another $+0.56$ ($\Delta_{\mathrm{bilinear}}$). Of the total $+1.92$ points, only $+0.63$ ($\Delta_{\mathrm{capacity}}$) is recovered by the role-blind MLP; the remaining $+1.29$ comes from Δ_{role} and $\Delta_{\mathrm{bilinear}}$ (Table 1).

Q4. Against the baselines of Table 2, SheaFormer attains the highest average accuracy (73.48; $+2.03$ over DSHN-Light, $+3.00$ over DSHN). Its improvement over DSHN-Light grows as CEHom decreases ($\rho = -0.68$, 95% CI $[-0.92, -0.13]$; Table 7). On the two molecular-reaction hypergraphs, SheaFormer attains the highest F1 on both ($+4.10$ over DSHN-Light on Molecular-1, $+1.69$ over DSHN on Molecular-2; Table 3). Overall, SheaFormer ranks first on eight of the fourteen datasets, six node-classification benchmarks and both molecular datasets, and among the best two on eleven.

Analysis. We further analyze the trained SheaFormer models of Table 2. The compatibility map M^p is the analogue of the query–key matrix of an attention head, whose singular directions identify compact task-relevant subspaces (Sen Sharma et al., 2026; Ahmad et al., 2025), a reading that the MLP controls of Q3 do not admit; we therefore ask whether its nominal rank c is used after training. Truncating every M^p to at most 16 of its leading singular directions preserves accuracy within 0.5 points on ten of the twelve benchmarks, whereas the same number of random directions reduces it by 0.8 to 36.2 points. We also measure the two mechanisms of Theorem 2 in the trained models through the Rayleigh quotient $z^* L z / \|z\|^2$, which vanishes exactly on $\ker L$: centering moves near-harmonic updates away from $\ker L$ on email-EU and email-Enron, and the conjugate-linear whitening is the dominant mechanism on Texas, Cornell and Wisconsin (Appendices H.1 and H.2).

6 Conclusions

We derived, directly from the standard sheaf diffusion layer, an exact stream-major decomposition that reveals its additive multi-head structure: stalk coordinates index the heads, topology-constrained sheaf operators occupy the position of attention matrices, and the original stalk and channel transformations induce structured value and output maps; we used it to design *SheaFormer*, a sheaf hypergraph neural network with a residual block and role-aware bilinear restriction maps. Theoretically, the role-aware restrictions encode directionality in the learned sheaf geometry: they distinguish orientations at zero charge and, under identical features, realize a strictly larger family of normalized scalar diffusion operators than DSHN, yielding one-layer graph distinctions that no one-layer DSHN with diagonal restriction maps recovers for any charge; in addition, whitening the propagated branch before the residual addition provides two further mechanisms for leaving the harmonic space, beyond the stalk mixing of neural sheaf diffusion. Experimentally, the residual block and the restriction generator provide complementary gains: the gain of the former increases as homophily decreases, whereas the latter improves eleven of twelve benchmarks without a detectable association with homophily, and role awareness and the bilinear compatibility together account for $+1.29$ of its $+1.92$ points after generator capacity is controlled. Finally, SheaFormer attains the highest average accuracy, 2.03 points above the strongest baseline, and the highest F1 on both molecular-reaction hypergraphs.

To face the natural limitations of this research, as directions for future work we identify the generalization of our construction to sheaves over simplicial and more general cellular complexes, whose higher-dimensional cells introduce incidence structures absent from directed hypergraphs.

Reproducibility Statement

The architecture is fully specified in Section 4 and Appendices B, C and D, and the sheaf diffusion layers it builds on are restated in Appendix A. The assumptions of every theoretical result are given in its statement, and complete proofs are given in Appendix I, together with the full statements of Theorems 2 and 4. The construction of the directed hypergraphs from the source graphs, the splits, the homophily measure, the hyperparameter search space and the training protocol are described in Appendix E; the molecular-reaction data, the hyperedge-classification readout and the evaluation metric in Appendix F; and the re-run of the baselines in Section 5. Appendix G analyzes the computational cost, Appendix G.2 reports the measured cost, and Appendix H the trained models. Code will be released upon acceptance.

The Use of Large Language Models

Large language models (LLMs) were used in the following ways. *Writing:* to edit and polish the text and to draft passages of several sections from the authors' notes, results and theorem statements; every sentence was revised by the authors. *Retrieval and discovery:* to search for related work on sheaf neural networks, attention and their connections; every cited work was retrieved and read by the authors before citation. *Research ideation and execution:* to write and debug analysis scripts and parts of the experimental pipeline under the authors' direction; the design of SheaFormer, the choice of experiments and the interpretation of the results are the authors'. *Proving mathematical claims:* to check proofs and to suggest proof steps and counterexample constructions; every theorem statement and every step of every proof in Appendix I was verified by the authors, who are responsible for their correctness. All reported numbers were produced by the authors' code and checked against the raw logs; LLMs were not used to generate or select results.

References

Areeb Ahmad, Abhinav Joshi, and Ashutosh Modi. Beyond components: Singular vector-based interpretability of transformer circuits. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-5593.

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. Sheaf neural networks with connection Laplacians. In *Proceedings of Topological, Algebraic, and Geometric Learning Workshops 2022*, volume 196 of *Proceedings of Machine Learning Research*, 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 Paul Chamberlain, Pietro Liò, and Michael M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Rémi Bourgerie, Šarūnas Girdzijauskas, and Viktoria Fodor. Deep neural sheaf diffusion. In *ICML 2026 Workshop on Graph Foundation Models*, 2026. URL https://arxiv.org/abs/2605.19021.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are AllSet: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hpBTIv2uy_E.

Davide D'Elia. Thermodynamic constraints in transformer architectures: A sheaf-theoretic perspective. Zenodo, https://doi.org/10.5281/zenodo.18165365, 2026.

Yihe Dong, Will Sawin, and Yoshua Bengio. HNHN: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278*, 2020.

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

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558.

Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Alessio Del Bue. Let there be direction in hypergraph neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=h48Ri6pmvi.

Stefano Fiorini, Hakan Emre Aktas, Iulia Duta, Pietro Morerio, Alessio Del Bue, Pietro Liò, and Stefano Coniglio. Sheaves reloaded: A directional awakening. In *International Conference on Learning Representations*, volume 2026, pp. 5144–5168, 2026a.

Stefano Fiorini, Edoardo Coppola, and Pietro Liò. Benchmarking sheaf neural networks for inductive tasks, 2026b. URL https://arxiv.org/abs/2608.02558.

Mustafa Hajij, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John L. Davenport, Sheik Dawood, Balaji Cherukuri, Joseph G. Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, and Tolga Birdal. Copresheaf topological neural networks: A generalized deep learning framework. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-5009.

Jakob Hansen and Thomas Gebhart. Sheaf neural networks, 2020. URL https://arxiv.org/abs/2012.06333.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Harold V. Henderson and S. R. Searle. The vec-permutation matrix, the vec operator and Kronecker products: a review. *Linear and Multilinear Algebra*, 9(4):271–288, 1981. doi: 10.1080/03081088108817379. URL https://doi.org/10.1080/03081088108817379.

Jing Huang and Jie Yang. UniGNN: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.

Steven M. Kearnes, Michael R. Maser, Michael Wleklinski, Anton Kast, Abigail G. Doyle, Spencer D. Dreher, Joel M. Hawkins, Klavs F. Jensen, and Connor W. Coley. The Open Reaction Database. *Journal of the American Chemical Society*, 143(45):18820–18826, 2021. doi: 10.1021/jacs.1c09820. URL https://doi.org/10.1021/jacs.1c09820.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL https://arxiv.org/abs/1412.6980.

Binbin Lin, Wei Chen, Yalun Li, Wenxiao Wang, Jieping Ye, and Xiaofei He. From self-attention to connection Laplacian: A unified operator view of transformers, 2026. URL https://arxiv.org/abs/2607.10677.

Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

Jose Lugo-Martinez, Daniel Zeiberg, Thomas Gaudelet, Noël Malod-Dognin, Natasa Przulj, and Predrag Radivojac. Classification in biological networks with hypergraphlet kernels. *Bioinformatics*, 37(7):1000–1007, 2021. doi: 10.1093/bioinformatics/btaa768.

Zitong Ma, Wenbo Zhao, and Zhe Yang. Directed hypergraph representation learning for link prediction. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 3268–3276. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/ma24b.html.

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*, volume 2026, pp. 81483–81512, 2026.

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.

Shilin Qu, Weiqing Wang, Yuan-Fang Li, Xin Zhou, and Fajie Yuan. Hypergraph node representation learning with one-stage message passing. *Neural Networks*, 193:107973, 2026. doi: 10.1016/j.neunet.2025.107973.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Brandon J. Reizman, Yi-Ming Wang, Stephen L. Buchwald, and Klavs F. Jensen. Suzuki–Miyaura cross-coupling optimization enabled by automated feedback. *Reaction Chemistry & Engineering*, 1(6):658–666, December 2016. doi: 10.1039/c6re00153j.

Guillermo Restrepo. Spaces of mathematical chemistry. *Theory in Biosciences*, 143(4):237–251, 2024. doi: 10.1007/s12064-024-00425-4.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t. URL http://dx.doi.org/10.1021/ci100050t.

Arnab Sen Sharma, Giordano Rogers, Natalie Shapira, and David Bau. LLMs process lists with general filter heads. In *International Conference on Learning Representations*, volume 2026, pp. 63227–63263, 2026.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31613–31632, 2023.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.

Francesco Tudisco, Austin R. Benson, and Konstantin Prokopchik. Nonlinear higher-order label spreading. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2402–2413, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450035. URL https://doi.org/10.1145/3442381.3450035.

Charles F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1–2):85–100, 2000. doi: 10.1016/S0377-0427(00)00393-9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations (ICLR)*, 2023a.

Yuxin Wang, Quan Gan, Xipeng Qiu, Xuanjing Huang, and David Wipf. From hypergraph energy functions to hypergraph neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 35605–35623, 2023b.

Chaoqi Yang, Ruijie Wang, Shuochao Yao, and Tarek Abdelzaher. Semi-supervised hypergraph node classification on hypergraph line expansion. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 2352–2361, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557447. URL https://doi.org/10.1145/3511808.3557447.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

A Directed Sheaf Hypergraph Diffusion

Let $\mathcal{H} = (V, E)$ be a directed hypergraph with $|V| = n$ and $|E| = m$. Each hyperedge $e \in E$ has a tail set $T(e) \subseteq V$ and a head set $H(e) \subseteq V$; we write $e = T(e) \cup H(e)$ for its incident nodes and $\delta_e = |e|$ for its cardinality.

Let $d \in \mathbb{N}$ be the stalk dimension, and associate with every node and hyperedge a vector space isomorphic to $\mathbb{C}^d$. For each incidence $i \triangleleft e$, let $F_{i \triangleleft e} \in \mathbb{R}^{d \times d}$ be a real restriction map. Following Mule et al. (2026), directionality is introduced in the directed restriction map $\vec{F}_{i \triangleleft e} = S^{(q)}_{i \triangleleft e} F_{i \triangleleft e}$, which depends on a charge parameter $q \in \mathbb{R}$ and the complex scalar

$$S^{(q)}_{i \triangleleft e} = \begin{cases} 1, & i \in H(e), \\ e^{-2\pi \mathbf{i} q}, & i \in T(e), \\ 0, & i \notin e, \end{cases} \tag{13}$$

which captures the head–tail role of the incidence. When $q = 0$, the phase carries no orientation.

The directed sheaf boundary operator $B^{(q)} \in \mathbb{C}^{md \times nd}$, i.e., the co-boundary operator of the sheaf, is defined as the block matrix whose (e, i) block is $\delta_e^{-1/2} \vec{F}_{i \triangleleft e}$ when $i \in e$ and zero otherwise. The corresponding signless sheaf operator and directed sheaf Laplacian are

$$Q_{\vec{\mathcal{F}}} = B^{(q)*} B^{(q)}, \qquad L_{\vec{\mathcal{F}}} = D_{\text{nds}} - Q_{\vec{\mathcal{F}}}, \tag{14}$$

where $D_{\text{nds}} = \operatorname{diag}(D_1, \ldots, D_n)$ and $D_i = \sum_{e : i \in e} F^{\top}_{i \triangleleft e} F_{i \triangleleft e}, i \in V$. The phases disappear from D_{nds} because $|S^{(q)}_{i \triangleleft e}| = 1$ on every incidence.

Let c denote the feature dimension. Each node i carries a signal $X_i \in \mathbb{C}^{d \times c}$, and the node signals are stacked in node-major order into $X \in \mathbb{C}^{nd \times c}$. Sheaf diffusion originates from the continuous-time sheaf heat equation

$$\dot{X}(t) = -\Delta_{\mathcal{F}} X(t), \tag{15}$$

where $\Delta_{\mathcal{F}}$ denotes the sheaf Laplacian. Applying a forward Euler discretization with unit step size gives

$$X^{(t+1)} = X^{(t)} - \Delta_{\mathcal{F}} X^{(t)} = \big(I_{nd} - \Delta_{\mathcal{F}}\big) X^{(t)}. \tag{16}$$

Different sheaf neural-network architectures turn this basic diffusion step into different neural layers. Neural Sheaf Diffusion (NSD) (Bodnar et al., 2022) parameterizes the Laplacian update and applies the nonlinearity to this update before subtracting it from the current representation:

$$X^{(t+1)} = X^{(t)} - \sigma\Big(\Delta^{(t)}_{\mathcal{F}} (I_n \otimes W^{(t)}_1) X^{(t)} W^{(t)}_2\Big). \tag{17}$$

The Directed Sheaf Neural Network (DSNN) (Fiorini et al., 2026a) adopts the same residual form, using its directed sheaf Laplacian.

The Sheaf Hypergraph Networks of Duta et al. (2023) apply the nonlinearity after the complete discrete diffusion step,

$$X^{(t+1)} = \sigma\Big(\big(I_{nd} - \Delta^{(t)}_{\mathcal{F}}\big)(I_n \otimes W^{(t)}_1) X^{(t)} W^{(t)}_2\Big). \tag{18}$$

The Directional Sheaf Hypergraph Network (DSHN) (Mule et al., 2026) uses the same form. Writing

$$Q^{(t)}_{\vec{\mathcal{F}}_N} = I_{nd} - L^{(t)}_{\vec{\mathcal{F}}_N} \tag{19}$$

for its normalized signless directed sheaf hypergraph operator, the DSHN convolutional layer reads

$$X^{(t+1)} = \sigma\Big(Q^{(t)}_{\vec{\mathcal{F}}_N} (I_n \otimes W^{(t)}_1) X^{(t)} W^{(t)}_2\Big). \tag{20}$$

Although these formulations differ in the choice and placement of the diffusion operator, the nonlinearity, and the residual connection, the learnable matrices $W^{(t)}_1$ and $W^{(t)}_2$ play the same role. The matrix $W^{(t)}_1 \in \mathbb{C}^{d \times d}$ mixes stalk coordinates within each node, whereas $W^{(t)}_2 \in \mathbb{C}^{c \times c'}$ maps the c input feature channels to c' output feature channels and is shared across all stalk coordinates. Thus, c' denotes the output feature dimension.

B CONSTRUCTION OF THE RESTRICTION MAPS

In most SNN architectures, the restriction maps are generated from local features according to a shared rule. NSD (Bodnar et al., 2022) considers three parameterizations. For general restriction maps, the restriction associated with the incidence $i \triangleleft e$ with $e = \{i, j\}$ is constructed from the representations of its two endpoints as

$$F_{i \triangleleft e} = \Phi(X_i, X_j), \qquad \Phi(X_i, X_j) = \mathrm{reshape}_{d \times d}\left(\sigma(V[X_i \| X_j])\right), \tag{21}$$

where V is learnable and σ is a nonlinearity. The map Φ is not required to be symmetric, allowing the two incidences of the same edge to receive different restriction maps. For diagonal restriction maps, only d values are predicted:

$$F_{i \triangleleft e} = \mathrm{diag}(\mathbf{r}_{ij}), \qquad \mathbf{r}_{ij} = \sigma(V_{\mathrm{diag}}[X_i \| X_j]) \in \mathbb{R}^d. \tag{22}$$

For orthogonal restriction maps, only the parameters needed to construct an orthogonal matrix are produced via $\boldsymbol{\theta}_{ij} = \sigma(V_{\mathrm{orth}}[X_i \| X_j]) \in \mathbb{R}^{d(d-1)/2}$. The entries of $\boldsymbol{\theta}_{ij}$ are placed below the diagonal of a unit lower-triangular matrix and interpreted as Householder-reflection parameters:

$$\widetilde{M}_{ij} = I_d + \mathrm{tril}(\boldsymbol{\theta}_{ij}), \qquad F_{i \triangleleft e} = \mathrm{HH}(\widetilde{M}_{ij}) \in O(d), \tag{23}$$

where V_{orth} is learned and $\mathrm{tril}(\boldsymbol{\theta}_{ij})$ denotes the strictly lower-triangular matrix obtained by placing the entries of $\boldsymbol{\theta}_{ij}$ below the diagonal. The operator HH interprets column j of $\widetilde{M}_{ij}$ as a Householder vector $\mathbf{j}_j = (0, \ldots, 0, 1, \widetilde{M}_{j+1,j}, \ldots, \widetilde{M}_{d,j})^\top$ and constructs the reflection $R_j = I_d - 2\frac{\mathbf{j}_j \mathbf{j}_j^\top}{\mathbf{j}_j^\top \mathbf{j}_j}$. The restriction map is then the product $F_{i \triangleleft e} = R_1 \cdots R_{d-1}$, which is orthogonal because each R_j is orthogonal.

Barbero et al. (2022a) avoid learning the restrictions by backpropagation and instead derive orthogonal transport maps from the geometry of the node features. For each node i, let $i_1, \ldots, i_{N_i}$ denote its neighbors and form the centered neighborhood matrix $\widetilde{X}_i = [X_{i_1} - X_i, \ldots, X_{i_{N_i}} - X_i]$. A local tangent-space basis is obtained from $\widetilde{X}_i = U_i \Sigma_i V_i^\top$ by retaining the first d left singular vectors as the columns of O_i. For adjacent nodes i and j, the orthogonal transport is then obtained from

$$O_i^\top O_j = U \Sigma V^\top, \qquad O_{ij} = U V^\top \in O(d). \tag{24}$$

Unlike the preceding constructions, neither O_i nor O_{ij} is produced by a learned generator: once the node representations are fixed, both matrices are obtained deterministically through the two SVDs above.

For hypergraphs, Duta et al. (2023) replace the second endpoint in the graph-based construction with a representation of the incident hyperedge, which they construct from the incident node representations. Among the different aggregations they propose, which exhibit similar performance, the simplest one is

$$\mathbf{h}_e = \frac{1}{|e|} \sum_{j \in e} X_j, \qquad \boldsymbol{\theta}_{i,e} = \sigma(\mathrm{MLP}(X_i \| \mathbf{h}_e)) \in \mathbb{R}^{d^2}, \qquad F_{i \triangleleft e} = \mathrm{reshape}_{d \times d}(\boldsymbol{\theta}_{i,e}), \tag{25}$$

where MLP denotes a multilayer perceptron with a linear output layer, while σ is an additional output nonlinearity, chosen as either a sigmoid or a hyperbolic tangent. In the diagonal parameterization, which is the best-performing one in their experiments,

$$\boldsymbol{\theta}_{i,e} \in \mathbb{R}^d, \qquad F_{i \triangleleft e} = \mathrm{diag}(\boldsymbol{\theta}_{i,e}). \tag{26}$$

DSHN (Mule et al., 2026) retains the node–hyperedge construction for a real, orientation-independent restriction map. Since the node and hyperedge representations may be complex-valued, they are first mapped to real vectors through unwind, defined after Eq. 3. The restriction map is then constructed as

$$F_{i \triangleleft e} = \sigma\big(\mathrm{MLP}\big(\mathrm{unwind}(X_i) \| \,\mathrm{unwind}(\mathbf{h}_e)\big)\big) \in \mathbb{R}^{d \times d}, \tag{27}$$

where the MLP has a linear output layer and σ is an output nonlinearity. When explicit hyperedge features are unavailable, $\mathbf{h}_e$ is obtained by a mean or sum aggregation of the incident node features.

The construction is analogous to that used for directed graphs by Fiorini et al. (2026a), where the restriction map is instead generated from the two endpoint representations as $F_{i \triangleleft e} = \Phi(\mathrm{unwind}(X_i) \| \,\mathrm{unwind}(X_j))$, with unwind used to map complex-valued representations to the real domain before applying the generator. Directionality is then introduced separately by multiplying the real restriction map by a complex phase.

C STRUCTURE OF THE ROLE-AWARE BILINEAR RESTRICTION MAPS

We expand the role-aware bilinear restrictions of Section 4 to make explicit what a single hyperedge contributes to the operator, and in particular how the head and tail contexts of the hyperedge enter. Throughout, the stream p is fixed.

The common denominator $k_e = \max\{1, \delta_e - 1\}$ of the two contexts in Eq. 11 makes them a decomposition of the role-blind context of Section 5, $\widehat{H}_{i \triangleleft e} := \widehat{H}^H_{i \triangleleft e} + \widehat{H}^T_{i \triangleleft e} = \frac{1}{k_e} \sum_{j \in e \setminus \{i\}} \widehat{X}_j$, so the construction retains both role identity and relative role cardinality.

With $\mathbf{u}_i$ as in Section 4, split W^p_{edge} by role, $W^p_{\text{edge}} = \begin{bmatrix} W^{p,H}_{\text{edge}} \\ W^{p,T}_{\text{edge}} \end{bmatrix}$ with $W^{p,H}_{\text{edge}}, W^{p,T}_{\text{edge}} \in \mathbb{R}^{2dc \times c}$. Since unwind is linear, the projected context is

$$\mathbf{h}^p_{i \triangleleft e} = \frac{1}{k_e} \sum_{l \in H(e) \setminus \{i\}} \mathbf{u}_l \, W^{p,H}_{\text{edge}} + \frac{1}{k_e} \sum_{l \in T(e) \setminus \{i\}} \mathbf{u}_l \, W^{p,T}_{\text{edge}}. \tag{28}$$

Define the two *role kernels* and the corresponding pairwise compatibilities

$$M^{p,\rho} := \frac{1}{\sqrt{c}} \, W^p_{\text{node}} \big(W^{p,\rho}_{\text{edge}}\big)^\top \in \mathbb{R}^{2dc \times 2dc}, \qquad \beta^{p,\rho}(i,l) := \mathbf{u}_i \, M^{p,\rho} \, \mathbf{u}_l^\top, \tag{29}$$

for $\rho \in \{H, T\}$, so that $\beta^{p,H}(i,l)$ is the compatibility of node i with node l when l is read as a head of a hyperedge, and $\beta^{p,T}(i,l)$ when l is read as a tail. The two kernels are the two halves of the compatibility map M^p of Section 4, $[M^{p,H} \; M^{p,T}] = M^p$, and share the left factor W^p_{node}; hence $\operatorname{rank}[M^{p,H} \; M^{p,T}] \le c$. Substituting into Eq. 12,

$$F^p_{i \triangleleft e} = F^{p,H}_{i \triangleleft e} + F^{p,T}_{i \triangleleft e}, \qquad F^{p,\rho}_{i \triangleleft e} := \frac{1}{k_e} \sum_{l \in \rho(e) \setminus \{i\}} \beta^{p,\rho}(i,l), \tag{30}$$

where $\rho(e)$ stands for $H(e)$ or $T(e)$. The restriction-map coefficient of the incidence $i \triangleleft e$ is thus the role-specific compatibility contribution of i from the heads of e plus the corresponding contribution from the tails of e, with node i itself excluded. The two means use different kernels, so a node can be compatible with the heads of a hyperedge and not with its tails: the restriction maps are role-aware beyond the phase $S^{(q)}_{i \triangleleft e}$, which only records the role of i itself.

Single-hyperedge contribution. By Eq. 1 and Proposition 1, the (i,j) entry of $Q^{pp} = (B^p)^* B^p$ is a sum over the hyperedges shared by i and j, $Q^{pp}_{ij} = \sum_{e \ni i,j} Q^{pp,e}_{ij}$, with

$$\begin{aligned} Q^{pp,e}_{ij} &= \frac{\overline{S^{(q)}_{i \triangleleft e}} \, S^{(q)}_{j \triangleleft e}}{\delta_e} \, F^p_{i \triangleleft e} F^p_{j \triangleleft e} \\ &= \frac{e^{2\pi \mathbf{i} q \, (\mathbf{1}[i \in T(e)] - \mathbf{1}[j \in T(e)])}}{\delta_e} \sum_{\rho, \rho' \in \{H,T\}} F^{p,\rho}_{i \triangleleft e} \, F^{p,\rho'}_{j \triangleleft e}. \end{aligned} \tag{31}$$

The normalized block Q^{pp}_N of Eq. 10 rescales this entry by $(D^p_{ii} D^p_{jj})^{-1/2} > 0$, which changes neither its phase nor its sign. The phase depends only on the roles of i and j in e: it equals 1 when they play the same role and $e^{\pm 2\pi \mathbf{i} q}$ when a tail meets a head. The real, signed amplitude that multiplies this phase is, up to the factor $1/\delta_e$, the sum of four compatibility products, indexed by the sides of the hyperedge through which i and j read it,

$$\underbrace{F^{p,H}_{i \triangleleft e} \, F^{p,H}_{j \triangleleft e}}_{HH} + \underbrace{F^{p,H}_{i \triangleleft e} \, F^{p,T}_{j \triangleleft e}}_{HT} + \underbrace{F^{p,T}_{i \triangleleft e} \, F^{p,H}_{j \triangleleft e}}_{TH} + \underbrace{F^{p,T}_{i \triangleleft e} \, F^{p,T}_{j \triangleleft e}}_{TT}, \tag{32}$$

where, for instance, HT couples i and j through the compatibility of i with the heads of e and that of j with its tails. Expanding the means, each term is a sum over pairs of witnesses, $F^{p,\rho}_{i \triangleleft e} \, F^{p,\rho'}_{j \triangleleft e} = k_e^{-2} \sum_{l \in \rho(e) \setminus \{i\}} \sum_{m \in \rho'(e) \setminus \{j\}} \beta^{p,\rho}(i,l) \, \beta^{p,\rho'}(j,m)$, so that i and j are coupled through e to the extent that both are compatible with the same sides of the hyperedge, in the same or in opposite roles.

D WHITENING BATCH NORMALIZATION

D.1 THE WHITENING MAP

The normalization Norm in Eq. 10 is the complex batch normalization of Trabelsi et al. (2018), in the form used by DSHN (Mule et al., 2026): each feature channel is treated as a two-dimensional real signal, its real and imaginary parts, and whitened by the inverse square root of its 2×2 covariance, with statistics computed jointly over nodes and streams. Let $\widehat{U} \in \mathbb{C}^{n \times dc}$ be the output of the propagated branch. For a channel $\ell \in [c]$, collect its nd entries $u_i^{p\ell}$, $i \in V$, $p \in [d]$, in node-major order into $v_\ell \in \mathbb{C}^{nd}$, and stack their real and imaginary parts as

$$\mathbf{Z}_\ell := [\Re v_\ell \quad \Im v_\ell] \in \mathbb{R}^{nd \times 2}, \tag{33}$$

whose row indexed by (i, p) is $(\mathbf{v}_i^{p\ell})^\top$ with $\mathbf{v}_i^{p\ell} := (\Re u_i^{p\ell}, \Im u_i^{p\ell})^\top \in \mathbb{R}^2$. The per-channel mean and empirical covariance are

$$\boldsymbol{\mu}_\ell = \frac{1}{nd}\, \mathbf{Z}_\ell^\top \mathbf{1} \in \mathbb{R}^2, \qquad C_\ell^0 = \frac{1}{nd}\big(\mathbf{Z}_\ell - \mathbf{1}\boldsymbol{\mu}_\ell^\top\big)^\top \big(\mathbf{Z}_\ell - \mathbf{1}\boldsymbol{\mu}_\ell^\top\big) \in \mathbb{R}^{2\times 2}, \tag{34}$$

where $\mathbf{1} \in \mathbb{R}^{nd}$ is the all-ones vector; the diagonal entries of C_ℓ^0 are the empirical variances of $\Re v_\ell$ and $\Im v_\ell$, and its off-diagonal entry their empirical covariance. The whitening transform uses the regularized covariance

$$C_\ell = C_\ell^0 + \epsilon\big(1 + \operatorname{tr} C_\ell^0\big) I_2, \qquad \epsilon > 0. \tag{35}$$

Since $C_\ell^0 \succeq 0$ and the regularization term is strictly positive, $C_\ell \succ 0$, so its principal inverse square root $C_\ell^{-1/2}$ is always well-defined, including for constant or purely real-valued channels. The normalized entry is

$$\begin{bmatrix} \Re\Big(\mathrm{Norm}(\widehat{U})_i^{p\ell}\Big) \\ \Im\Big(\mathrm{Norm}(\widehat{U})_i^{p\ell}\Big) \end{bmatrix} = \Gamma_\ell C_\ell^{-1/2}\big(\mathbf{v}_i^{p\ell} - \boldsymbol{\mu}_\ell\big) + \boldsymbol{\beta}_\ell, \tag{36}$$

or, writing $w_\ell \in \mathbb{C}^{nd}$ for the channel-ℓ entries of $\mathrm{Norm}(\widehat{U})$ collected in the same order,

$$[\Re w_\ell \quad \Im w_\ell] = \big(\mathbf{Z}_\ell - \mathbf{1}\boldsymbol{\mu}_\ell^\top\big) C_\ell^{-1/2} \Gamma_\ell^\top + \mathbf{1}\boldsymbol{\beta}_\ell^\top. \tag{37}$$

Here $\Gamma_\ell \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{\beta}_\ell \in \mathbb{R}^2$ are learnable per-channel affine parameters, initialized to $\Gamma_\ell = I_2/\sqrt{2}$ and $\boldsymbol{\beta}_\ell = \mathbf{0}$. Before the affine transformation, the empirical covariance of the normalized real–imaginary pair is $C_\ell^{-1/2} C_\ell^0 C_\ell^{-1/2}$, and after the affine transformation at initialization, with $\Gamma_\ell = I_2/\sqrt{2}$, it is $\frac{1}{2} C_\ell^{-1/2} C_\ell^0 C_\ell^{-1/2}$. In the unregularized limit $\epsilon \to 0$ this converges to $I_2/2$ whenever C_ℓ^0 is positive definite, so the real and imaginary parts have variance $1/2$ each and the complex variable has unit total variance; for $\epsilon > 0$ the regularization introduces a corresponding shrinkage while keeping the transform well-defined for singular empirical covariance. Since training is full-batch and transductive, the statistics are computed from the current input at inference as well, and no running averages are kept. A normalization of the same type is applied once more to the output of the final layer, before unwind and the classifier.

D.2 REAL-LINEAR STRUCTURE OF THE WHITENING MAP AND DIRECTED HARMONIC ESCAPE

The map in Eq. 36 applies the same real 2×2 matrix $\Gamma_\ell C_\ell^{-1/2}$ to the pair $(\Re u, \Im u)$ of every entry of a channel. Conditional on the channel statistics, the per-entry transformation is therefore real-linear but, in general, not complex-linear: it need not commute with multiplication by $\mathbf{i}$.

Fix a channel ℓ and collect its nd entries in node-major order into $v \in \mathbb{C}^{nd}$. Define $P := I_{nd} - \frac{1}{nd}\mathbf{1}\mathbf{1}^*$ and $m(v) := \frac{1}{nd}\mathbf{1}^* v$, so that $Pv = v - m(v)\mathbf{1}$ is the centered channel; the complex number $m(v)$ corresponds to the two-dimensional mean $\boldsymbol{\mu}_\ell$. Let $\mathbf{Z}_c(v) := [\Re(Pv) \quad \Im(Pv)] \in \mathbb{R}^{nd\times 2}$ and

$$C^0(v) := \frac{1}{nd}\mathbf{Z}_c(v)^\top \mathbf{Z}_c(v), \qquad C(v) := C^0(v) + \epsilon\big(1 + \operatorname{tr} C^0(v)\big) I_2, \tag{38}$$

so that $C^0(v) = C_\ell^0$ and $C(v) = C_\ell$ for the channel under consideration. In this notation, Eq. 36 applies the real-linear map

$$A(v) := \Gamma_\ell C(v)^{-1/2} = \begin{pmatrix} a_{11}(v) & a_{12}(v) \\ a_{21}(v) & a_{22}(v) \end{pmatrix} \in \mathbb{R}^{2\times 2} \tag{39}$$

entrywise to Pv and adds the complex shift $\xi := \beta_{\ell,1} + \mathbf{i}\,\beta_{\ell,2} \in \mathbb{C}$, where $\boldsymbol{\beta}_\ell = (\beta_{\ell,1}, \beta_{\ell,2})^\top$. The matrix $A(v)$ depends on v through its channel covariance.

Lemma 1 (Real-linear form of complex whitening)**.** *For every fixed $v \in \mathbb{C}^{nd}$, the real-linear map represented by $A(v)$ admits the unique decomposition $z \mapsto \alpha(v)z + \eta(v)\overline{z}$, where*

$$\alpha(v) = \frac{a_{11}(v) + a_{22}(v)}{2} + \frac{\mathbf{i}}{2}\big(a_{21}(v) - a_{12}(v)\big), \quad \eta(v) = \frac{a_{11}(v) - a_{22}(v)}{2} + \frac{\mathbf{i}}{2}\big(a_{21}(v) + a_{12}(v)\big). \tag{40}$$

Consequently,

$$\mathrm{Norm}(v) = \xi\mathbf{1} + \alpha(v)Pv + \eta(v)\overline{Pv}, \tag{41}$$

and $\eta(v) = 0$ if and only if $A(v)$ is complex-linear. Suppose, in addition, that $\Gamma_\ell = \gamma I_2$ for some $\gamma \in \mathbb{R} \setminus \{0\}$; this includes pure covariance whitening ($\gamma = 1$) and the initialization ($\gamma = 1/\sqrt{2}$). Then

$$\eta(v) = 0 \quad \Longleftrightarrow \quad \frac{1}{nd}\sum_{j=1}^{nd} (Pv)_j^2 = 0, \tag{42}$$

i.e., under scalar affine scaling the conjugate-linear coefficient vanishes exactly when the empirical pseudo-variance of the centered channel vanishes. For a general learned $\Gamma_\ell \in \mathbb{R}^{2\times 2}$, the expressions in Eq. 40 still apply, but the equivalence in Eq. 42 need not hold.

Proof in Appendix I.

Since Γ_ℓ is initialized to $I_2/\sqrt{2}$, the scalar-affine case of Lemma 1 applies at initialization: $\eta(v)$ vanishes exactly on centered channels with zero empirical pseudo-variance. Once Γ_ℓ is learned, however, zero pseudo-variance alone no longer guarantees $\eta(v) = 0$: if $C^0(v)$ is isotropic, then $C(v)^{-1/2}$ is a scalar matrix, but a non-complex-linear Γ_ℓ can still produce $\eta(v) \neq 0$.

Charge conjugation. We now specialize the preceding decomposition to the directed sheaf Laplacian. Write L_q for the operator obtained at charge q from fixed real restriction maps $F_{i\lhd e}$. Since the restriction maps are real and $S_{i\lhd e}^{(-q)} = \overline{S_{i\lhd e}^{(q)}}$ by Eq. 13, the incidence matrices satisfy $B^{(-q)} = \overline{B^{(q)}}$: the directed sheaf operators at charge $-q$ are the entrywise conjugates of those at charge q. The same holds for the normalized operators, since the degree normalization is real and independent of the sign of the charge. In particular,

$$L_{-q} = \overline{L_q}. \tag{43}$$

Corollary 1 (Directed harmonic escape)**.** *Let $L_q = L_q^* \succeq 0$ be the directed sheaf Laplacian at charge q, and define $\mathcal{K}_q := \ker L_q$ and $\mathcal{E}_q(z) := z^* L_q z$. Assume that the real restriction-map coefficients do not depend on the charge, which enters only through the directional phases, so that $L_{-q} = \overline{L_q}$. Then*

$$\overline{\mathcal{K}_q} = \mathcal{K}_{-q}. \tag{44}$$

Let $x, v \in \mathcal{K}_q$, suppose that $m(v) = 0$ and $\xi = 0$, and set $y := x + \mathrm{Norm}(v)$. Then

$$\mathcal{E}_q(y) = |\eta(v)|^2 \mathcal{E}_q(\overline{v}) = |\eta(v)|^2 \mathcal{E}_{-q}(v). \tag{45}$$

Consequently, $\mathcal{E}_q(y) > 0 = \mathcal{E}_q(x)$ if and only if $\eta(v) \neq 0$ and $v \notin \mathcal{K}_{-q}$: the conjugate-linear component moves a q-harmonic branch along the direction $\overline{v}$, which is harmonic for the opposite charge $-q$ but not for the current charge q. If L_q is real, in particular at $q = 0$, then conjugation preserves $\mathcal{K}_q$ and this harmonic-escape mechanism vanishes.

Proof in Appendix I.

The assumptions $m(v) = 0$ and $\xi = 0$ isolate the conjugate-linear mechanism. In general, $\mathrm{Norm}(v) = \xi\mathbf{1} + \alpha(v)(v - m(v)\mathbf{1}) + \eta(v)(\overline{v} - \overline{m(v)}\mathbf{1}) = \alpha(v)v + \eta(v)\overline{v} + \zeta(v)\mathbf{1}$ with $\zeta(v) := \xi - \alpha(v)m(v) - \eta(v)\overline{m(v)}$. Thus, away from the centered zero-shift setting, normalization may also inject the constant direction $\mathbf{1}$, which need not be harmonic for a sheaf Laplacian.

Table 4: Statistics of the twelve benchmarks, computed with our data loader: number of nodes, input features, hyperedges, classes, and incidences $\nu = \sum_e |e|$; average and maximum hyperedge size; average node degree, i.e., number of hyperedges containing a node; CEHom as in Eq. 46; and, in the last column, the clique-expansion homophily reported by Mule et al. (2026), whose clique expansion retains the diagonal. Datasets are sorted by CEHom, most heterophilic first. "–" marks datasets without node features, where the node degree is used as a one-dimensional input feature.

Dataset	# Nodes	# Features	# Hyperedges	# Classes	ν	avg $\lvert e\rvert$	max $\lvert e\rvert$	avg $\lvert v\rvert$	CEHom	CEHom (DSHN)
Roman-empire	22,662	300	22,662	18	61,805	2.73	16	2.73	0.0502	0.2363
Squirrel	2,223	2,089	2,060	5	49,058	23.81	290	22.07	0.1889	0.2448
Chameleon	890	2,325	797	5	9,651	12.11	131	10.84	0.2574	0.3221
email-EU	986	–	787	10	34,123	43.36	294	34.61	0.2581	0.2608
Telegram	245	1	183	4	9,095	49.70	148	37.12	0.2775	0.2854
Cornell	183	1,703	96	5	391	4.07	94	2.14	0.3139	0.4200
email-Enron	143	–	139	7	2,722	19.58	78	19.03	0.3173	0.3251
Wisconsin	251	1,703	170	5	669	3.94	123	2.67	0.3423	0.4398
Amazon-ratings	24,492	300	24,456	5	137,732	5.63	6	5.62	0.3711	0.4460
Texas	183	1,703	110	5	419	3.81	105	2.29	0.3949	0.5049
Citeseer	3,312	3,703	1,951	6	6,542	3.35	100	1.98	0.7155	0.7947
Cora	2,708	1,433	1,565	7	6,994	4.47	167	2.58	0.7810	0.8035

E EXPERIMENTAL DETAILS

Datasets. Table 4 reports the statistics of the twelve benchmarks, computed with our data loader. Every benchmark is a directed hypergraph built from a directed graph as in Mule et al. (2026): each node u with at least one outgoing edge defines one hyperedge with tail $\{u\}$ and head its out-neighborhood $\mathcal{N}^+(u) \setminus \{u\}$. All counts and averages coincide with those reported by Mule et al. (2026); the only difference is in the homophily measure. Their clique-expansion homophily is computed on the clique expansion with its diagonal, so that every node counts as its own same-label neighbor, whereas the CEHom of Eq. 46 excludes it; we reproduce their values exactly when the diagonal is retained (last column of Table 4). The self-loop inflates the homophily of datasets with small hyperedges most (Roman-empire: 0.05 versus 0.24), and the resulting ordering differs from that of Mule et al. (2026) only in the position of Chameleon and in the relative order of Cornell and email-Enron.

Baselines. The undirected hypergraph methods of Table 2 are HGNN (Feng et al., 2019), HNHN (Dong et al., 2020), UniGCNII (Huang & Yang, 2021), LEGCN (Yang et al., 2022), HyperND (Tudisco et al., 2021), AllDeepSets and AllSetTransformer (Chien et al., 2022), ED-HNN (Wang et al., 2023a), SheafHyperGNN (Duta et al., 2023), the SHN of Section 1, and PhenomNN (Wang et al., 2023b); the directed ones are GeDi-HNN (Fiorini et al., 2024), DHGNN (Ma et al., 2024), with and without the node embeddings used by Mule et al. (2026), DSHN (Mule et al., 2026) and DSHN-Light, the variant of DSHN that does not backpropagate through the construction of the operator.

Protocol. All results are means and standard deviations over 10 runs under the split protocol of Mule et al. (2026): their predefined splits where provided (Roman-empire, Squirrel, Chameleon, Telegram, Cornell, Wisconsin, Amazon-ratings, Texas) and ten random 50/25/25 splits otherwise; each run trains for 500 epochs with early stopping (patience 200) and selects the model at the epoch of best validation accuracy. For each dataset and model, each Optuna trial is trained on the ten splits and hyperparameters are selected by mean validation accuracy across them, over 150 trials. The search space of SheaFormer and its variants is: stalk dimension $d \in \{1, 2, 4, 8, 16, 32\}$ and channel dimension c with $dc \in \{32, \dots, 2048\}$ (both on a power-of-two grid, $c \leq 512$); number of layers $L \in \{1, \dots, 5\}$; learning rate log-uniform in $[10^{-4}, 2 \cdot 10^{-2}]$; weight decay in $\{0, 5 \cdot 10^{-5}, 5 \cdot 10^{-4}, 5 \cdot 10^{-3}\}$; classifier dropout and branch dropout in $\{0, 0.1, \dots, 0.8\}$; charge $q \in \{0, 0.05, \dots, 0.25\}$. The classifier is a two-layer MLP with 64 hidden units. Adam (Kingma & Ba, 2015) is used throughout. The restriction-map coefficients enter the boundary operator without output nonlinearity and without an added identity term; the degree D^p_{ii} of each stream (Proposition 1) is floored at a small constant before $D^{-1/2}$ is formed, so a node whose coefficients vanish in a stream receives no propagated signal in that stream.

Homophily measure. CEHom is the node homophily of Pei et al. (2020) computed on the clique expansion of the hypergraph. Writing $e \subseteq V$ for the full node set of a hyperedge (heads and tails, roles ignored) and $\mathcal{N}(v) = \{u \neq v : \{u, v\} \subseteq e \text{ for some } e \in E\}$ for the set of nodes that share at

least one hyperedge with v,

$$\mathrm{CEHom} = \frac{1}{|V'|} \sum_{v \in V'} \frac{|\{u \in \mathcal{N}(v) : y_u = y_v\}|}{|\mathcal{N}(v)|}, \qquad V' = \{v \in V : \mathcal{N}(v) \neq \emptyset\}, \tag{46}$$

is the fraction of a node's clique-expansion neighbors that carry its label, averaged over the nodes that have at least one neighbor. Each pair of nodes is counted once regardless of the number of hyperedges it shares, and a node is not its own neighbor.

Table 5: Per-dataset decomposition (Section 5) of the improvement of SheaFormer over DSHN and DSHN-Light (accuracy points), with $\Delta_{\mathrm{restr}} = \Delta_{\mathrm{capacity}} + \Delta_{\mathrm{role}} + \Delta_{\mathrm{bilinear}}$. Datasets are sorted by CEHom, most heterophilic first.

	Roman-empire	**Squirrel**	**Chameleon**	**email-EU**	**Telegram**	**Cornell**
CEHom	0.0502	0.1889	0.2574	0.2581	0.2775	0.3139
Δ_{arch} vs. DSHN	$+3.21$	$+2.20$	$+0.71$	$+2.19$	$+5.58$	$+1.62$
Δ_{arch} vs. DSHN-Light	$+3.27$	$+0.88$	$+1.65$	$+0.44$	$+3.08$	-0.27
Δ_{restr}	$+0.06$	$+1.05$	$+1.08$	$+5.67$	$+0.19$	$+2.98$
$\Delta_{\mathrm{capacity}}$	-0.13	$+1.58$	$+1.43$	-0.53	$+0.77$	$+1.36$
Δ_{role}	$+0.92$	-0.69	-1.96	$+4.50$	$+2.50$	$+1.35$
$\Delta_{\mathrm{bilinear}}$	-0.73	$+0.16$	$+1.61$	$+1.70$	-3.08	$+0.27$
Δ_{total} vs. DSHN	$+3.27$	$+3.25$	$+1.79$	$+7.86$	$+5.77$	$+4.60$
Δ_{total} vs. DSHN-Light	$+3.33$	$+1.93$	$+2.73$	$+6.11$	$+3.27$	$+2.71$

	email-Enron	**Wisconsin**	**Amazon-ratings**	**Texas**	**Citeseer**	**Cora**
CEHom	0.3173	0.3423	0.3711	0.3949	0.7155	0.7810
Δ_{arch} vs. DSHN	-0.81	-0.39	$+1.02$	-0.27	-0.20	-1.88
Δ_{arch} vs. DSHN-Light	$+1.35$	$+0.20$	-0.96	-5.67	-0.80	-1.85
Δ_{restr}	$+6.48$	$+0.59$	-0.64	$+4.05$	$+0.23$	$+1.28$
$\Delta_{\mathrm{capacity}}$	0.00	-1.18	$+0.42$	$+1.62$	$+1.11$	$+1.11$
Δ_{role}	$+1.89$	$+0.59$	-0.53	$+0.54$	-0.15	-0.27
$\Delta_{\mathrm{bilinear}}$	$+4.59$	$+1.18$	-0.53	$+1.89$	-0.73	$+0.44$
Δ_{total} vs. DSHN	$+5.67$	$+0.20$	$+0.38$	$+3.78$	$+0.03$	-0.60
Δ_{total} vs. DSHN-Light	$+7.83$	$+0.79$	-1.60	-1.62	-0.57	-0.57

Table 6: Accuracy (%, mean $\pm$ std over 10 runs) of the six models of the decomposition; Avg. is the mean over the twelve benchmarks. DSHN and DSHN-Light report our re-run under the protocol of Section 5.

Model	**Roman-empire**	**Squirrel**	**Chameleon**	**email-EU**	**Telegram**	**Cornell**
DSHN	87.80 ± 0.62	41.97 ± 2.52	45.67 ± 3.42	73.68 ± 3.18	78.65 ± 4.59	72.70 ± 2.55
DSHN-Light	87.74 ± 0.55	43.29 ± 2.54	44.73 ± 3.25	75.43 ± 3.16	81.15 ± 2.69	74.59 ± 4.22
SheaFormer (DSHN restr.)	91.01 ± 0.34	44.17 ± 2.41	46.38 ± 4.37	75.87 ± 2.79	84.23 ± 5.72	74.32 ± 4.46
SheaFormer (MLP role-blind)	90.88 ± 0.35	45.75 ± 2.58	47.81 ± 4.16	75.34 ± 3.44	85.00 ± 5.79	75.68 ± 5.10
SheaFormer (MLP role-aware)	91.80 ± 0.44	45.06 ± 1.32	45.85 ± 2.98	79.84 ± 2.50	87.50 ± 4.64	77.03 ± 2.92
SheaFormer	91.07 ± 0.42	45.22 ± 2.29	47.46 ± 4.10	81.54 ± 2.42	84.42 ± 4.75	77.30 ± 4.07

Model	**email-Enron**	**Wisconsin**	**Amazon-ratings**	**Texas**	**Citeseer**	**Cora**	**Avg.**
DSHN	67.30 ± 3.91	86.47 ± 5.15	48.78 ± 0.40	79.19 ± 5.55	76.14 ± 1.15	87.42 ± 0.76	70.48
DSHN-Light	65.14 ± 4.75	85.88 ± 5.46	50.76 ± 0.55	84.59 ± 4.02	76.74 ± 1.08	87.39 ± 0.90	71.45
SheaFormer (DSHN restr.)	66.49 ± 4.45	86.08 ± 6.76	49.80 ± 0.59	78.92 ± 6.47	75.94 ± 1.49	85.54 ± 0.95	71.56
SheaFormer (MLP role-blind)	66.49 ± 7.12	84.90 ± 3.59	50.22 ± 0.54	80.54 ± 5.67	77.05 ± 1.28	86.65 ± 0.74	72.19
SheaFormer (MLP role-aware)	68.38 ± 4.42	85.49 ± 4.15	49.69 ± 0.43	81.08 ± 5.70	76.90 ± 1.05	86.38 ± 1.30	72.92
SheaFormer	72.97 ± 6.24	86.67 ± 2.89	49.16 ± 0.62	82.97 ± 5.41	76.17 ± 1.17	86.82 ± 0.58	73.48

Table 7: Spearman correlation ρ between CEHom and the per-dataset cumulative improvement of each model of Table 1 over DSHN and DSHN-Light, and between CEHom and the restriction increment Δ_{restr}, which does not depend on the baseline, with 95% percentile-bootstrap confidence intervals. Each ρ is computed from twelve points, the CEHom of a dataset and its improvement Δ in Table 5; it is recomputed on $20{,}000$ resamples of the twelve datasets drawn with replacement, and the interval spans the 2.5th to the 97.5th percentile of the resampled values.

Model	ρ vs. DSHN [95% CI]	ρ vs. DSHN-Light [95% CI]
SheaFormer (DSHN restr.)	-0.74 [-0.95, -0.30]	-0.84 [-0.97, -0.50]
SheaFormer (MLP role-blind)	-0.72 [-0.94, -0.33]	-0.74 [-0.93, -0.32]
SheaFormer (MLP role-aware)	-0.56 [-0.89, $+0.01$]	-0.69 [-0.89, -0.22]
SheaFormer	-0.50 [-0.91, $+0.18$]	-0.68 [-0.92, -0.13]
Δ_{restr}	$+0.10$ [-0.50, $+0.67$]	

F Additional experiments

This appendix details the two MLP controls used in the decomposition of Section 5 (Q3) and reports hyperedge classification on two molecular-reaction hypergraphs (Table 8).

MLP controls. The role-aware MLP control of Section 5 computes $(F^1_{i\triangleleft e}, \dots, F^d_{i\triangleleft e}) = \mathrm{ReLU}(\mathbf{z}_{i\triangleleft e} W_a) W_b$ from $\mathbf{z}_{i\triangleleft e} = [\mathbf{u}_i \| \mathbf{h}_{i\triangleleft e}] \in \mathbb{R}^{1\times 6dc}$, with $W_a \in \mathbb{R}^{6dc\times dc}$ and $W_b \in \mathbb{R}^{dc\times d}$ shared across layers, as in SheaFormer. The hidden width dc gives W_a exactly the shape and number of parameters of the stacked node and context projections of the bilinear generator: partitioning

$$W_a = \begin{bmatrix} W_{a,\text{node}} \\ W_{a,\text{edge}} \end{bmatrix}, \qquad W_{a,\text{node}} \in \mathbb{R}^{2dc\times dc}, \quad W_{a,\text{edge}} \in \mathbb{R}^{4dc\times dc}, \tag{47}$$

and each block into d groups of c columns yields d node and d context projections of the same dimensionality as W^p_{node} and W^p_{edge}. The two generators thus project the same input to the same total width and differ in how the projections are combined: SheaFormer contracts corresponding c-dimensional node and context projections into a scaled inner product, whereas the control adds them through W_a, applies a ReLU and uses the linear readout W_b. At equal d and c they have $6(dc)^2$ and $6(dc)^2 + d^2c$ parameters, so the control is parameter-matched up to W_b. The role-blind control keeps this MLP unchanged, including its parameter count, and erases only the role information: $\mathbf{h}_{i\triangleleft e}$ is replaced by $[\mathrm{unwind}(\widehat{H}_{i\triangleleft e}) \| \mathrm{unwind}(\widehat{H}_{i\triangleleft e})]$, with $\widehat{H}_{i\triangleleft e} = \widehat{H}^H_{i\triangleleft e} + \widehat{H}^T_{i\triangleleft e} = k_e^{-1} \sum_{j\in e\setminus\{i\}} \widehat{X}_j$ the role-blind context, so the two $2dc \times dc$ row blocks of $W_{a,\text{edge}}$, which read the head and the tail context, act on it through their sum, and the only difference between the two MLP controls is whether head and tail membership is visible to the generator.

Both MLP controls are tuned under the protocol and search space of Appendix E; their accuracies are in Table 6, whose per-dataset differences give the Δ_{capacity}, Δ_{role} and Δ_{bilinear} rows of Table 5. On average the SheaFormer (DSHN restr.) model and the two MLP controls lie between the baselines and SheaFormer (71.56, 72.19 and 72.92 against 71.45 for DSHN-Light and 73.48 for SheaFormer), and the role-aware MLP attains the highest mean accuracy of all methods of Table 2 on Roman-empire and Telegram. The three increments are not uniform across benchmarks: Δ_{role} exceeds $+1.3$ points on email-EU, Telegram, email-Enron and Cornell and is negative on five benchmarks, on four of which (Chameleon, Squirrel, Amazon-ratings and Citeseer) the role-blind MLP is the most accurate of the four generators; Δ_{bilinear} exceeds $+1.6$ points on email-Enron, Texas, email-EU and Chameleon; and Δ_{capacity} lies within ± 1.7 points on every benchmark and averages $+0.63$. Amazon-ratings remains the single benchmark on which the proposed restrictions do not improve over the DSHN generator.

Table 8: Hyperedge classification on the molecular-reaction hypergraphs (F1 as the harmonic mean of macro precision and macro recall, %, mean ± std over 10 runs). Baselines are our re-run of the public implementation of Mule et al. (2026) under the protocol of Section 5, with HGraphormer adapted from the authors' implementation and tuned under the same protocol; OOM: out of memory. **First**, second and third best per dataset; ties on the mean are broken by the lower standard deviation.

Model	Molecular-1	Molecular-2
HGNN	69.38 ± 0.55	81.55 ± 5.12
HNHN	31.13 ± 1.14	56.34 ± 12.19
UniGCNII	71.80 ± 0.71	85.92 ± 2.40
LEGCN	OOM	83.42 ± 3.35
HyperND	45.52 ± 0.83	83.44 ± 1.91
AllDeepSets	79.03 ± 0.41	85.48 ± 2.44
AllSetTransformer	79.79 ± 0.58	85.25 ± 4.05
ED-HNN	82.01 ± 0.55	85.83 ± 2.24
SheafHyperGNN	57.99 ± 2.75	80.25 ± 2.20
PhenomNN	63.14 ± 1.59	85.94 ± 1.93
HGraphormer	74.81 ± 2.30	85.26 ± 2.32
GeDi-HNN	82.36 ± 0.58	84.60 ± 2.94
DHGNN	OOM	85.79 ± 2.01
DSHN	OOM	86.07 ± 4.46
DSHN-Light	83.49 ± 2.37	85.95 ± 3.81
SheaFormer	$\mathbf{87.59 \pm 0.52}$	$\mathbf{87.76 \pm 2.67}$

Molecular reaction hypergraphs. We further evaluate SheaFormer on the two real-world directed hypergraphs of molecular reactions of Mule et al. (2026), cast as a hyperedge classification task. The data merge reactions from Kearnes et al. (2021), Reizman et al. (2016) and Lugo-Martinez et al. (2021), modeled as directed hypergraphs following Restrepo (2024): each hyperedge is a reaction, with the set of reagents as its tail and the set of products as its head, so the direction is intrinsic to the data. Molecular-1 has 100,523 nodes, 50,016 hyperedges and 10 classes; Molecular-2 has 956 nodes, 3,021 hyperedges and 6 classes. Node features are Morgan fingerprints (Rogers & Hahn, 2010). To classify hyperedges, the node representations produced by the last layer are summed over the members of each hyperedge before the classifier: with $X_{\text{node}} \in \mathbb{R}^{n \times c_{\text{out}}}$ the final node features and $\mathbf{J} \in \{0,1\}^{n \times m}$ the node–hyperedge incidence matrix, $X_{\text{edge}} = \mathbf{J}^\top X_{\text{node}} \in \mathbb{R}^{m \times c_{\text{out}}}$ is fed to the classifier, which outputs one score per class for each of the m hyperedges. Since the classes are imbalanced we report the F1 score used by Mule et al. (2026), the harmonic mean of the macro-averaged precision and the macro-averaged recall over the classes, which differs from the mean of the per-class F1 scores, as mean and standard deviation over 10 runs under the evaluation protocol of Mule et al. (2026): transductive hyperedge classification with random 50/25/25 train/validation/test splits of the hyperedges; each run trains for at most 500 epochs with early stopping, and hyperparameters are selected by mean validation F1. As for the benchmarks of Section 5, all baseline numbers were obtained under the protocol of Appendix E, by re-running the public implementation of Mule et al. (2026) and, for HGraphormer, the implementation of its authors adapted to the same pipeline and tuned over its authors' search space. The public DSHN implementation runs out of memory on Molecular-1. SheaFormer is the best model on both datasets, by $+4.10$ points over DSHN-Light on Molecular-1 and $+1.69$ over DSHN on Molecular-2.

G Computational complexity

G.1 Forward-pass cost

We compare the implementation of SheaFormer used in Section 5 with the public implementation of DSHN[4] of Mule et al. (2026) used as our experimental baseline. Table 9 reports the cost of a forward pass of the two, both with diagonal restriction maps. Here $d_{\text{model}} = dc$ where d is the stalk dimension and c the number of channels, $\nu = \sum_e |e|$ the number of incidences, $\nu_2 = \sum_e |e|^2$ the number of ordered pairs of nodes within hyperedges, counted once per hyperedge, $\tilde{\nu}_2$ the number of distinct ordered pairs of nodes sharing at least one hyperedge, and h the classifier width. Both columns are derived from the two implementations: each cost counts the dense and sparse products

[4] https://github.com/EmaMule/DirectionalSheafHypergraphs



Figure 2: Share of the trainable parameters by component, SheaFormer and DSHN, over the stalk dimensions $d \leq 16$ that the search space admits at $c = 128$ ($dc \leq 2048$) and four of the input widths C_{in} of the benchmarks, computed from Table 9 with leading constants ($c^2 + 4cd$ generator, $2d_{\text{model}}h$ classifier), $L = 1$ and $h = 64$. Each bar is one configuration; the printed value is the share of the largest component.

that the code executes to compute the output, and each parameter count the weights of the modules it instantiates. For DSHN, this differs from the analysis of Mule et al. (2026) in two terms: the propagation runs over the $\tilde{\nu}_2$ distinct nonzero blocks of the assembled operator rather than over the ν_2 pairs, and the DSHN restriction generator used in our experiments contains, besides the $4c \times d$ map, a $c \times c$ linear map, distinct from W_2, that transforms the node representations entering the hyperedge context. The embedding, the stalk and feature maps (W_1 and W_2, i.e., W_{val} and W_{out}), normalization, nonlinearity, and readout have the same cost in the two models; they differ in the propagation and in the restriction generator.

Table 9: Forward-pass cost of SheaFormer and DSHN with diagonal restriction maps, up to constants, derived from the operations executed by the implementations used in Section 5: ours for SheaFormer and, for DSHN, the public implementation of Mule et al. (2026) with the restriction generator used in our experiments. n nodes, m hyperedges, $\nu = \sum_e |e|$ incidences, $\nu_2 = \sum_e |e|^2$, $\tilde{\nu}_2 \leq \nu_2$ distinct ordered pairs of nodes sharing a hyperedge, d stalk dimension, c channels, $d_{\text{model}} = dc$, L layers, C_{in} input channels, h classifier width.

	SheaFormer	**DSHN**
Embedding	$nC_{\text{in}}d_{\text{model}}$	$nC_{\text{in}}d_{\text{model}}$
Restriction maps	$L[(n+m)d_{\text{model}}^2 + \nu d_{\text{model}}]$	$L\,(nc^2 + \nu d_{\text{model}})$
Operator construction	—	$L\,d\nu_2$
Propagation	$L\nu d_{\text{model}}$	$L\tilde{\nu}_2 d_{\text{model}}$
Stalk and feature maps	$Ln(d+c)d_{\text{model}}$	$Ln(d+c)d_{\text{model}}$
Normalization, nonlinearity, residual	Lnd_{model}	Lnd_{model}
Readout	$nd_{\text{model}}h$	$nd_{\text{model}}h$
Parameters	$C_{\text{in}}d_{\text{model}} + 6c^2d^2 + L(d^2+c^2) + d_{\text{model}}h$	$C_{\text{in}}d_{\text{model}} + L\,(c^2+cd) + L(d^2+c^2) + d_{\text{model}}h$

Propagation. SheaFormer applies $Q^{pp} = (B^p)^* B^p$ as two sparse products with the boundary operator, a gather over the incidences, a sum over each hyperedge, and a scatter back to the nodes, at cost $\nu\, d_{\mathrm{model}}$ per layer, and never forms the operator. At every layer, the DSHN implementation assembles the normalized operator as a chain of sparse–sparse products, which evaluates the $d\,\nu_2$ pairwise products within the hyperedges and accumulates them on the $d\,\tilde{\nu}_2$ nonzeros of the operator, and applies it as a sparse–dense product, at cost $\tilde{\nu}_2\, d_{\mathrm{model}}$. On graphs ($|e| = 2$) $\nu = 2m$ and $\tilde{\nu}_2 \le \nu_2 = 4m$, and the propagation costs of the two models differ only by a constant factor; on hypergraphs the assembly grows with ν_2/ν, the size-weighted mean hyperedge size, i.e., the quadratic dependence on $|e|$ noted by Mule et al. (2026), and the propagation with $\tilde{\nu}_2/\nu$.

Restriction maps. The role-aware bilinear generator projects all d_{model} channels of the node and of the hyperedge context through W^p_{node} and W^p_{edge}, $p = 1, \dots, d$, at cost $(n+m)d^2_{\mathrm{model}}$ per layer: since the context of Eq. 11 is a sum, the projections commute with it and are applied to the n node and m hyperedge rows rather than to the ν incidences, and only the d inner products are evaluated per incidence. One set of generator weights, with $6d^2_{\mathrm{model}}$ parameters, is shared by the L layers and evaluated at each of them. The DSHN generator used in our experiments averages the d stalks of each node, applies a $c \times c$ linear map, distinct from W_2, to the n averaged representations, sums the result over each hyperedge, and evaluates a $4c \times d$ linear map, the generator Φ of Eq. 3 applied to the averaged node representation and $\mathbf{h}_e$, on every incidence, at cost $nc^2 + \nu\, d_{\mathrm{model}}$ per layer. The cost per evaluation of the generator is therefore higher by a factor of order $(n+m)\, d^2_{\mathrm{model}}/(nc^2 + \nu\, d_{\mathrm{model}})$ for SheaFormer, which evaluates d rank-c compatibility maps over d_{model} channels at every layer, whereas SheaFormer incurs neither the operator assembly, quadratic in the hyperedge size, nor the propagation over the $\tilde{\nu}_2$ blocks of the assembled operator.

Parameter budget. Figure 2 reports the parameter counts of Table 9 as shares of the total, over the stalk dimension d and the input width C_{in}. The two models embed the input with the same $C_{\mathrm{in}} \times d_{\mathrm{model}}$ map and read out with the same classifier, and differ in what is spent between the two. The DSHN generator, a $c \times c$ and a $4c \times d$ linear map, holds between 0.3% and 34% of the parameters, and its share decreases with d, since its c^2 term is independent of d while the embedding and the classifier grow with d_{model}. The generator and the stalk and feature maps W_1 and W_2 exceed 10% only on the benchmarks with one-dimensional input ($C_{\mathrm{in}} = 1$), up to $d = 8$ and $d = 4$ respectively, and at $d \le 2$ for $C_{\mathrm{in}} = 300$. The budget is otherwise held by the embedding or, when $C_{\mathrm{in}} = 1$, by the classifier, which holds between 33% and 86% of the parameters and at $d = 1$ shares them equally with the generator and with W_1 and W_2. In SheaFormer the restriction generator holds the majority of the parameters as soon as $6d_{\mathrm{model}} > C_{\mathrm{in}}$, and between 76% and 99% at $d = 16$, where besides the generator only the embedding of the two widest inputs exceeds 3% of the parameters (12% at $C_{\mathrm{in}} = 1703$ and 23% at $C_{\mathrm{in}} = 3703$). Capacity therefore moves from the input projection to the maps that define the operator.

G.2 Measured cost

Tables 10–14 report the cost of the implementations used in Section 5 for SheaFormer, DSHN and DSHN-Light on one NVIDIA A100 GPU with 64 GB of memory, for $L = 3$ layers and $c = 128$ channels at the stalk dimensions $d \in \{1, 4, 16\}$, i.e., $d_{\mathrm{model}} \in \{128, 512, 2048\}$. All computations are in single precision (float32, TF32 disabled). FLOPs are counted per executed operation with a PyTorch dispatch-level counter that includes the sparse–dense and sparse–sparse products, counting a multiply–add as two operations and complex arithmetic in real operations. Times are measured with CUDA events and reported as the median of three repetitions after three warm-up training steps; the three repetitions differ by a median of 0.5%. Peak memory is the maximum GPU memory allocated during one step, including parameters and data, and, for a training step, gradients and optimizer state. The measurements are consistent with Table 9. For $d \ge 4$, SheaFormer executes more floating-point operations than DSHN, since the $(n+m)d^2_{\mathrm{model}}$ term of its restriction generator dominates; for $d = 1$ it executes fewer on eight of the twelve benchmarks. The operations of the two models differ in arithmetic intensity: those of SheaFormer are dense matrix products, whereas the assembly of the operator of DSHN, $L\, d\, \nu_2$, is a chain of sparse–sparse products and its propagation, $L\, \tilde{\nu}_2\, d_{\mathrm{model}}$, a sparse–dense product, both of lower arithmetic intensity than dense products. Accordingly, the forward pass of SheaFormer is the fastest of the three models for $d = 1$ on all twelve benchmarks and for $d = 4$ on ten, and the forward pass of DSHN is faster only on Citeseer at $d = 16$. In training, the backward pass of DSHN differentiates through the sparse–sparse products

of every layer, whereas that of SheaFormer consists of the gathers and scatters of its propagation and of dense products. The peak memory of a training step of DSHN is 9.8 times that of its forward pass (median over the configurations; maximum 78), against 2.2 for SheaFormer, and exceeds the GPU memory for $d \geq 4$ on Roman-empire and Amazon-ratings and for $d = 16$ on Squirrel and email-EU. In the remaining configurations, the training step of SheaFormer is 2.5–11 times shorter than that of DSHN for $d = 1$, 2.8–25 times for $d = 4$ and 2.0–3.2 times for $d = 16$, with lower peak memory. DSHN-Light does not backpropagate through the construction of the operator (Mule et al., 2026): its forward pass requires about the same memory as that of DSHN, and its training step 3.6 times less (median; maximum 67). Relative to SheaFormer, DSHN-Light requires less training memory for $d \geq 4$ on every benchmark except Citeseer at $d = 4$ and a shorter training step for $d = 16$ on nine benchmarks, where the d^2_{model} term of the generator dominates; SheaFormer has the shorter training step for $d = 1$ on ten benchmarks and for $d = 4$ on nine.

Table 10: Floating-point operations (GFLOPs) of one forward pass, for $L = 3$ and $c = 128$ ($d_{\text{model}} = 128d$).

	$d = 1$, $d_{\text{model}} = 128$			$d = 4$, $d_{\text{model}} = 512$			$d = 16$, $d_{\text{model}} = 2048$		
Dataset	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light
Roman-empire	30.5	17.9	14.7	390	58.2	54.6	5,842	226	220
Squirrel	4.52	7.28	6.78	43.6	27.7	26.7	582	110	107
Chameleon	1.72	2.61	2.46	16.9	9.89	9.66	229	39.3	38.7
email-EU	1.59	2.27	1.99	17.1	8.43	7.75	240	33.3	31.1
Telegram	0.40	0.30	0.23	4.20	1.03	0.87	58.4	4.03	3.49
Cornell	0.32	0.36	0.33	3.56	1.32	1.29	50.7	5.24	5.19
email-Enron	0.21	0.12	0.09	2.51	0.39	0.34	36.8	1.53	1.38
Wisconsin	0.49	0.53	0.50	5.73	1.99	1.95	83.0	7.89	7.82
Amazon-ratings	33.9	20.0	16.4	425	65.5	60.9	6,326	254	246
Texas	0.33	0.38	0.35	3.74	1.40	1.37	53.6	5.53	5.48
Citeseer	6.77	11.6	11.2	59.8	44.6	44.1	762	177	177
Cora	3.97	4.65	4.27	42.5	17.0	16.6	596	67.2	66.6

Table 11: Time (ms) of one forward pass without gradients. Setting as in Table 10; we report "–" for configurations whose training step exceeds the GPU memory.

	$d = 1$, $d_{\text{model}} = 128$			$d = 4$, $d_{\text{model}} = 512$			$d = 16$, $d_{\text{model}} = 2048$		
Dataset	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light
Roman-empire	11.3	27.6	16.6	53.7	–	42.4	395.3	–	154.0
Squirrel	6.1	27.2	13.9	18.0	54.6	27.5	88.0	–	88.9
Chameleon	6.1	19.7	10.8	6.4	22.5	12.4	26.3	32.9	19.2
email-EU	6.2	29.0	15.1	12.2	52.1	28.1	50.6	–	84.7
Telegram	5.9	19.3	10.9	6.4	21.8	12.5	14.9	30.4	15.4
Cornell	5.4	17.8	10.0	5.7	18.1	10.2	6.1	20.0	11.2
email-Enron	6.0	18.4	10.1	6.3	19.6	10.9	7.5	22.3	12.1
Wisconsin	5.6	17.8	9.9	5.7	18.7	10.7	8.3	21.9	12.4
Amazon-ratings	17.0	34.9	20.1	78.6	–	51.8	515.5	–	196.1
Texas	5.7	18.2	10.2	6.2	19.0	10.9	6.6	20.8	12.1
Citeseer	6.1	20.6	11.9	8.5	24.1	14.4	55.4	52.6	39.9
Cora	7.5	21.6	12.2	7.5	27.1	15.3	48.5	53.6	34.3

Table 12: Peak GPU memory (GiB) of one forward pass without gradients. Setting and "–" as in Table 11.

	$d = 1$, $d_{\text{model}} = 128$			$d = 4$, $d_{\text{model}} = 512$			$d = 16$, $d_{\text{model}} = 2048$		
Dataset	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light
Roman-empire	0.39	0.49	0.49	1.40	–	1.10	5.55	–	4.02
Squirrel	0.28	0.30	0.29	1.00	0.79	0.75	3.96	–	2.74
Chameleon	0.08	0.09	0.09	0.23	0.11	0.11	0.90	0.32	0.31
email-EU	0.18	0.17	0.16	0.68	0.52	0.48	2.75	–	1.86
Telegram	0.06	0.05	0.05	0.20	0.12	0.12	0.82	0.43	0.42
Cornell	0.02	0.02	0.02	0.04	0.03	0.03	0.18	0.07	0.07
email-Enron	0.03	0.03	0.03	0.08	0.03	0.03	0.33	0.09	0.08
Wisconsin	0.02	0.03	0.03	0.05	0.04	0.04	0.21	0.09	0.08
Amazon-ratings	0.76	0.81	0.81	2.88	–	1.21	11.41	–	4.44
Texas	0.02	0.02	0.02	0.04	0.03	0.03	0.18	0.07	0.07
Citeseer	0.10	0.31	0.30	0.23	0.32	0.32	0.79	0.75	0.75
Cora	0.07	0.11	0.11	0.20	0.19	0.18	0.76	0.55	0.55

Table 13: Time (ms) of one training step, comprising the forward pass, the backward pass and the parameter update. Setting as in Table 10; OOM: the training step exceeds the GPU memory.

	$d = 1$, $d_{\text{model}} = 128$			$d = 4$, $d_{\text{model}} = 512$			$d = 16$, $d_{\text{model}} = 2048$		
Dataset	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light
Roman-empire	36.6	149.5	39.8	160.1	OOM	118.8	1,188	OOM	460.1
Squirrel	29.0	216.9	27.4	50.7	693.8	49.8	221.8	OOM	166.2
Chameleon	19.0	54.4	22.8	23.1	66.9	24.6	74.6	151.6	39.0
email-EU	21.8	244.0	28.6	31.2	792.1	47.8	121.2	OOM	150.4
Telegram	20.1	56.2	24.7	21.8	68.5	25.8	43.2	109.9	29.5
Cornell	17.0	48.0	21.9	17.5	50.5	22.2	22.9	55.4	24.2
email-Enron	18.0	49.0	21.9	19.3	55.1	22.9	27.6	63.2	24.4
Wisconsin	17.1	48.7	21.9	17.5	52.3	23.0	28.4	63.3	25.2
Amazon-ratings	49.3	181.9	44.0	213.2	OOM	137.3	1,448	OOM	544.8
Texas	17.7	50.8	23.1	19.3	53.8	24.3	24.2	59.9	25.8
Citeseer	19.0	54.4	24.0	29.2	98.9	32.3	170.8	539.5	91.7
Cora	24.7	61.5	25.0	27.9	102.5	30.2	144.5	433.6	80.7

Table 14: Peak GPU memory (GiB) of one training step. Setting and OOM as in Table 13.

	$d = 1$, $d_{\text{model}} = 128$			$d = 4$, $d_{\text{model}} = 512$			$d = 16$, $d_{\text{model}} = 2048$		
Dataset	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light	SheaFormer	DSHN	DSHN-Light
Roman-empire	1.63	5.81	1.63	6.36	OOM	5.41	25.53	OOM	20.67
Squirrel	0.46	7.85	0.43	1.72	30.16	1.04	6.96	OOM	3.49
Chameleon	0.13	0.36	0.14	0.44	1.10	0.32	1.89	4.04	1.00
email-EU	0.28	10.37	0.22	1.09	40.83	0.61	4.49	OOM	2.19
Telegram	0.09	0.37	0.06	0.32	1.25	0.15	1.42	4.78	0.50
Cornell	0.03	0.07	0.04	0.10	0.20	0.08	0.64	0.73	0.24
email-Enron	0.04	0.09	0.03	0.13	0.28	0.06	0.69	1.01	0.18
Wisconsin	0.04	0.11	0.05	0.12	0.36	0.10	0.68	1.36	0.32
Amazon-ratings	1.96	7.12	1.89	7.70	OOM	6.06	30.80	OOM	22.70
Texas	0.04	0.08	0.04	0.10	0.24	0.08	0.65	0.88	0.25
Citeseer	0.28	0.56	0.47	0.96	2.41	1.05	3.82	24.50	3.33
Cora	0.22	0.58	0.27	0.77	1.91	0.75	3.20	17.14	2.67

H Additional analysis of trained models

The two analyses in this appendix are computed on trained SheaFormer checkpoints of the configurations selected in Section 5, with 10 runs per dataset and deterministic evaluation.

H.1 Effective rank of the learned compatibility maps

Each stream p scores an incidence through the compatibility map $M^p = \frac{1}{\sqrt{c}} W^p_{\text{node}} (W^p_{\text{edge}})^\top$ of Section 4, whose rank is at most c by construction. This analysis is motivated by singular-direction views of Transformer circuits. An attention head's query–key interaction can be represented by the low-rank query–key matrix $W^h_{\text{qry}} (W^h_{\text{key}})^\top$ of Eq. 4 (Elhage et al., 2021), and recent mechanistic analyses have decomposed this matrix and other Transformer components into singular directions and intervened on subsets of them to identify compact task-relevant subspaces (Sen Sharma et al., 2026; Ahmad et al., 2025). M^p is the analogous bilinear object in SheaFormer. We therefore ask whether its nominal rank c reflects the dimensionality used after training, evaluating this not only spectrally but functionally through post-hoc singular-direction truncation.

Measurements. For every trained M^p we report the cumulative spectral energy

$$E(k) = \frac{\sum_{j \le k} \sigma_j^2}{\sum_j \sigma_j^2},$$

where $\sigma_1 \ge \cdots \ge \sigma_c$ are the singular values of M^p, and k_{90}, the number of singular directions at which E reaches 0.9, averaged over streams and runs, together with the same statistics at the random initialization of the same run. E depends only on the singular values of M^p, whereas the restriction-map coefficients $F^p_{i \lhd e}$ of Eq. 12 evaluate M^p on the node and context vectors of the dataset. Let $\mathbf{u}_i := \text{unwind}(\widehat{X}_i) \in \mathbb{R}^{1 \times 2dc}$ be the node vector of i, as in Section 4, and $\mathbf{h}_{i \lhd e} := [\text{unwind}(\widehat{H}^H_{i \lhd e}) \| \text{unwind}(\widehat{H}^T_{i \lhd e})] \in \mathbb{R}^{1 \times 4dc}$ the context vector of the incidence $i \lhd e$, built from Eq. 11, so that $\mathbf{x}^p_i = \mathbf{u}_i W^p_{\text{node}}$ and $\mathbf{h}^p_{i \lhd e} = \mathbf{h}_{i \lhd e} W^p_{\text{edge}}$ are their projections in Eq. 12. Let $M^p = \sum_j \sigma_j \boldsymbol{\phi}_j^\top \boldsymbol{\psi}_j$ be the singular value decomposition of M^p, with orthonormal singular vectors

$\boldsymbol{\phi}_j \in \mathbb{R}^{1\times 2dc}$ and $\boldsymbol{\psi}_j \in \mathbb{R}^{1\times 4dc}$. Then $F^p_{i\triangleleft e} = \mathbf{u}_i\, M^p\, \mathbf{h}^\top_{i\triangleleft e} = \sum_j \sigma_j\, (\mathbf{u}_i \boldsymbol{\phi}_j^\top)(\boldsymbol{\psi}_j \mathbf{h}^\top_{i\triangleleft e})$, and the best rank-$k$ approximation of M^p keeps the terms $j \le k$. Writing $F^p_{i\triangleleft e}(k)$ for the restriction-map coefficient computed with this approximation, we measure the relative reconstruction score of the restriction-map coefficients, pooled over the d streams and all incidences $i \triangleleft e$ of the hypergraph,

$$E_F(k) = 1 - \frac{\sum_{p=1}^{d} \sum_{i\triangleleft e} \left(F^p_{i\triangleleft e}(k) - F^p_{i\triangleleft e}\right)^2}{\sum_{p=1}^{d} \sum_{i\triangleleft e} \left(F^p_{i\triangleleft e}\right)^2},$$

so that a stream contributes in proportion to the squared norm of its restriction-map coefficients. $E(k)$ measures how many singular directions are needed to reconstruct the learned compatibility map M^p as a matrix, while $E_F(k)$ measures how many directions are needed to reproduce its action on the data. We summarize E_F by k^F_{90}, the smallest $k \in \{1, 2, 4, 8, 16\}$ at which E_F, averaged over runs, reaches 0.9. E_F is computed at the first layer, where the truncated and the full model receive the same input, so that it depends only on M^p and the data; computed over all layers, whose inputs beyond the first layer already reflect the truncation, k^F_{90} is equal or smaller on every dataset except email-Enron, where it is 2 instead of 1 (Table 15). We then replace every M^p by its best rank-k approximation and, as a control, by k of its c singular directions drawn uniformly at random. For each run, the paired accuracy change is the test accuracy of the truncated model minus that of the $k = c$ model of the same run; we report its mean over the 10 runs with a 95% Student's t interval, and the smallest rank whose interval stays within the 0.5-point margin, i.e., above -0.5 points.

Table 15: Spectral concentration, coefficient reconstruction and post-hoc rank truncation of the learned compatibility maps M^p. c is the number of channels, k_{90} the number of singular directions holding 90% of the spectral energy, averaged over streams and runs (trained, and at the initialization of the same run), and $E(1)$ the spectral energy of the leading singular direction, averaged over streams and runs. $E_F(1)$ and $E_F(4)$ are the reconstruction scores of the restriction-map coefficients at ranks 1 and 4, averaged over runs, and k^F_{90} the smallest rank at which E_F reaches 0.9, computed at the first layer and over all layers. k is the smallest rank at which the 95% interval of the paired accuracy change stays within the 0.5-point margin, and Δ the paired accuracy change at that rank with its 95% interval; where no rank below c does, $k = c$ and Δ is omitted. All ranks are numbers of retained singular directions out of c.

		Spectrum of M^p			Restriction-map coefficients				Truncation	
Dataset	c	k_{90}	k_{90} (init.)	$E(1)$	$E_F(1)$	$E_F(4)$	k^F_{90} (first)	k^F_{90} (all)	k	Δ at k
Telegram	256	1.5	203	0.872	0.714	0.999	2	2	2	$+1.15 \pm 0.96$
Texas	128	2.5	110	0.756	0.695	1.000	2	2	4	$+0.27 \pm 0.61$
Wisconsin	128	2.8	107	0.709	0.783	0.999	2	2	16	$+0.20 \pm 0.44$
Amazon-ratings	32	1.5	28	0.872	0.852	0.999	2	2	4	-0.08 ± 0.12
Citeseer	64	5.6	55	0.849	0.892	0.986	2	2	2	$+0.06 \pm 0.24$
Chameleon	256	22.7	214	0.378	0.365	0.808	8	4	256	–
Roman-empire	32	6.5	28	0.435	0.601	0.947	4	4	8	-0.24 ± 0.13
Cora	64	20.0	56	0.544	0.722	0.952	4	1	4	-0.07 ± 0.23
Cornell	4	1.9	4	0.786	0.912	1.000	1	1	4	–
Squirrel	16	7.8	14	0.417	0.774	0.972	4	4	12	-0.05 ± 0.41
email-Enron	512	305.8	311	0.031	0.940	0.999	1	2	2	$+0.00 \pm 0.00$
email-EU	512	308.1	311	0.023	0.891	0.981	2	1	1	$+0.16 \pm 0.34$

Results. Table 15 and Figure 3 report the three complementary notions of effective rank on the twelve datasets, ordered by k_{90}/c. First, the trained spectra are substantially more concentrated than at initialization on every dataset except email-Enron and email-EU. For example, 90% of the spectral energy is carried by only 1.5 of the 256 available directions on Telegram and 2.5 of 128 on Texas, compared with 203 and 110 directions at initialization. In contrast, email-Enron and email-EU remain spectrally diffuse after training, requiring 306 and 308 of 512 directions to retain 90% of the energy, versus 311 at initialization. The restriction-map coefficients, however, can depend on substantially fewer directions than the spectrum of M^p itself suggests. This distinction is most evident on email-Enron and email-EU, where the leading singular direction alone gives $E_F(1) = 0.940$ and 0.891 while accounting for only 3.1% and 2.3% of the spectral energy. Thus, most of the norm of M^p lies in directions that contribute little to the bilinear evaluations realized on the observed node–context pairs. More generally, E_F reaches 0.9 with at most 2 directions on eight of the twelve datasets and with at most 8 on all of them, while k_{90} ranges from 1.5 to 308 and exceeds k^F_{90} by

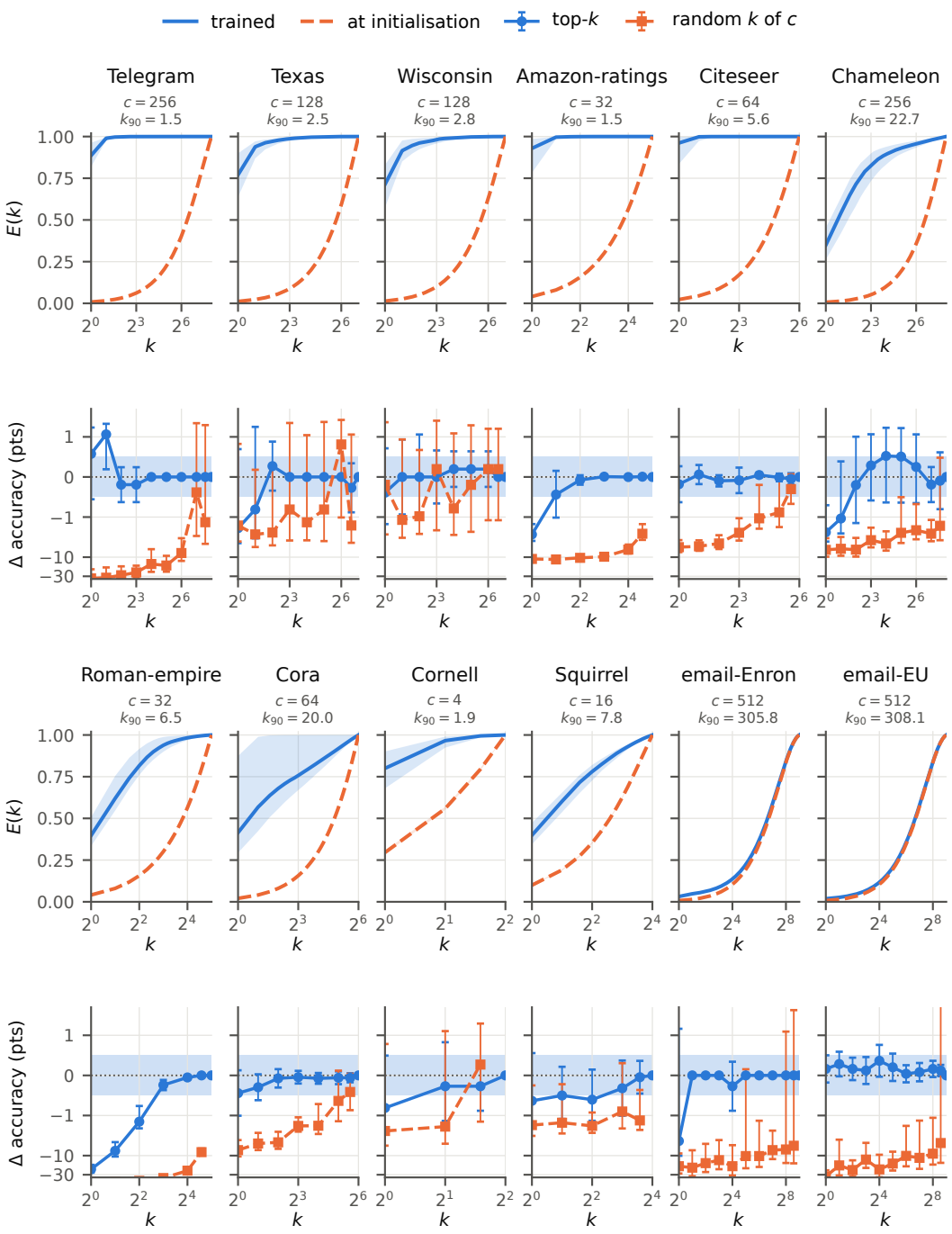


Figure 3: Effective rank of the learned compatibility maps, on the twelve datasets in two blocks of six, ordered by k_{90}/c with k_{90} averaged over streams and runs as in Table 15, so that the columns run from the most to the least concentrated spectrum. **Top row of each block:** cumulative spectral energy $E(k)$ of M^p against the retained rank k, median and interquartile range over streams and runs, for the trained maps and for the initialization of the same runs. **Bottom row of each block:** paired accuracy change when every M^p is replaced at inference by its best rank-k approximation (top-k) or by k randomly chosen singular directions (random k), with respect to the $k = c$ model of the same run, with error bars giving the 95% confidence interval over the 10 runs, on a symmetric-logarithmic axis that is linear for $|\Delta| \leq 1$ point and common to both blocks; the shaded band is the 0.5-point margin. Both rows share the same abscissa, so the rank at which E saturates can be compared with the rank at which accuracy is preserved.

two orders of magnitude on email-Enron and email-EU. On Telegram, email-Enron and email-EU the input is one-dimensional ($C_{\text{in}} = 1$; email-Enron and email-EU have no node features, Table 4) and the input embedding is linear, so at the first layer all node vectors take the form $\mathbf{u}_i = s_i \mathbf{w}$ for a common vector $\mathbf{w}$ and scalar input s_i, so each $F^p_{i \lhd e}$ depends on M^p only through $\mathbf{w} M^{p,H} \mathbf{w}^\top$ and $\mathbf{w} M^{p,T} \mathbf{w}^\top$ of Eq. 29. Over all layers, where later node representations are no longer collinear, k^F_{90} is still at most 2 on all three datasets (Table 15). The data-dependent measure E_F also better reflects the ranks required to preserve predictive performance than spectral energy alone. Chameleon, for example, concentrates 90% of its spectral energy in only 9% of the available directions, compared with 31% on Cora. However, at $k = 4$, E_F is only 0.81 on Chameleon but 0.95 on Cora, and accuracy is preserved at $k = 4$ on Cora but at no rank below c on Chameleon. Truncation remains within the 0.5-point margin with at most 4 directions on seven of the twelve datasets and with at most 16 on ten of them, and with a single direction of 512 on email-EU. No rank below c satisfies this criterion on Chameleon, where the mean change is at least -0.20 points for $k \geq 4$ but the lower bound of the interval remains below -0.5, and on Cornell ($c = 4$), where a single additional error in one of the ten runs excludes $k = 3$. At the same ranks, randomly selected singular directions remain within the margin on none of the ten datasets with a qualifying rank, reducing accuracy by 0.8 to 36.2 points, with the 95% interval entirely below zero on eight of them.

Interpretation. On these twelve datasets, the rank-c factorization is overcomplete relative to the compatibility function realized on the data after training. A small subset of the leading singular directions is sufficient both to reconstruct the restriction-map coefficients and, on ten of the twelve datasets, to preserve predictive accuracy. In contrast, retaining the same number of randomly selected singular directions stays within the margin on none of these ten. This indicates that training concentrates the functionally relevant computation into a small, structured subset of singular directions, even when the spectrum of M^p itself is not strongly low-rank.

H.2 Near-harmonic behavior of the complex whitening

Theorem 2 describes two mechanisms by which Norm moves a harmonic propagated update out of $\ker L$. On the trained operators, the normalized Laplacian $L = I_{nd} - Q_{\vec{\mathcal{F}}^N}$, whose spectrum lies in $[0, 1]$, has at most three eigenvalues below 10^{-5} at any layer on eight of the ten datasets whose spectrum is computed, so exact membership in $\ker L$ is uninformative on these datasets; on Citeseer and Cora, $\ker L$ is nontrivial and spans up to 10% and 2% of the space on which L acts, and the size of Roman-empire and Amazon-ratings precludes a dense eigendecomposition. We therefore evaluate the same mechanisms through the Rayleigh quotient rather than through membership in $\ker L$,

$$\mathrm{rq}(z) = \frac{z^* L z}{\|z\|^2} \in [0, 1], \qquad z \neq 0, \tag{48}$$

which measures normalized Laplacian energy and vanishes exactly on the harmonic space; we call z near-harmonic when $\mathrm{rq}(z) \leq 0.1$, a tenth of the upper end of its range.

Measurements. For each run, layer and feature channel we record rq at the four successive stages of the normalization of Eq. 10, written in the form of Lemma 1 as $\mathrm{Norm}(v) = \xi \mathbf{1} + \alpha P v + \eta \overline{Pv}$: the propagated branch v, the centered branch Pv, the whitened update $\alpha Pv + \eta \overline{Pv}$, and $\mathrm{Norm}(v)$ after the learned shift. The increment from the first to the second stage is the centering mechanism, from the second to the third the conjugate mechanism, which vanishes when $\eta = 0$ since $\mathrm{rq}(\alpha z) = \mathrm{rq}(z)$ for $\alpha \neq 0$, and the last increment is due to the shift, which the theorem sets to zero. The conjugate mechanism can act only along a direction that Pv does not already span. Every value quoted below is a median over channels and runs plotted in Figure 4, and every increment is the difference between the medians of two consecutive stages.

Results. Figure 4 reports the stage-wise Rayleigh quotients on the twelve datasets. On email-EU and email-Enron the premise of the theorem is approximately satisfied: at the first layer the layer input has $\mathrm{rq}(x) = 0.066$ and 0.032, and the branch is near-harmonic at every layer, with $\mathrm{rq}(v) = 0.001$ and 0.003–0.010. Centering alone raises the update to $\mathrm{rq}(Pv) = 0.659$ and 0.538 at the first layer, and the Rayleigh quotient of the layer input x increases with depth, from 0.066 to 0.270 and from 0.032 to 0.260. The configurations selected on both datasets have $q = 0$, so L is real and the exact kernel-escape mechanism of Corollary 1 does not apply; empirically, the increment from the second to the third stage is also negligible on both, at most $+0.002$ at every layer and

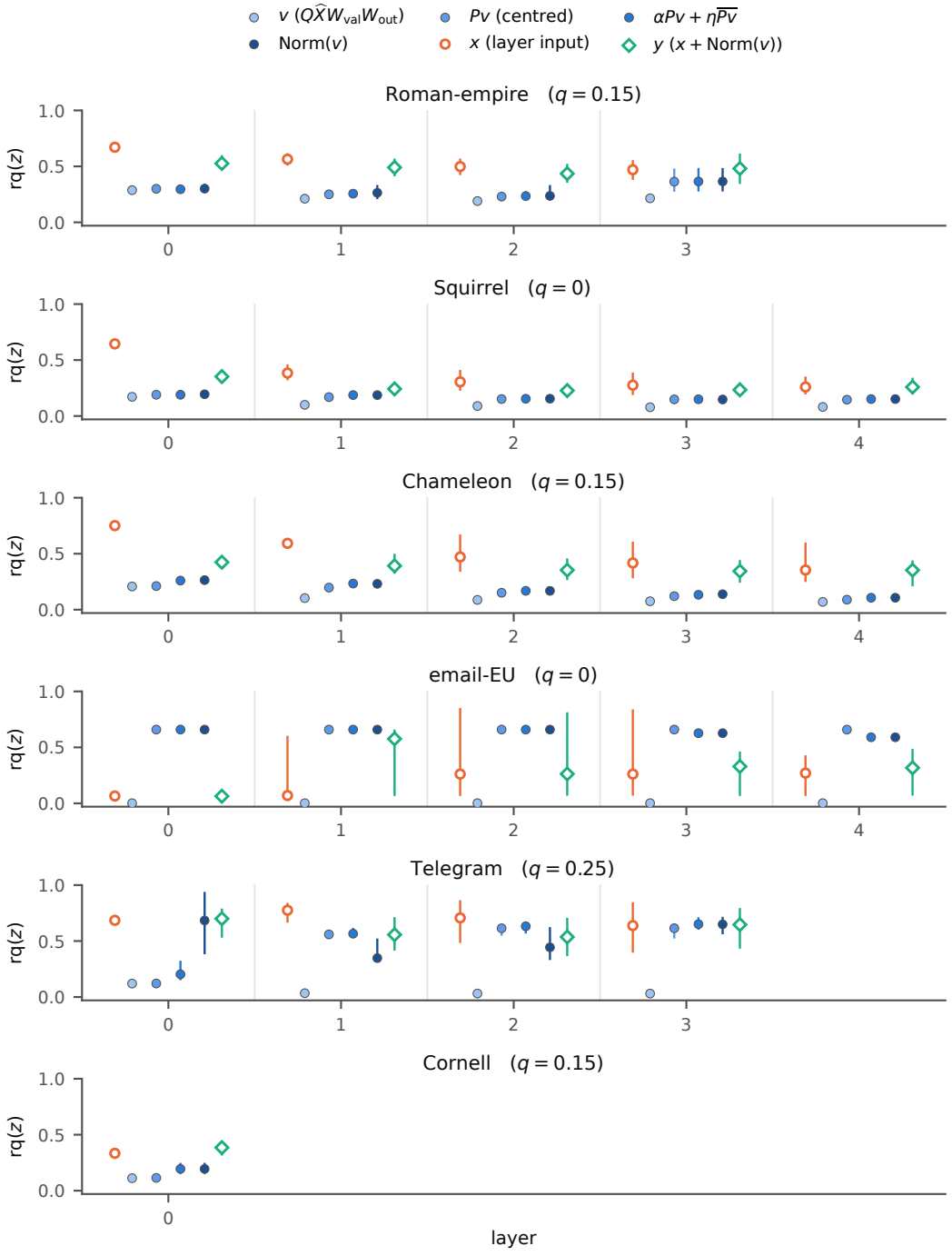


Figure 4: Per-channel Rayleigh quotients of Eq. 48 of the propagated update at the four successive stages of the complex whitening, per dataset and layer, summarized by the median (points) and interquartile range (bars) over channels and runs. Within a layer the four stages are drawn left to right, preceded by the layer input x and followed by the pre-activation output y, and vertical rules separate consecutive layers. The twelve datasets are ordered by CEHom as in Table 4 and share one layer axis; the figure continues on the next page.

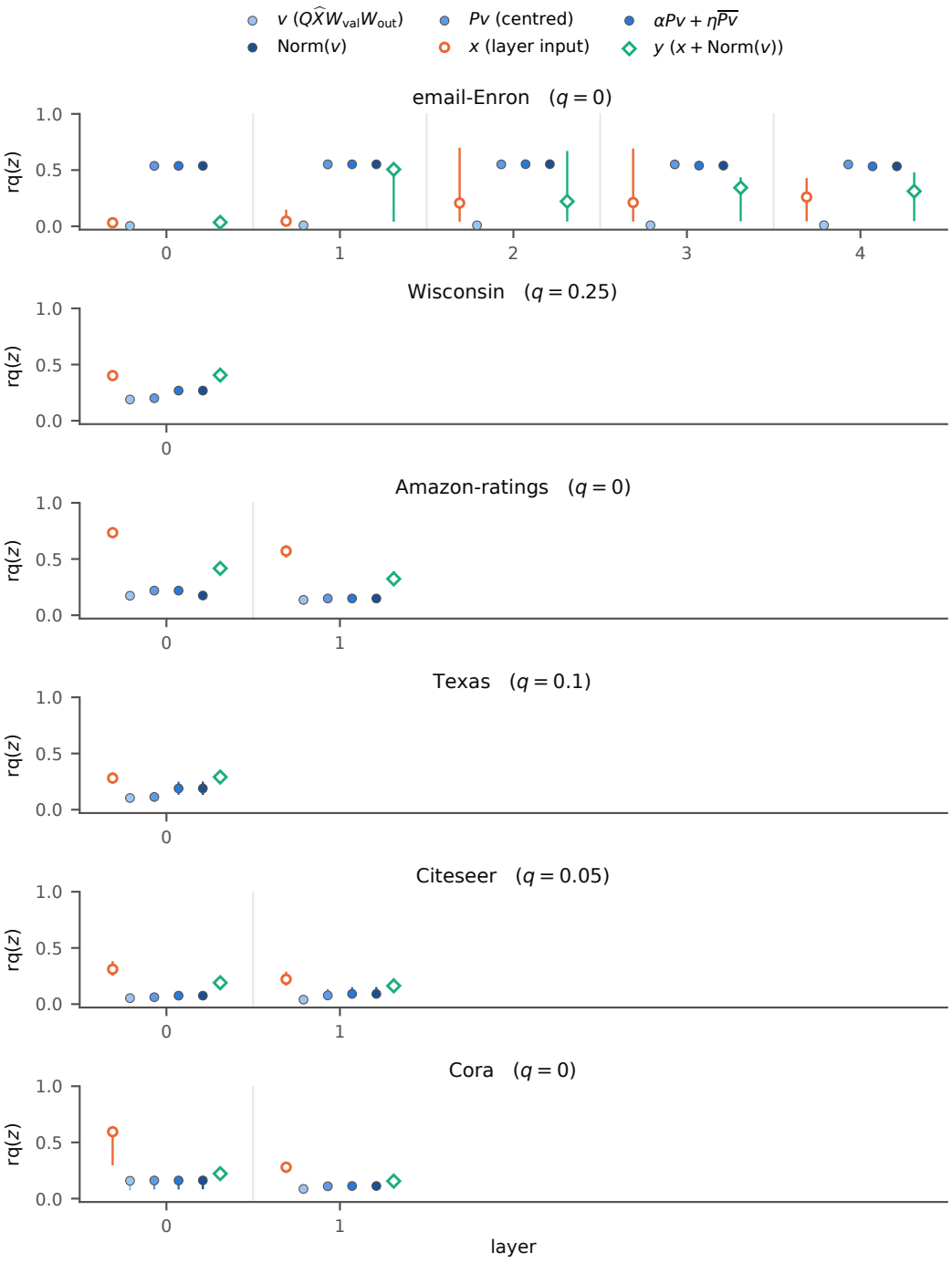


Figure 4 (continued).

negative in the last two. Across the twelve datasets, centering increases rq by more than 0.5 only on layers whose branch is similarly close to the harmonic space: in addition to email-EU and email-Enron, this occurs only on Telegram from the second layer onward, where $\mathrm{rq}(v) = 0.030$–0.034 and centering contributes $+0.53$ to $+0.58$ per layer; on all other layers it contributes at most $+0.15$. The conjugate stage contributes at most $+0.02$ on the five models with $q = 0$ and up to $+0.08$ on the seven with $q \neq 0$, and it is the dominant mechanism on the single-layer Texas, Cornell and Wisconsin models, where centering contributes at most $+0.01$.

Interpretation. Both mechanisms of Theorem 2 are observed in the trained models, in the approximate form imposed by the near-harmonic regime: centering accounts for the increase at every layer of email-EU and email-Enron and at the near-harmonic layers of Telegram, and the conjugate mechanism contributes where the charge is nonzero and $\overline{v}$ provides a direction that v does not. The analysis establishes that the mechanisms occur, not that they account for the accuracy of the model. Two further observations qualify the statement of the theorem. The learned shift ξ, which the theorem sets to zero, increases rq from 0.203 to 0.684 at the first layer of Telegram and decreases it from 0.566 to 0.348 and from 0.632 to 0.444 at the following two layers; on the other datasets it changes rq by at most 0.01, except for a decrease of 0.04 at the first layer of Amazon-ratings. Moreover, rq replaces membership in $\ker L$, so the stages are verified in their approximate form.

I PROOFS

Proposition 1. *With diagonal restriction maps $F_{i\lhd e} = \mathrm{diag}(F^1_{i\lhd e}, \dots, F^d_{i\lhd e}), F^p_{i\lhd e} \in \mathbb{R}$, in the stream-major view the directed sheaf hypergraph Laplacian decomposes as the direct sum*

$$\widehat{L_{\vec{\mathcal{F}}}} = \bigoplus_{p=1}^{d} L^p, \qquad L^p = D^p - (B^p)^* B^p \in \mathbb{C}^{n\times n}, \tag{6}$$

where $B^p := B^{pp}$ is the p-th diagonal block of $\widehat{B}$, with $(B^p)_{ei} = \delta_e^{-1/2} S^{(q)}_{i\lhd e} F^p_{i\lhd e}$, and $D^p_{ii} = \sum_{e\ni i}(F^p_{i\lhd e})^2$; likewise $\widehat{Q_{\vec{\mathcal{F}}}} = \bigoplus_{p=1}^d (B^p)^* B^p$, and analogously for the normalized operators whenever every $D^p_{ii} > 0$.

Proof of Proposition 1. Since the restriction maps are diagonal, different streams never mix in the boundary operator: $B^{rs} = 0$ for $r \neq s$. Hence, in the stream-major view,

$$\widehat{B} = \mathrm{diag}(B^1, \dots, B^d),$$

with

$$(B^p)_{ei} = \frac{1}{\sqrt{\delta_e}} S^{(q)}_{i\lhd e} F^p_{i\lhd e}.$$

Therefore,

$$\widehat{Q_{\vec{\mathcal{F}}}} = \widehat{B}^* \widehat{B} = (B^1)^* B^1 \oplus \cdots \oplus (B^d)^* B^d.$$

Likewise,

$$F^\top_{i\lhd e} F_{i\lhd e} = \mathrm{diag}\big((F^1_{i\lhd e})^2, \dots, (F^d_{i\lhd e})^2\big),$$

so the degree operator decomposes as

$$\widehat{D}_{\mathrm{nds}} = D^1 \oplus \cdots \oplus D^d.$$

Since $\widehat{L_{\vec{\mathcal{F}}}} = \widehat{D}_{\mathrm{nds}} - \widehat{Q_{\vec{\mathcal{F}}}}$, we obtain

$$\widehat{L_{\vec{\mathcal{F}}}} = \bigoplus_{p=1}^{d} (D^p - (B^p)^* B^p) = \bigoplus_{p=1}^{d} L^p.$$

Each summand is exactly the scalar directed sheaf hypergraph Laplacian associated with the restrictions $F^p_{i\lhd e}$. Finally, $\widehat{D}^{-1/2}_{\mathrm{nds}} = \bigoplus_{p=1}^d (D^p)^{-1/2}$, so conjugating Eq. 6 by it gives $\widehat{L_{\vec{\mathcal{F}}N}} = \bigoplus_{p=1}^d (D^p)^{-1/2} L^p (D^p)^{-1/2} =: \bigoplus_{p=1}^d L^p_N$, and $\widehat{Q_{\vec{\mathcal{F}}N}} = I_{nd} - \widehat{L_{\vec{\mathcal{F}}N}} = \bigoplus_{p=1}^d (I_n - L^p_N)$, which is the analogous decomposition for the normalized operators. □

Theorem 1. *Letting $\widehat{\Omega} = \Pi\Omega\Pi^\top = [\Omega^{ps}]_{p,s\in[d]}$ be the stream-major view of Ω and letting $\widehat{X} \in \mathbb{C}^{n\times dc}$ and $\widehat{Z} \in \mathbb{C}^{n\times dc'}$ be the stream representations of input and output of Eq. 7, we have $\widehat{Z} = \sum_{p=1}^{d}\sum_{s=1}^{d} \Omega^{ps}\widehat{X}W^s_{\mathrm{val}}W^p_{\mathrm{out}}$ with $W^s_{\mathrm{val}} = (W_1^\top \mathbf{e}_s)\otimes I_c \in \mathbb{C}^{dc\times c}$ and $W^p_{\mathrm{out}} = \mathbf{e}_p^\top \otimes W_2 \in \mathbb{C}^{c\times dc'}$. If $\widehat{\Omega}$ is block diagonal in the stream index, so that $\Omega^{ps} = 0$ for $p \neq s$, this reduces to*

$$\widehat{Z} = \sum_{p=1}^{d} \Omega^{pp}\widehat{X}W^p_{\mathrm{val}}W^p_{\mathrm{out}}. \tag{8}$$

In particular, this holds for $\Omega = Q_{\bar{\mathcal{F}}}$ whenever all restriction maps are diagonal, by Proposition 1.

Proof of Theorem 1. Set

$$U = (I_n \otimes W_1)XW_2.$$

For each node i, the corresponding block is $U_i = W_1X_iW_2$. Using the row-wise vectorization identity $\mathrm{rvec}(AXB) = \mathrm{rvec}(X)(A^\top \otimes B)$, where rvec concatenates the rows of a matrix (Henderson & Searle, 1981), the stream representations satisfy

$$\widehat{U} = \widehat{X}(W_1^\top \otimes W_2). \tag{49}$$

Since $\widehat{\Omega} = [\Omega^{ps}]_{p,s\in[d]}$, its action on $\widehat{U}$ can be written as

$$\widehat{Z} = \sum_{p=1}^{d}\sum_{s=1}^{d} \Omega^{ps}\widehat{U}(\mathbf{e}_s\mathbf{e}_p^\top \otimes I_{c'}). \tag{50}$$

The right factor selects input stream s and places the propagated result in output stream p. Substituting Eq. 49 into Eq. 50 and applying the mixed-product rule for Kronecker products gives

$$\begin{aligned}(W_1^\top \otimes W_2)(\mathbf{e}_s\mathbf{e}_p^\top \otimes I_{c'}) &= (W_1^\top \mathbf{e}_s\mathbf{e}_p^\top)\otimes W_2 \\ &= \big[(W_1^\top \mathbf{e}_s)\otimes I_c\big]\big[\mathbf{e}_p^\top \otimes W_2\big] = W^s_{\mathrm{val}}W^p_{\mathrm{out}}.\end{aligned} \tag{51}$$

Substitution yields the stated expression for $\widehat{Z}$. If $\Omega^{ps} = 0$ for all $p \neq s$, only the terms with $p = s$ remain, giving Eq. 8. □

Theorem 2 (Full statement)**.** *Let $L = L^* \succeq 0$, $\mathcal{K} := \ker L$, $\mathcal{E}(z) := z^*Lz$, $x \in \mathcal{K}$ one channel of the current representation, and $v \in \mathbb{C}^{nd}$ the same channel of the propagated update in Eq. 10 before normalization. With $m(v) := \frac{1}{nd}\mathbf{1}^*v$, the centering projection $P := I_{nd} - \frac{1}{nd}\mathbf{1}\mathbf{1}^*$, so that $Pv = v - m(v)\mathbf{1}$, and the learned shift of* Norm *set to zero, Lemma 1 (Appendix D.2) gives $\mathrm{Norm}(v) = \alpha(v)Pv + \eta(v)\overline{Pv}$ with $\alpha(v), \eta(v) \in \mathbb{C}$ determined by the 2×2 whitening matrix. For $y := x + \mathrm{Norm}(v)$: **(1)** if $v \in \mathcal{K}$ and $\eta(v) = 0$, then $Ly = -\alpha(v)m(v)L\mathbf{1}$ and $\mathcal{E}(y) = |\alpha(v)m(v)|^2\mathcal{E}(\mathbf{1})$; hence, provided $\alpha(v) \neq 0$, centering leaves $\mathcal{K}$ if and only if $m(v) \neq 0$ and $\mathbf{1} \notin \mathcal{K}$; **(2)** if $v \in \mathcal{K}$ and $m(v) = 0$, then $Ly = \eta(v)L\overline{v}$ and $\mathcal{E}(y) = |\eta(v)|^2\mathcal{E}(\overline{v})$; hence whitening leaves $\mathcal{K}$ if and only if $\eta(v) \neq 0$ and $\overline{v} \notin \mathcal{K}$.*

Proof of Theorem 2. We assume $v \in \mathcal{K}$ and analyze the two additional mechanisms due to normalization. Since

$$Pv = v - m(v)\mathbf{1}, \qquad \overline{Pv} = \overline{v} - \overline{m(v)}\mathbf{1},$$

and the learned shift is set to zero, we have

$$\mathrm{Norm}(v) = \alpha(v)v + \eta(v)\overline{v} - \big(\alpha(v)m(v) + \eta(v)\overline{m(v)}\big)\mathbf{1}.$$

Hence

$$y = x + \alpha(v)v + \eta(v)\overline{v} - \big(\alpha(v)m(v) + \eta(v)\overline{m(v)}\big)\mathbf{1}.$$

Since $x, v \in \mathcal{K}$, we have $Lx = Lv = 0$, and therefore

$$Ly = \eta(v)L\overline{v} - \big(\alpha(v)m(v) + \eta(v)\overline{m(v)}\big)L\mathbf{1}. \tag{52}$$

For the centering mechanism, suppose that the normalization is complex-linear, so that $\eta(v) = 0$. Eq. 52 reduces to

$$Ly = -\alpha(v)m(v)L\mathbf{1}.$$

Since $L = L^* \succeq 0$ and $\ker L = \ker L^{1/2}$,

$$\mathcal{E}(y) = \|L^{1/2}y\|^2 = |\alpha(v)m(v)|^2\|L^{1/2}\mathbf{1}\|^2 = |\alpha(v)m(v)|^2\mathcal{E}(\mathbf{1}).$$

Thus, provided $\alpha(v) \neq 0$, the residual update is non-harmonic if and only if $m(v) \neq 0$ and $\mathbf{1} \notin \mathcal{K}$. In this case it is the centering operation $v \mapsto Pv$ that introduces the non-harmonic component.

For the complex-whitening mechanism, suppose instead that $m(v) = 0$. Then Eq. 52 becomes

$$Ly = \eta(v)L\overline{v}.$$

Again using $L = L^* \succeq 0$ and $\ker L = \ker L^{1/2}$,

$$\mathcal{E}(y) = \|L^{1/2}y\|^2 = |\eta(v)|^2\|L^{1/2}\overline{v}\|^2 = |\eta(v)|^2\mathcal{E}(\overline{v}).$$

Therefore $y \notin \mathcal{K}$ if and only if $\eta(v) \neq 0$ and $\overline{v} \notin \mathcal{K}$. This second normalization-induced mechanism is specific to the conjugate-linear component of the complex whitening map. □

Theorem 3. *For any hypergraph $\mathcal{H}$ with node features and orientation-independent hyperedge features and any two orientations $\vec{\mathcal{H}}_1$, $\vec{\mathcal{H}}_2$ of $\mathcal{H}$, DSHN with $q = 0$ produces identical node representations on $\vec{\mathcal{H}}_1$ and $\vec{\mathcal{H}}_2$ at every depth and for every parameter choice, so no permutation-invariant readout distinguishes them. In contrast, there exist $(\mathcal{H}, \vec{\mathcal{H}}_1, \vec{\mathcal{H}}_2)$ and SheaFormer parameters for which a single SheaFormer layer with $q = 0$ produces different representations on $\vec{\mathcal{H}}_1$ and $\vec{\mathcal{H}}_2$.*

Proof of Theorem 3. At layer $t = 0$, the DSHN node representations agree on $\vec{\mathcal{H}}_1$ and $\vec{\mathcal{H}}_2$ by assumption. Suppose that they agree at some layer t. Reorienting a hyperedge does not change its incident node set and therefore does not change the orientation-independent hyperedge representation used by the DSHN restriction generator. Hence, for every incidence,

$$F^{(t)}_{i \lhd e}(\vec{\mathcal{H}}_1) = F^{(t)}_{i \lhd e}(\vec{\mathcal{H}}_2).$$

At $q = 0$,

$$S^{(0)}_{i \lhd e} = 1$$

on both heads and tails, so the directed restriction maps, the boundary operator, the degree operator, and consequently the normalized signless operator $Q_{\vec{\mathcal{F}}_N^{(t)}}$ are identical on the two orientations. The shared linear maps, nonlinearities, optional residual connection, and normalization therefore produce the same next-layer representation. Induction proves the claim at every depth.

For the second claim, consider the hypergraph on three nodes with one hyperedge

$$e = \{1, 2, 3\}$$

and the two orientations

$$T_1(e) = \{1\}, \qquad H_1(e) = \{2, 3\},$$

and

$$T_2(e) = \{1, 2\}, \qquad H_2(e) = \{3\}.$$

Give every node the same nonzero real feature and consider one scalar stream of SheaFormer, with scalar stalk and channel maps $W_1 = W_2 = 1$. Since the node representation is constant, the two role kernels of Eq. 29 can be chosen so that their pairwise compatibilities are

$$\beta^H(i, j) = 1, \qquad \beta^T(i, j) = -2$$

for every pair of nodes. With $k_e = 2$, the leave-one-out formula of Eq. 30 gives, for the first orientation,

$$(F_{1 \lhd e}, F_{2 \lhd e}, F_{3 \lhd e}) = \left(1, -\frac{1}{2}, -\frac{1}{2}\right),$$

whereas for the second orientation it gives

$$(F_{1 \lhd e}, F_{2 \lhd e}, F_{3 \lhd e}) = \left(-\frac{1}{2}, -\frac{1}{2}, -2\right).$$

The same argument is unchanged under any common positive rescaling of the hyperedge context, since only the resulting signs are used below.

Let

$$\overline{Q} := D^{-1/2} B^* B D^{-1/2}$$

be the normalized scalar signless operator. Since there is only one hyperedge and $q = 0$,

$$D_{ii} = F_{i \lhd e}^2$$

and therefore

$$\overline{Q}_{ij} = \frac{1}{3} \operatorname{sgn}(F_{i \lhd e}) \operatorname{sgn}(F_{j \lhd e}).$$

Consequently,

$$\overline{Q}_1 \mathbf{1} = \left(-\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)^\top, \qquad \overline{Q}_2 \mathbf{1} = (1, 1, 1)^\top.$$

Thus the first propagation branch is nonconstant, whereas the second is constant.

Consider this block as the last layer of a one-layer SheaFormer, so that Drop and σ are omitted from Eq. 10. Set the learned shift $\boldsymbol{\beta}_\ell$ of Norm to zero and its learned scale Γ_ℓ to the identity. For the second orientation, centering annihilates the constant branch, so its normalized update is zero. For the first orientation, the centered branch is nonzero and nonconstant; since the covariance used by the whitening transform is ridge-regularized, its inverse square root is nonsingular, and the normalized update therefore remains nonzero and nonconstant. Adding the same constant residual representation to all nodes preserves this distinction, and, with its scale and shift chosen in the same way, the output normalization of the last layer, like unwind, is injective entrywise and preserves it too. Hence the two final node-representation multisets are different after one layer, proving the second claim. □

Theorem 4 (Full statement)**.** *Consider finite, loopless, unweighted directed graphs G without isolated nodes, with identical nonzero node features and identical orientation-independent edge features, and let $\mathcal{Q}_{\mathrm{D}}(G)$ and $\mathcal{Q}_{\mathrm{S}}(G)$ be the families of normalized scalar diffusion operators realizable on G by a scalar stream of DSHN with diagonal restriction maps and by a scalar stream of SheaFormer over all charges and restriction parameters. Then $\mathcal{Q}_{\mathrm{D}}(G) \subseteq \mathcal{Q}_{\mathrm{S}}(G)$ for every G, with strict inclusion for some G. Moreover, there exist two orientations G_A, G_B of the same undirected graph whose one-layer node-representation multisets under DSHN with diagonal restriction maps coincide for every charge and parameter choice, whereas one SheaFormer layer at $q = 0$ separates them for some parameter choice.*

Proof of Theorem 4. Throughout, G is finite, loopless and unweighted and every node has positive degree, and we consider the parameter choices for which the degree matrices are positive definite, so that the normalized operators are defined; both models are applied for one layer, and $\overline{Q}_{\mathrm{D}}$ and $\overline{Q}_{\mathrm{S}}$ denote the normalized scalar signless operators $D^{-1/2} B^* B D^{-1/2}$ of a scalar stream of DSHN and of SheaFormer. We first establish the operator inclusion. Since all node features and all orientation-independent edge features are identical, the DSHN restriction generator receives the same input at every incidence. Hence, within a scalar stream, it produces a common real restriction-map coefficient r. For a node i of degree d_i,

$$D_i^{\mathrm{D}} = r^2 d_i,$$

so that $r \neq 0$ for every operator in $\mathcal{Q}_{\mathrm{D}}(G)$. If i and j are joined by exactly one directed edge, the corresponding off-diagonal entry of the normalized signless operator has magnitude

$$\left| (\overline{Q}_{\mathrm{D}})_{ij} \right| = \frac{1}{2\sqrt{d_i d_j}}, \tag{53}$$

while its phase is determined by the charge through the phase factors $S^{(q)}$ of the two incidences.

Under identical features the bilinear generator of Eq. 12 is tied by role: $\mathbf{x}_i^p = \mathrm{unwind}(\widehat{X}_i) W_{\mathrm{node}}^p =: \mathbf{w}$ is common to all nodes and nonzero for a suitable W_{node}^p, and on a graph the context $\mathbf{h}_{i \lhd e}$ of a tail incidence is $[\mathrm{unwind}(\widehat{X}_j) \| 0]$ while that of a head incidence is $[0 \| \mathrm{unwind}(\widehat{X}_i)]$, with $\widehat{X}_i = \widehat{X}_j \neq 0$; the two role blocks of W_{edge}^p therefore assign one coefficient to every tail incidence and one to every head incidence, and any prescribed pair of values is realizable. SheaFormer can thus reproduce every such operator. Indeed, choose the same charge and let its role-aware restriction generator assign the same nonzero coefficient a to both head and tail incidences. Then

$$D_i^{\mathrm{S}} = a^2 d_i,$$

and both the magnitude and the charge-dependent phase of every normalized entry coincide with those of DSHN. Therefore

$$\mathcal{Q}_{\mathrm{D}}(G) \subseteq \mathcal{Q}_{\mathrm{S}}(G)$$

for every directed graph G.

The inclusion can be strict because SheaFormer may instead assign a coefficient $a > 0$ to every tail incidence and a different coefficient $b > 0$ to every head incidence. In that case,

$$D_i^{\mathrm{S}} = a^2 d_i^{\mathrm{out}} + b^2 d_i^{\mathrm{in}}, \tag{54}$$

and for adjacent nodes i and j,

$$\left|(\overline{Q}_{\mathrm{S}})_{ij}\right| = \frac{ab}{2\sqrt{D_i^{\mathrm{S}} D_j^{\mathrm{S}}}}. \tag{55}$$

Thus, although a common rescaling of all restriction-map coefficients cancels under normalization, the relative head–tail weight a/b does not: the magnitude of the normalized diffusion operator can depend on the nodes' in–out role profiles, whereas Eq. 53 depends only on their ordinary degrees.

To see that this produces a strict inclusion, consider the two orientations of the undirected four-cycle

$$\begin{aligned} G_A: &\quad 1 \to 2, \qquad 2 \to 3, \qquad 1 \to 4, \qquad 4 \to 3, \\ G_B: &\quad 1 \to 2, \qquad 3 \to 2, \qquad 1 \to 4, \qquad 4 \to 3. \end{aligned}$$

On G_A, every node has ordinary degree two, so every DSHN operator satisfies

$$\left|(\overline{Q}_{\mathrm{D}})_{ij}\right| = \frac{1}{4}$$

on every edge. For SheaFormer, however,

$$(d_1^{\mathrm{out}}, d_1^{\mathrm{in}}) = (2, 0), \qquad (d_3^{\mathrm{out}}, d_3^{\mathrm{in}}) = (0, 2),$$

while nodes 2 and 4 both have role degree $(1, 1)$. Hence

$$D_1^{\mathrm{S}} = 2a^2, \qquad D_3^{\mathrm{S}} = 2b^2, \qquad D_2^{\mathrm{S}} = D_4^{\mathrm{S}} = a^2 + b^2.$$

If $a \neq b$, for example,

$$\left|(\overline{Q}_{\mathrm{S}})_{12}\right| = \frac{b}{2\sqrt{2(a^2+b^2)}}, \qquad \left|(\overline{Q}_{\mathrm{S}})_{23}\right| = \frac{a}{2\sqrt{2(a^2+b^2)}},$$

which are different. No DSHN charge can reproduce this pattern because the charge changes only phases, not magnitudes. Therefore

$$\mathcal{Q}_{\mathrm{D}}(G_A) \subsetneq \mathcal{Q}_{\mathrm{S}}(G_A).$$

It remains to show that the additional operator expressivity yields an actual graph-distinguishing separation. The two orientations G_A and G_B above have the same multiset of out–in degree pairs,

$$\{\{(d_i^{\mathrm{out}}, d_i^{\mathrm{in}}) : i \in V\}\} = \{\{(2,0), (0,2), (1,1), (1,1)\}\}. \tag{56}$$

For DSHN, let

$$s = e^{-2\pi \mathbf{i} q_{\mathrm{D}}}.$$

Since the underlying graph is 2-regular, applying the normalized scalar operator to the constant signal gives

$$(\overline{Q}_{\mathrm{D}}\mathbf{1})_i = \frac{1}{2} + \frac{1}{4}\left(d_i^{\mathrm{out}}\overline{s} + d_i^{\mathrm{in}} s\right), \tag{57}$$

up to exchanging s and $\overline{s}$ according to the chosen phase convention. Thus the propagated representation of node i depends only on $(d_i^{\mathrm{out}}, d_i^{\mathrm{in}})$. By Eq. 56, G_A and G_B therefore produce the same multiset of propagated representations for every value of q_{D}. The same reasoning applies independently to every diagonal stream, while shared channel maps, pointwise nonlinearities, residual connections, and permutation-invariant normalization preserve equality of the resulting node-representation multisets. Hence

$$\forall q_{\mathrm{D}},\ \forall \theta_{\mathrm{D}}, \qquad \{\{X^{(1)}_{\mathrm{DSHN,i}}(G_A; q_{\mathrm{D}}, \theta_{\mathrm{D}})\}\} = \{\{X^{(1)}_{\mathrm{DSHN,i}}(G_B; q_{\mathrm{D}}, \theta_{\mathrm{D}})\}\}.$$

We finally construct a one-layer SheaFormer that distinguishes them. Set $q = 0$, take scalar stalk and channel maps $W_1 = W_2 = 1$, and choose tail and head restriction-map coefficients $a > 0$ and $b > 0$, respectively, with $a \neq b$. Let

$$r_0 = \sqrt{a^2 + b^2}.$$

Using Eq. 54 and Eq. 55, the multiset of $\overline{Q}_{\mathrm{S}}\mathbf{1}$ on G_A is

$$\left\{ \frac{1}{2} + \frac{b}{\sqrt{2}\, r_0}, \frac{1}{2} + \frac{a}{\sqrt{2}\, r_0}, \frac{1}{2} + \frac{a+b}{2\sqrt{2}\, r_0}, \frac{1}{2} + \frac{a+b}{2\sqrt{2}\, r_0} \right\}, \tag{58}$$

whereas on G_B it is

$$\left\{ \frac{3}{4} + \frac{b}{2\sqrt{2}\, r_0}, \frac{3}{4} + \frac{a}{2\sqrt{2}\, r_0}, \frac{1}{2} + \frac{ab}{2r_0^2} + \frac{a}{2\sqrt{2}\, r_0}, \frac{1}{2} + \frac{ab}{2r_0^2} + \frac{b}{2\sqrt{2}\, r_0} \right\}. \tag{59}$$

For instance, choosing $a = 1$ and $b = 2$, the multiset in Eq. 58 contains a repeated value, whereas the four values in Eq. 59 are pairwise distinct.

Consider the block as the last layer of a one-layer SheaFormer, so that Drop and σ are omitted from Eq. 10. Choose the learned scale Γ_ℓ of the normalization to be the identity and its shift $\boldsymbol{\beta}_\ell$ to be zero. Centering subtracts the same mean from every node, and whitening then applies to every entry the same real-linear map $C^{-1/2}$, which is nonsingular because the covariance is ridge-regularized; hence $\mathrm{Norm}(z_i) = \mathrm{Norm}(z_j)$ if and only if $z_i = z_j$, and equality multiplicities within each graph are preserved. The normalized representation of G_A still contains a repeated node value, whereas that of G_B contains four distinct node values. Adding the same constant residual representation to every node preserves this property, and, with its scale and shift chosen in the same way, the output normalization of the last layer, like unwind, is injective entrywise and preserves it too. Hence, for this choice of θ_{S},

$$\{\{X^{(1)}_{\mathrm{SF,i}}(G_A; 0, \theta_{\mathrm{S}})\}\} \neq \{\{X^{(1)}_{\mathrm{SF,i}}(G_B; 0, \theta_{\mathrm{S}})\}\},$$

which proves the claimed one-layer separation. □

Lemma 1 (Real-linear form of complex whitening)**.** *For every fixed $v \in \mathbb{C}^{nd}$, the real-linear map represented by $A(v)$ admits the unique decomposition $z \mapsto \alpha(v)z + \eta(v)\overline{z}$, where*

$$\alpha(v) = \frac{a_{11}(v) + a_{22}(v)}{2} + \frac{\mathbf{i}}{2}\big(a_{21}(v) - a_{12}(v)\big), \quad \eta(v) = \frac{a_{11}(v) - a_{22}(v)}{2} + \frac{\mathbf{i}}{2}\big(a_{21}(v) + a_{12}(v)\big). \tag{40}$$

Consequently,

$$\mathrm{Norm}(v) = \xi\mathbf{1} + \alpha(v)Pv + \eta(v)\overline{Pv}, \tag{41}$$

and $\eta(v) = 0$ if and only if $A(v)$ is complex-linear. Suppose, in addition, that $\Gamma_\ell = \gamma I_2$ for some $\gamma \in \mathbb{R} \setminus \{0\}$; this includes pure covariance whitening ($\gamma = 1$) and the initialization ($\gamma = 1/\sqrt{2}$). Then

$$\eta(v) = 0 \quad \Longleftrightarrow \quad \frac{1}{nd}\sum_{j=1}^{nd} (Pv)_j^2 = 0, \tag{42}$$

i.e., under scalar affine scaling the conjugate-linear coefficient vanishes exactly when the empirical pseudo-variance of the centered channel vanishes. For a general learned $\Gamma_\ell \in \mathbb{R}^{2\times 2}$, the expressions in Eq. 40 still apply, but the equivalence in Eq. 42 need not hold.

Proof of Lemma 1. Fix v and abbreviate $A = A(v)$. For $z = x + \mathbf{i}y$, the real-linear map represented by A gives $z' = (a_{11}x + a_{12}y) + \mathbf{i}(a_{21}x + a_{22}y)$. Substituting $x = (z + \overline{z})/2$ and $y = (z - \overline{z})/(2\mathbf{i})$,

$$z' = \left(\frac{a_{11} + a_{22}}{2} + \frac{\mathbf{i}}{2}(a_{21} - a_{12}) \right) z + \left(\frac{a_{11} - a_{22}}{2} + \frac{\mathbf{i}}{2}(a_{21} + a_{12}) \right) \overline{z}, \tag{60}$$

that is, $A(z) = \alpha z + \eta\overline{z}$ with α and η as in Eq. 40. The decomposition is unique: if $\alpha z + \eta\overline{z} = 0$ for every $z \in \mathbb{C}$, then $z = 1$ gives $\alpha + \eta = 0$ and $z = \mathbf{i}$ gives $\alpha - \eta = 0$, so $\alpha = \eta = 0$. Applying the map entrywise to the centered channel Pv and adding the shift ξ gives Eq. 41, and $z \mapsto \alpha z + \eta\overline{z}$ is complex-linear if and only if $\eta = 0$.

Now suppose $\Gamma_\ell = \gamma I_2$ with $\gamma \neq 0$, so that $A(v) = \gamma C(v)^{-1/2}$. Since $C(v) \succ 0$, its principal inverse square root is real and symmetric; hence $A(v)$ is real and symmetric, $a_{21}(v) = a_{12}(v)$,

and $\eta(v) = \frac{a_{11}(v)-a_{22}(v)}{2} + \mathbf{i}\, a_{12}(v)$. Consequently $\eta(v) = 0$ if and only if $A(v) = aI_2$ for some $a \in \mathbb{R} \setminus \{0\}$, i.e., since $A(v) = \gamma C(v)^{-1/2}$, if and only if $C(v) = cI_2$ for some $c > 0$. By Eq. 38, $C(v) = C^0(v) + \rho(v)I_2$ with $\rho(v) := \epsilon(1 + \operatorname{tr} C^0(v)) > 0$; since $\rho(v)I_2$ is itself a scalar matrix, $C(v)$ is a scalar multiple of I_2 if and only if $C^0(v)$ is. Write the centered complex entries as $z_j := (Pv)_j = x_j + \mathbf{i}y_j$, $j = 1, \ldots, nd$. By definition,

$$C^0(v) = \frac{1}{nd} \begin{bmatrix} \sum_j x_j^2 & \sum_j x_j y_j \\ \sum_j x_j y_j & \sum_j y_j^2 \end{bmatrix}, \tag{61}$$

whereas $\frac{1}{nd}\sum_j z_j^2 = \frac{1}{nd}\sum_j (x_j^2 - y_j^2) + \frac{2\mathbf{i}}{nd}\sum_j x_j y_j$. Therefore $\frac{1}{nd}\sum_j z_j^2 = 0$ if and only if $\sum_j x_j^2 = \sum_j y_j^2$ and $\sum_j x_j y_j = 0$, which are precisely the conditions under which $C^0(v)$ is a scalar multiple of I_2. Hence $\eta(v) = 0$ if and only if $\frac{1}{nd}\sum_j (Pv)_j^2 = 0$, which proves Eq. 42. □

Corollary 1 (Directed harmonic escape)**.** *Let $L_q = L_q^* \succeq 0$ be the directed sheaf Laplacian at charge q, and define $\mathcal{K}_q := \ker L_q$ and $\mathcal{E}_q(z) := z^* L_q z$. Assume that the real restriction-map coefficients do not depend on the charge, which enters only through the directional phases, so that $L_{-q} = \overline{L_q}$. Then*

$$\overline{\mathcal{K}_q} = \mathcal{K}_{-q}. \tag{44}$$

Let $x, v \in \mathcal{K}_q$, suppose that $m(v) = 0$ and $\xi = 0$, and set $y := x + \mathrm{Norm}(v)$. Then

$$\mathcal{E}_q(y) = |\eta(v)|^2 \mathcal{E}_q(\overline{v}) = |\eta(v)|^2 \mathcal{E}_{-q}(v). \tag{45}$$

Consequently, $\mathcal{E}_q(y) > 0 = \mathcal{E}_q(x)$ if and only if $\eta(v) \neq 0$ and $v \notin \mathcal{K}_{-q}$: the conjugate-linear component moves a q-harmonic branch along the direction $\overline{v}$, which is harmonic for the opposite charge $-q$ but not for the current charge q. If L_q is real, in particular at $q = 0$, then conjugation preserves $\mathcal{K}_q$ and this harmonic-escape mechanism vanishes.

Proof of Corollary 1. Let $v \in \mathcal{K}_q$, so that $L_q v = 0$. Taking entrywise complex conjugates and using Eq. 43 gives $L_{-q}\overline{v} = \overline{L_q}\,\overline{v} = \overline{L_q v} = 0$, hence $\overline{\mathcal{K}_q} \subseteq \mathcal{K}_{-q}$; the same argument with $-q$ in place of q gives the reverse inclusion and proves Eq. 44.

Since $m(v) = 0$ and $\xi = 0$, Lemma 1 gives $\mathrm{Norm}(v) = \alpha(v)v + \eta(v)\overline{v}$. Set $h := x + \alpha(v)v$. Because $x, v \in \mathcal{K}_q$, also $h \in \mathcal{K}_q$ and $L_q h = 0$, so $y = h + \eta(v)\overline{v}$ and, since all terms containing h vanish,

$$\mathcal{E}_q(y) = \big(h + \eta(v)\overline{v}\big)^* L_q \big(h + \eta(v)\overline{v}\big) = |\eta(v)|^2 \overline{v}^* L_q \overline{v} = |\eta(v)|^2 \mathcal{E}_q(\overline{v}). \tag{62}$$

Since L_q is Hermitian, $\mathcal{E}_q(\overline{v})$ is real, and therefore $\mathcal{E}_q(\overline{v}) = \overline{\mathcal{E}_q(\overline{v})} = \overline{\overline{v}^* L_q \overline{v}} = v^* \overline{L_q} v = v^* L_{-q} v = \mathcal{E}_{-q}(v)$, which proves Eq. 45. Because $L_{-q} \succeq 0$, $\mathcal{E}_{-q}(v) = 0$ if and only if $v \in \ker L_{-q} = \mathcal{K}_{-q}$; it follows that $\mathcal{E}_q(y) > 0$ if and only if $\eta(v) \neq 0$ and $v \notin \mathcal{K}_{-q}$. Finally, if L_q is real, then for every $v \in \mathcal{K}_q$, $L_q\overline{v} = \overline{L_q v} = 0$; thus $\overline{v} \in \mathcal{K}_q$, so $\mathcal{E}_q(\overline{v}) = 0$ and the conjugate-linear component cannot produce this escape mechanism. □