ANALYTIC-WALK ROTARY POSITIONAL ENCODINGS FOR GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rotary position encodings make attention sensitive to relative position, but extending them to graphs requires choosing how graph structure enters the rotation. Previous works assign each node a rotation from spectral coordinates, so the rotary factor between two nodes depends only on their endpoints and cannot distinguish the routes connecting them. We introduce *Analytic-Walk Rotary Positional Encodings* (AW-RoPE), which place the rotations on edges and sum the transported features over all walks, so contributions along different routes can reinforce or cancel. An exact variant evaluates the complete sum by a differentiable linear solve, and a sparse variant truncates it at a finite depth. We prove forward and parameter-derivative truncation bounds at fixed inputs and parameters. Both variants act on projected queries and keys, and the sparse recurrence also augments message-passing networks. Across five synthetic tasks both variants reduce nRMSE by 15–58% relative to the strongest baseline, and on real superpixel, peptide and OGB benchmarks the sparse recurrence attains the best mean on every dataset with Performer kernels and on twelve of thirteen datasets with GIN. Analysis shows that AW-RoPE can distinguish routes whose only cue is how two endpoints are connected, while node-wise rotary encodings cannot.

1 INTRODUCTION

Positional encodings (PEs) specify how order enters a Transformer's otherwise permutation-equivariant attention, in two broad styles: *absolute* encodings attach a descriptor to each input (Vaswani et al., 2017), and *relative* encodings modify the interaction between pairs (Shaw et al., 2018). Graphs stress this distinction (Rampášek et al., 2022). With no canonical node ordering, the absolute route replaces indices with structural descriptors such as spectral coordinates and random-walk features (Dwivedi & Bresson, 2020; Kreuzer et al., 2021; Dwivedi et al., 2022a; Lim et al., 2023), while the relative route encodes pairwise graph relationships directly in attention (Ying et al., 2021; Ma et al., 2023).

Rotary positional encoding (RoPE) realizes the relative style by rotating query and key channel pairs so that their dot product depends on relative coordinates (Su et al., 2021), building on relative position representations for sequences. Previous works extend this idea from sequence positions to *Laplacian* spectral coordinates (Reid et al., 2026), bringing global graph information into node-wise rotations while remaining compatible with linear attention.

Graph relationships can involve several routes between the same endpoints. With one phase per node, relative rotations telescope to the endpoint phase difference, so routes receive identical accumulated phases (Figure 1). Spectral coordinates still encode cycles and global structure; the restriction concerns how this information acts on features, motivating learned edge rotations whose contributions along different routes combine.

We introduce *Analytic-Walk Rotary Positional Encodings* (AW-RoPE) to implement this idea. A shared scorer reads the features of both endpoints and learns antisymmetric edge displacements, so reversing an edge reverses its rotation. Successive rotate-and-aggregate steps accumulate phases along walks, and a discounted sum combines their feature contributions, so different routes can reinforce or cancel. *Exact* AW-RoPE computes the complete walk action with a differentiable linear solve, and *Sparse* AW-RoPE truncates the sum after L steps with sparse rotate-and-aggregate recurrences. The

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.

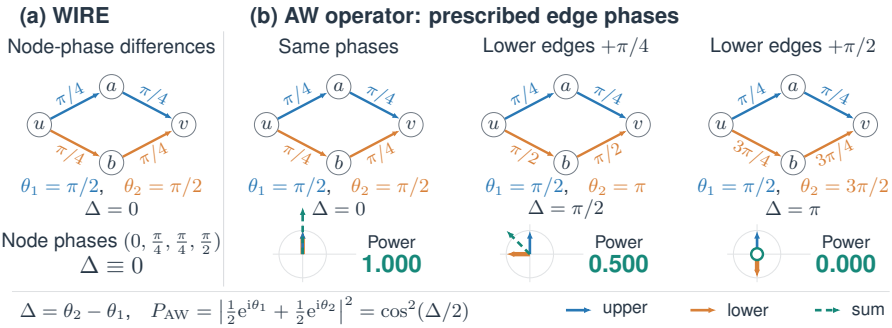

Figure 1: **Node-phase cancellation and controlled edge-phase changes.** Edge labels are phases in radians ($\omega = 1$). (a) With node-wise rotations, every route between the same endpoints accumulates the same phase, so the route gap $\Delta = \theta_2 - \theta_1$ is zero for any node phases. (b) The first AW example uses exactly the same phases. The next two add $\pi/4$ or $\pi/2$ to each forward lower edge. Reverse phases negate. The powers are normalized AW responses with prescribed phases, without fitting the scorer.

operator transforms projected queries and keys before attention, and the sparse recurrence also serves as a residual adapter for message passing (Figure 2).

Our contributions are:

- **A route-sensitive rotary action.** Learned edge rotations make a route's phase a sum of edge terms, so phases accumulated around a cycle (the cycle circulation) can be nonzero, which node-wise phases provably cannot express (Proposition B.2).
- **Algorithms with guarantees.** We prove forward and parameter-derivative error bounds at fixed inputs and parameters for the exact solve and its finite-depth truncation, and empirically verify that the bounds hold at all 30 trained checkpoints (sampled local VJPs on validation inputs).
- **Evaluation across prediction, transfer, and cost.** Both variants obtain lower mean error than the strongest baseline on all five synthetic tasks, with nRMSE reductions of 15–58%, and on two constructed route-discrimination tasks only dynamically generated edge phases succeed: all phase-free, fixed-field, and node-level controls remain at chance, while *Sparse* AW-RoPE reaches 98.9%. The sparse recurrence improves over the strongest baseline in linear attention and over the NoPE baseline in GIN message passing.

2 Related Work

Graph positional and rotary encodings. Laplacian eigenvectors supply node coordinates (Dwivedi & Bresson, 2020), which SAN feeds to spectral attention (Kreuzer et al., 2021) and SignNet makes invariant to sign and basis choices (Lim et al., 2023). LSPE decouples positional from structural streams, covering random-walk encodings (Dwivedi et al., 2022a), and Graph-RoPE's RWPE variant reads features off return probabilities (Reid et al., 2026). Systematic benchmarks compare these encodings across GNN and graph-transformer backbones (Grötschla et al., 2026). Graphormer injects pairwise distances and shortest paths into attention scores (Ying et al., 2021), sparse and tokenized variants extend the family to larger graphs, and GraphGPS couples such attention with local message passing (Shirzad et al., 2023; Chen et al., 2023; Rampášek et al., 2022). RoPE rotates query and key channel pairs so that dot products depend on relative coordinates (Su et al., 2021). WIRE lifts the rotation angles to spectral coordinates, linking rotary phases to grid positions and effective resistance (Reid et al., 2026). In all of these the rotary factor is endpoint-determined. AW-RoPE instead learns edge rotations and sums walk contributions, so phases can vary with intermediate edges.

Learned transport, diffusion, and attention. Vector diffusion maps propagate features through edge transformations (Singer & Wu, 2012), an idea directional graph networks turn into learned edge directions (Beaini et al., 2021). MagNet's magnetic Laplacian similarly attaches complex phases to edge direction (Zhang et al., 2021), and neural sheaf diffusion learns restriction maps between node and edge spaces (Bodnar et al., 2022), with polynomial variants enabling sparse evaluation (Borgi et al., 2025). MagNet prescribes phases from graph directionality for a Hermitian

spectral operator. AW-RoPE instead generates antisymmetric phases from current endpoint features and applies this input-dependent connection to rotary query and key channels through discounted walks. One parameterization supports an exact resolvent and sparse truncation with approximation guarantees, as PPNP and APPNP do for undirected diffusion (Klicpera et al., 2019) (Appendix B.9). Transport is orthogonal to the attention kernel: AW acts on queries and keys before Performer's random features or any other feature map (Choromanski et al., 2021), and a gated residual adapter equips GIN (Xu et al., 2019) with optional ReZero initialization (Bachlechner et al., 2021).

3 METHODS

3.1 PRELIMINARIES: FROM ROPE TO WIRE

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ have node set $\mathcal{V}$ and directed-edge set $\mathcal{E}$, with $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$. An undirected edge is stored in both directions. We use single-head notation and omit projection biases. An attention layer receives node features $\mathbf{H} \in \mathbb{R}^{N\times d}$ of even rotary width d and forms $\mathbf{Q} = \mathbf{H}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{H}\mathbf{W}_K$, $\mathbf{V} = \mathbf{H}\mathbf{W}_V$ with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$; column u of each transposed feature matrix is written $\mathbf{h}_u, \mathbf{q}_u, \mathbf{k}_u, \mathbf{v}_u$. We use $\mathbf{K}$ for keys and the nonnegative integer L for maximum walk depth.

RoPE rotates the $d/2$ adjacent query and key channel pairs. For pair ℓ, write $\mathbf{q}_{u\ell}, \mathbf{k}_{u\ell} \in \mathbb{R}^2$ at node u, and let $\mathbf{R}(\alpha) = \left[\begin{smallmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{smallmatrix}\right]$ denote the counterclockwise rotation. Conjugation cancels the leading angle, $\mathbf{R}(\alpha)^\top \mathbf{R}(\beta) = \mathbf{R}(\beta - \alpha)$. If node u has coordinates $\mathbf{r}_u$, let $\boldsymbol{\nu}_\ell$ be the rotary frequency vector for channel pair ℓ, with the same dimension as $\mathbf{r}_u$ and shared across nodes and between Q and K. The resulting phase is $\phi_{u\ell} = \boldsymbol{\nu}_\ell^\top \mathbf{r}_u$. Applying this identity to the rotated inputs gives

$$(\mathbf{R}(\phi_{u\ell})\mathbf{q}_{u\ell})^\top (\mathbf{R}(\phi_{v\ell})\mathbf{k}_{v\ell}) = \mathbf{q}_{u\ell}^\top \mathbf{R}(\phi_{v\ell} - \phi_{u\ell})\mathbf{k}_{v\ell}. \tag{1}$$

Thus an absolute node phase produces a relative rotary factor. For sequences, $\mathbf{r}_u$ is the token position. WIRE uses graph spectral coordinates, for example components of selected Laplacian eigenvectors, optionally with eigenvalue-dependent scaling (Su et al., 2021; Reid et al., 2026). This identity concerns the raw dot product. The nonlinear feature map used in linear attention does not commute with rotations, so the relative phase is exact only for dot-product attention.

This construction induces an endpoint-dependent rotary phase. Along a route $p = (v_0 = u, \ldots, v_k = v)$ of length k (the number of traversed edges), the differences of node phases telescope:

$$\sum_{j=0}^{k-1} (\phi_{v_{j+1},\ell} - \phi_{v_j,\ell}) = \phi_{v,\ell} - \phi_{u,\ell}. \tag{2}$$

Routes with the same endpoints therefore have identical accumulated phases regardless of their intermediate nodes and edges (Proposition B.2): the route gap $\Delta = \theta_2 - \theta_1$ in Figure 1(a) is always zero. AW-RoPE instead assigns edge rotations and aggregates features along walks, allowing these routes to contribute different phases.

3.2 FROM EDGE ROTATIONS TO MULTI-ROUTE TRANSPORT

The subscript θ denotes dependence on trainable AW parameters including scorer weights, frequencies and decay. A shared scalar scorer s_θ reads the current endpoint features, and $[\cdot,\cdot]$ denotes concatenation along the feature dimension. With a fixed bound $a_{\max} > 0$, define the score difference b_{uv} and bounded edge displacement a_{uv} by

$$b_{uv} = s_\theta([\mathbf{h}_u, \mathbf{h}_v]) - s_\theta([\mathbf{h}_v, \mathbf{h}_u]), \qquad a_{uv} = a_{\max} \tanh(b_{uv}/a_{\max}). \tag{3}$$

Oddness of $\tanh$ gives $a_{vu} = -a_{uv}$, so reversing an edge inverts its rotation. Sharing s_θ makes the displacements equivariant to node relabeling (Lemma B.1) and the operator permutation-equivariant (Proposition B.6). The field is computed once per transport call and held fixed across its walk depths and query and key actions.

Nonnegative edge weights w_{uv} scale neighbor v's contribution to node u. Row normalization gives $\mathbf{P} \in \mathbb{R}^{N\times N}$, $\mathbf{P}_{uv} = w_{uv}/\sum_x w_{ux}$; absent edges and rows without outgoing weight remain zero. Channel ℓ uses a real frequency ω_ℓ and rotation $\mathbf{R}(\omega_\ell a_{uv})$. Pairing real coordinates as $x_\mathrm{r} + \mathrm{i}x_\mathrm{i}$

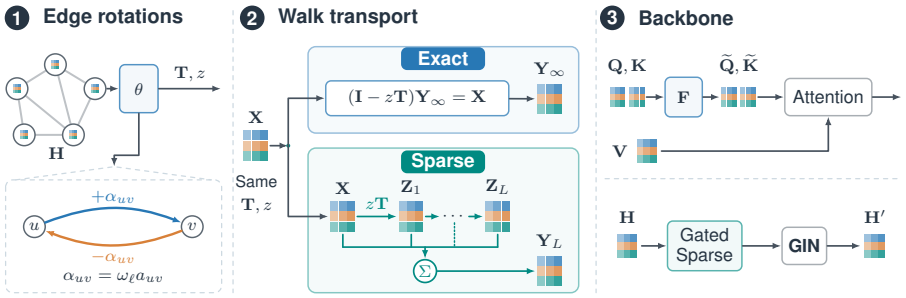

Figure 2: **AW-RoPE implementation.** (1) A shared scorer maps node features $\mathbf{H}$ to edge displacements a_{uv}, with ω_ℓ the channel frequency and θ collecting learned parameters. (2) For fixed input $\mathbf{X}$, one-step transport $\mathbf{T}$ and decay z, *Exact* solves for $\mathbf{Y}_\infty$, with identity $\mathbf{I}$. *Sparse* sums $\mathbf{Z}_k = z^k\mathbf{T}^k\mathbf{X}$ from depth zero through L to approximate it by $\mathbf{Y}_L$. (3) Either operator $\mathbf{F}$ transforms queries and keys, while V bypasses transport. GIN uses a gated *Sparse* residual with output $\mathbf{H}'$. Tiles denote features, and channel indices are suppressed.

(real/imaginary parts, $\mathrm{i}^2 = -1$) represents this rotation by $\mathrm{e}^{\mathrm{i}\omega_\ell a_{uv}}$ without a complex-valued network. Suppressing ℓ, let $\mathbf{X} \in \mathbb{C}^N$ collect channel inputs across nodes. The one-step matrix $\mathbf{T}_\theta \in \mathbb{C}^{N\times N}$ is

$$[\mathbf{T}_\theta]_{uv} = \mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega a_{uv}}, \qquad (\mathbf{T}_\theta\mathbf{X})_u = \sum_{v:(u,v)\in\mathcal{E}} \mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega a_{uv}}\mathbf{X}_v. \tag{4}$$

This is one rotate-and-aggregate step. An index $u \to v$ denotes a walk leaving u, but the row action gathers v's feature into u: we compute $\mathbf{TX}$, not $\mathbf{T}^\top\mathbf{X}$.

Applying the same step twice makes the path interpretation explicit:

$$(\mathbf{T}_\theta^2\mathbf{X})_u = \sum_v \sum_w \mathbf{P}_{uv}\mathbf{P}_{vw}\mathrm{e}^{\mathrm{i}\omega(a_{uv}+a_{vw})}\mathbf{X}_w. \tag{5}$$

Each v defines a length-two route from u to w. For a general walk $p = (v_0 = u, \ldots, v_k = v)$ of $|p| = k$ edges, define its weight $\pi(p) = \prod_{j=0}^{k-1}\mathbf{P}_{v_jv_{j+1}}$ and accumulated displacement $A(p) = \sum_{j=0}^{k-1} a_{v_jv_{j+1}}$. Then

$$(\mathbf{T}_\theta^k\mathbf{X})_u = \sum_v \sum_{\substack{p:u\rightsquigarrow v\\ |p|=k}} \pi(p)\mathrm{e}^{\mathrm{i}\omega A(p)}\mathbf{X}_v. \tag{6}$$

Walks may repeat vertices and edges (induction in Lemma B.4). Unlike Eq. (2), general learned $A(p)$ need not depend only on the endpoints. Summing complex contributions can reinforce or cancel different routes. The total action also mixes amplitudes and is generally not a single orthogonal rotation.

3.3 Complete Walk Action and Sparse Truncation

A learned decay $0 \le z_\theta < 1$ weights length-k walks by z_θ^k, with edge weights inside $\mathbf{T}_\theta^k$. Set $\mathbf{B}_\theta = z_\theta\mathbf{T}_\theta$; the $N \times N$ identity $\mathbf{I} = \mathbf{T}_\theta^0$ retains $\mathbf{X}$ at depth zero. Omitting θ, define $\mathbf{F}_L = \sum_{k=0}^{L}\mathbf{B}^k \in \mathbb{C}^{N\times N}$ and $\mathbf{Y}_L = \mathbf{F}_L\mathbf{X} \in \mathbb{C}^N$. Cancellation gives

$$(\mathbf{I}-\mathbf{B})\sum_{k=0}^{L}\mathbf{B}^k = \mathbf{I} - \mathbf{B}^{L+1}. \tag{7}$$

The limit $L \to \infty$ follows from a geometric bound (Proposition B.3). For matrices, $\|\cdot\|_\infty$ denotes the induced infinity norm, the maximum absolute row sum. Because every phase has modulus one, $\|\mathbf{T}\|_\infty = \max_u \sum_v |\mathbf{T}_{uv}| \le 1$, and hence $\|\mathbf{B}^{L+1}\|_\infty \le z^{L+1} \to 0$. The same geometric bound makes the sums Cauchy, defining the complete walk operator $\mathbf{F}_\infty = \lim_{L\to\infty}\mathbf{F}_L$ and its output $\mathbf{Y}_\infty$. Taking limits in Eq. (7) gives

$$\mathbf{F}_\infty = (\mathbf{I}-\mathbf{B})^{-1}, \qquad \mathbf{Y}_\infty = \mathbf{F}_\infty\mathbf{X}, \qquad (\mathbf{I}-\mathbf{B})\mathbf{Y}_\infty = \mathbf{X}. \tag{8}$$

Exact AW-RoPE evaluates this action up to numerical error with a differentiable linear solve per graph and rotary channel.

Sparse AW-RoPE evaluates the finite sum without constructing a dense matrix. Let $\mathbf{S}^{(k)}$ be the state after k transport steps, before walk-length decay, and $\mathbf{Y}^{(k)}$ the accumulated output through step k. Initialize $\mathbf{S}^{(0)} = \mathbf{X}$ and $\mathbf{Y}^{(0)} = \mathbf{X}$, then compute

$$\mathbf{S}^{(k+1)} = \mathbf{T}_\theta \mathbf{S}^{(k)}, \qquad \mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + z_\theta^{k+1} \mathbf{S}^{(k+1)} \quad (k = 0, \dots, L-1). \tag{9}$$

Induction gives $\mathbf{S}^{(k)} = \mathbf{T}_\theta^k \mathbf{X}$. The weighted states of Figure 2 are $\mathbf{Z}_k = z_\theta^k \mathbf{S}^{(k)}$. The returned $\mathbf{Y}^{(L)} = \mathbf{Y}_L = \mathbf{F}_L \mathbf{X}$ includes every depth from zero through L. Both versions share the field, frequency and decay parameterization, initialize frequencies with the usual RoPE schedule and cap learned decay below one; they evaluate the complete or truncated sum, respectively.

The Exact implementation uses $\mathcal{O}(N^3 d)$ forward work and $\mathcal{O}(N^2 d)$ storage. For $L \geq 1$, *Sparse* uses $\mathcal{O}(L(N+M)d)$ work and $\mathcal{O}((N+M)d)$ streaming storage (Proposition B.12). These costs exclude field construction, the backbone, and additional autodiff storage. *Sparse* transport has graph-linear cost at fixed width and depth, although backpropagation may retain intermediate sparse states and dense attention still costs $\mathcal{O}(N^2 d)$, so graph-linear end-to-end scaling requires a linear-attention or sparse message-passing backbone.

3.4 Using AW-RoPE in the Backbone

For a *Graph Transformer*, $\mathbf{F} = \mathbf{F}_\infty$ or $\mathbf{F}_L$ transforms projected queries and keys with the same edge field. Let $\mathbf{q}^{(\ell)}, \mathbf{k}^{(\ell)} \in \mathbb{C}^N$ collect one paired channel across nodes, and $\mathbf{F}^{(\ell)}$ denote the operator at frequency ω_ℓ. Tildes mark transported features:

$$\widetilde{\mathbf{q}}^{(\ell)} = \mathbf{F}^{(\ell)} \mathbf{q}^{(\ell)}, \qquad \widetilde{\mathbf{k}}^{(\ell)} = \mathbf{F}^{(\ell)} \mathbf{k}^{(\ell)}, \qquad \widetilde{\mathbf{V}} = \mathbf{V}. \tag{10}$$

Converting these channels back to real features gives the matrices $\widetilde{\mathbf{Q}}, \widetilde{\mathbf{K}}$. Ordinary dense attention uses $\mathrm{softmax}(\widetilde{\mathbf{Q}}\widetilde{\mathbf{K}}^\top/\sqrt{d})\mathbf{V}$. For linear attention, the chosen feature map φ is applied to the transported queries and keys before the usual factorized contraction. Transport precedes φ whatever feature map it implements (offsets and batching in Appendix C.5).

For a real feature matrix, operator application means pairing channels, applying each channel's operator and unpacking. Our evaluated GIN adapter uses *Sparse* AW-RoPE with a learned scalar gate γ. Write $\mathbf{H}_{\mathrm{in}}$ for the GIN block's input and $\mathbf{H}_{\mathrm{next}}$ for its output:

$$\mathbf{H}_{\mathrm{in}} = \mathbf{H} + \gamma(\mathbf{F}_L \mathbf{H} - \mathbf{H}), \qquad \mathbf{H}_{\mathrm{next}} = \mathrm{GIN}(\mathbf{H}_{\mathrm{in}}). \tag{11}$$

The gate γ controls transport; ReZero initializes it to zero, recovering the base network. The GIN convolution, residual and normalization are retained.

3.5 Exact–Sparse Convergence with Increasing Walk Depth

The comparison fixes parameters and input. Let $0 < r < 1$ bound $\|\mathbf{B}_\theta\|_\infty$. Subtracting the partial sum from the complete action leaves only omitted walks:

$$\mathbf{F}_\infty - \mathbf{F}_L = \mathbf{B}^{L+1}(\mathbf{I} - \mathbf{B})^{-1}, \qquad \|\mathbf{F}_\infty - \mathbf{F}_L\|_\infty \leq \frac{r^{L+1}}{1-r} \quad \text{if } \|\mathbf{B}_\theta\|_\infty \leq r < 1. \tag{12}$$

Thus *Sparse* converges to *Exact* in operator norm. At differentiability points, $\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]$ denotes the parameter-directional derivative for Euclidean unit $\mathbf{v}$. On a parameter region with uniform $\|\mathbf{B}_\theta\|_\infty \leq r < 1$, let $G \geq 0$ bound $\|\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]\|_\infty$ for every such direction. The matrix product rule gives $\mathrm{D}(\mathbf{B}^k)[\mathbf{v}] = \sum_{j=0}^{k-1} \mathbf{B}^j \mathrm{D}\mathbf{B}[\mathbf{v}] \mathbf{B}^{k-1-j}$. Applying this rule to the remainder gives

$$\|\mathrm{D}(\mathbf{F}_\infty - \mathbf{F}_L)[\mathbf{v}]\|_\infty \leq G \, \frac{r^L((L+1) - Lr)}{(1-r)^2}. \tag{13}$$

Both bounds vanish for fixed $r < 1$ and bounded G (proof in Appendix A); near-unit decay can require greater depth. The derivative covers learned fields, frequencies and decay through $\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]$, but parameter-dependent inputs add product-rule terms outside its scope. Section 4.4 checks outputs and sampled local vector–Jacobian products (VJPs: fixed readout vectors contracted with local Jacobians) at trained checkpoints with fixed parameters and local inputs. The theorem concerns the ordinary unnormalized pair; Appendix B gives normalization, non-backtracking and gating variants. These are same-parameter approximation guarantees; training and prediction are compared experimentally.

4 EXPERIMENTS

4.1 EXPERIMENTAL QUESTIONS AND COMMON PROTOCOL

We ask whether AW improves prediction (Q1), whether phases add gains beyond phase-free propagation (Q2), how closely *Sparse* approximates *Exact* at fixed parameters (Q3), and how AW transfers across datasets and backbones and scales to large graphs (Q4). Variants combine a walk evaluation (*Exact* or *Sparse*) with a field (learned unrestricted, gradient-derived, zero, or fixed). We use the public GraphGPS and Graph-RoPE stack and follow the Graph-RoPE protocol on the synthetic and real-data benchmarks (Rampášek et al., 2022; Reid et al., 2026), with *Sparse* in real-data attention and GIN studies. Test metrics use validation-best checkpoints; independently trained predictive comparisons are separate from same-parameter operator checks, and published WIRE values are external references. Appendices G and E give hyperparameters and selection rules.

Table 1: **Synthetic graph regression.** Test nRMSE ↓; published SEs, local sample SDs. *Exact* and *Sparse* use a four-layer, width-32 dense-attention GraphGPS backbone (Table G.1); NoPE and WIRE are published reference means under this protocol (Reid et al., 2026). Bold compares the matched pair.

Dataset	NoPE	WIRE	*Exact*	*Sparse*
Monochromatic-0	$0.060_{\pm 0.001}$	$0.053_{\pm 0.002}$	$\mathbf{0.044}_{\pm 0.006}$	$0.045_{\pm 0.009}$
Monochromatic-5	$0.087_{\pm 0.001}$	$0.068_{\pm 0.005}$	$\mathbf{0.036}_{\pm 0.003}$	$0.036_{\pm 0.002}$
Monochromatic-10	$0.081_{\pm 0.001}$	$0.063_{\pm 0.002}$	$\mathbf{0.027}_{\pm 0.000}$	$0.032_{\pm 0.009}$
Monochromatic-15	$0.068_{\pm 0.002}$	$0.056_{\pm 0.004}$	$\mathbf{0.026}_{\pm 0.003}$	$0.028_{\pm 0.003}$
Watts–Strogatz SPD	$0.065_{\pm 0.005}$	$0.038_{\pm 0.006}$	$\mathbf{0.016}_{\pm 0.001}$	$0.017_{\pm 0.005}$

4.2 PREDICTIVE PERFORMANCE ON CONTROLLED SYNTHETIC TASKS

Setup. Four Graph-RoPE tasks predict the largest monochromatic connected subgraph on edge-deleted grids; the fifth predicts Watts–Strogatz shortest-path distances. All AW variants receive the same auxiliary LapPE input and are evaluated by normalized RMSE (nRMSE; lower is better). *Exact* and *Sparse* train separately with identical initializations, frozen task-specific configurations and equal parameter counts; *Sparse* uses $L = 8$ or 16 (Table G.1).

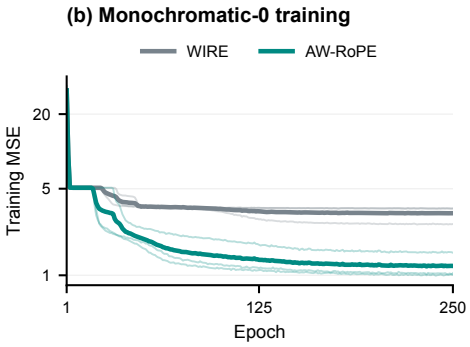

Figure 3: **Training behavior on Monochromatic-0.** Training MSE of the matched WIRE and AW mixing study over all 250 epochs, thin individual curves and thick means, without smoothing.

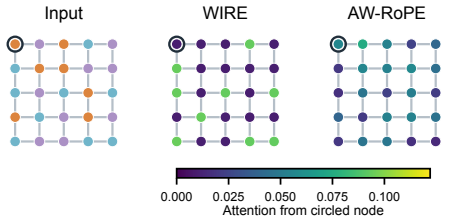

Figure 4: Monochromatic-0 attention from the circled node.

Matched Exact–Sparse results. Table 1 answers Q1. Both arms improve on the published NoPE and WIRE means on all five tasks, and *Sparse*, the practical variant, retains most of *Exact*'s predictive benefit despite truncating the walk sum; the archived fixed-field controls likewise improve on these references (Appendix C.6). *Exact* is faster on small graphs (Table E.2); *Sparse*'s motivation is scaling to larger graph sizes.

Table 2: **Feature-conditioned prediction on unseen cycle sizes.** Test accuracy (%), mean$_{\pm\text{SD}}$ over five matched runs. All arms use a width-16, one-block full softmax-attention backbone. The edge-field arms use maximum walk depth $L = 2$, $z = 0.8$, and $\mathbf{F}^{(\ell)} = \sum_{k=0}^{2} z^k (\mathbf{T}^{(\ell)})^k$ on queries and keys, where $[\mathbf{T}^{(\ell)}]_{uv} = \mathbf{P}_{uv} e^{i\omega_\ell a_{uv}}$. Here $D = I + z\mathbf{P} + z^2\mathbf{P}^2$. The learned static polynomial and WIRE arms use the operators shown below; the WIRE formulas also apply to keys. Architecture and protocol in Appendix I.3.

Method	Field or transport	Accuracy ↑
Mixing (Zero)	$a_{uv} \equiv 0$ (phase-free D)	$50.00_{\pm 0.00}$
WIRE	$\mathbf{q}'_{u\ell} = \mathbf{R}(\boldsymbol{\nu}_\ell^\top \mathbf{r}_u)\mathbf{q}_{u\ell}$	$49.97_{\pm 0.21}$
WIRE + mixing	$\mathbf{q}'_{u\ell} = \mathbf{R}(\boldsymbol{\nu}_\ell^\top \mathbf{r}_u)(D\mathbf{q})_{u\ell}$	$50.47_{\pm 1.07}$
Learned static polynomial	$\mathbf{Q}' = \sum_{k\le 2}(\mathbf{P}^k\mathbf{Q})\mathbf{M}_k^Q$, $\mathbf{K}$ likewise	$50.00_{\pm 0.00}$
Random fixed field	$a_{uv} \sim \mathcal{N}(0, 0.8^2)$, $a_{vu} = -a_{uv}$	$49.88_{\pm 0.17}$
Dynamic local gradient	$a_{uv} = b(\mathbf{h}_v) - b(\mathbf{h}_u)$	$50.00_{\pm 0.00}$
Dynamic nonlocal Flat	$a_i = \tilde{a}_i - \frac{1}{n}\sum_j \tilde{a}_j$	$94.83_{\pm 11.50}$
Sparse AW, phases zeroed	fitted AW with $a_{uv} \equiv 0$	$50.00_{\pm 0.00}$
Sparse AW	$a_{uv} = a_{\max}\tanh(b_{uv}/a_{\max})$	$\mathbf{98.67}_{\pm 2.92}$

4.3 What Does Multi-Route Phase Modeling Add?

Ablation variants. Tables 2 and 3 separate propagation, node phases and edge-field structure. Mixing (Zero) uses $D_{L,z} = \sum_{k=0}^{L} z^k \mathbf{P}^k$ without phases; WIRE uses node-wise rotations, and WIRE + mixing rotates after applying $D_{L,z}$. Table 2 also tests input-independent transport through the learned static polynomial and random fixed field. Dynamic local gradient uses $a_{uv} = b(\mathbf{h}_v) - b(\mathbf{h}_u)$, retaining feature dependence but forcing zero circulation. Dynamic nonlocal Flat projects AW's field onto gradients using the whole cycle, testing zero circulation without the pointwise-potential restriction. "Phases zeroed" sets $a_{uv} = 0$ in a fitted AW model while freezing all learned parameters, testing reliance on phases rather than retraining Mixing (Zero). *Sparse* AW truncates the walk sum; *Exact* AW uses the complete resolvent.

From fixed-graph intervention to a learned task. On a fixed diamond graph with two routes between the same endpoints (Figure I.1, Appendix I.1), varying only the route phase gap Δ gives response power $\cos^2(\Delta/2)$, from reinforcement to cancellation; an endpoint-factorized rotary factor cannot change its magnitude. The learned task uses two length-eight routes joining marked endpoints and distinguishes two internal motifs on one route from one on each. Under endpoint readout, phase-free, WIRE and local-gradient controls remain at chance, while *Sparse* AW reaches 98.85% and *Exact* AW reaches 100% (Table 3). A full attention block also lets mixing and gradient controls solve this task, so it isolates the route-sensitive transport capability.

Table 3: **Constructed route-interference task.** Test accuracy (%), mean$_{\pm\text{SD}}$ over five runs. Endpoint reads a marked node after transport; Attention uses one full block. Truncated walks use $L = 8$, $z = 0.8$; *Exact* uses the complete resolvent. Protocol in Appendix I.2.

Method	Endpoint	Attention
Mixing (Zero)	$50.00_{\pm 0.00}$	$97.92_{\pm 1.77}$
WIRE	$50.00_{\pm 0.00}$	$82.43_{\pm 0.75}$
WIRE + mixing	$50.00_{\pm 0.00}$	$93.89_{\pm 6.97}$
Dynamic local gradient	$49.99_{\pm 0.02}$	$\mathbf{100.00}_{\pm 0.00}$
Sparse AW, phases zeroed	$50.00_{\pm 0.00}$	$50.00_{\pm 0.00}$
Exact AW, phases zeroed	$50.00_{\pm 0.00}$	$50.00_{\pm 0.00}$
Sparse AW	$\mathbf{98.85}_{\pm 2.58}$	$\mathbf{99.94}_{\pm 0.13}$
Exact AW	$\mathbf{100.00}_{\pm 0.00}$	$\mathbf{100.00}_{\pm 0.00}$

Feature dependence in a complete attention network. A second task pairs different feature arrangements on the same anonymous cycle and tests graph sizes absent from training (Table 2). The variants share a one-block attention backbone and change the Q/K transformation as listed in the table, with pointwise adapters matching parameter budgets. AW learns its field from local endpoint features (Eq. (3)). The walk depth is $L = 2$; the learned static polynomial is degree two, and pure WIRE performs no walk propagation. In matched complete attention networks, phase-free, topology-only and pointwise-potential fields provably collide on this construction (Proposition I.1), and WIRE, WIRE + mixing and a random fixed field sit at chance empirically. *Sparse* AW retains

$98.67 \pm 2.92\%$, and zeroing its fitted phases returns every run to 50%. The two tasks expose complementary capabilities, and AW covers both with one local edge-conditioned parameterization. A dynamic nonlocal Flat field also solves the second task, with high per-run variation. Protocols, proofs and a per-arm ladder are collected in Appendices I.1–I.3.

Phases beyond phase-free propagation. In the matched study of Table I.4, WIRE, phase-free mixing $D_{L,z}$, WIRE composed with mixing, and AW-RoPE share backbone, budget and initialization. The matched controls attribute most of the benchmark gains to walk transport itself, with phase learning contributing task-dependent improvements. A matched field study (Appendix H.1) similarly finds the gradient field best on Monochromatic-0; on the real-data field ablations the unrestricted field attains the highest PascalVOC-SP mean, with contributions varying by task on Peptides-struct and MolHIV. Non-flat feature-conditioned transport thus provides additional modeling capacity whose benchmark payoff varies with the task. On fixed validation graphs (Figure 4), WIRE separates input color classes more sharply than AW.

4.4 Exact–Sparse Fidelity and Scalability

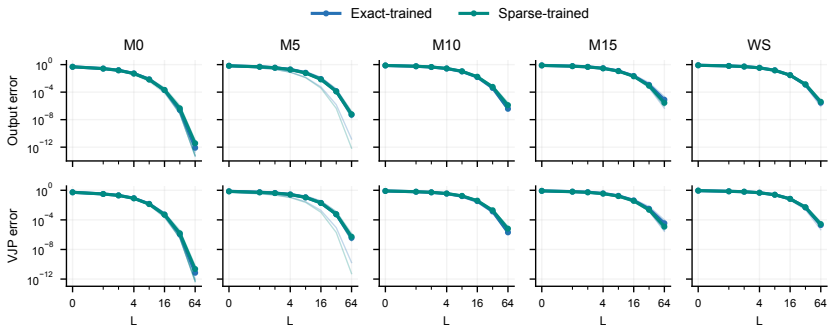

Figure 5: **Approximation at all 30 trained runs.** Columns are the five tasks. Rows are relative output and local VJP error. Each curve compares *Sparse* with *Exact* at identical parameters and local inputs. Thin lines retain all three seeds, and thick lines average seed-level context and probe means. Values below 10^{-16} use a plotting floor. Full settings: Appendix E.1.

Approximation of learned operators. We reuse all 30 validation-best models, comparing *Exact* with *Sparse* at $L \in \{0, 1, 2, 4, 8, 16, 32, 64\}$ on fixed validation inputs in float64. Relative Euclidean errors cover outputs and three fixed-readout vector–Jacobian products (VJPs) with respect to local inputs and AW parameters (Figure 5).

Across the ten task–checkpoint-origin groups, at $L = 8$ the mean relative output errors range from 0.674% to 15.023%, and mean VJP errors from 1.345% to 25.174%. At $L = 64$, the largest group-mean output and VJP errors are 7.851×10^{-6} and 3.504×10^{-5}, respectively. Learned layer decays span 0.468–0.867. These observations support useful trained-operator approximation at sufficient depth on the sampled inputs, while showing that shallow truncation need not already be close to *Exact*. Group means are not worst-case error guarantees.

Exact–Sparse size scaling. We time the production positional-attention layer on isolated H200 GPUs with shared fields, frequencies, decay and inputs, varying graph size, degree and *Sparse* depth (Appendix F). *Exact* is faster on small graphs, but its memory grows quadratically: it needs 72 GiB at 4,096 nodes and runs out of memory at 8,192 and 16,384 nodes, where $L = 16$ *Sparse* needs only 2.7 GiB. Tables F.1 and F.2 supply small-graph comparisons with NoPE and WIRE.

4.5 Generalization Across Datasets and Backbones

Transport without attention. A four-layer GIN with width 32 and dropout 0.2 tests transport without attention. A validation-only 110-trial grid freezes one configuration per task and method, confirmed over 40 runs. Ordinary AW attains the lowest mean on four of the five synthetic tasks

and every AW variant improves on NoPE (Table G.3), with CLUSTER controls in Appendix G.1, Table G.4.

Tasks and attention kernels. Table 4 compares Performer and GIN across datasets, including LRGB (Dwivedi et al., 2022b) and OGB (Hu et al., 2020), using benchmark accuracy, F1, average precision (AP), AUROC and mean absolute error (MAE). Performer references are NoPE and WIRE means from Reid et al. (2026). Local runs use official splits, metrics and benchmark GraphGPS and Graph-RoPE code paths, following the Graph-RoPE training configurations.

Table 4: **AW-RoPE across Performer and GIN backbones.** NoPE and WIRE Performer means are from Reid et al. (2026), except COCO-SP (three local ReLU runs) and MalNet-Tiny (local matched runs for all three methods). AW uses ReLU except PascalVOC-SP and COCO-SP (§), which use FAVOR+. Local entries show mean$_{\pm \text{SD}}$ at validation best; bold compares values within each backbone. PATTERN and CLUSTER use accuracy-SBM for Performer and accuracy for GIN.

Dataset	Metric	Performer backbone			GIN backbone	
		NoPE	WIRE	AW-RoPE	NoPE	AW-RoPE
MNIST	Acc. (%) ↑	97.560	98.100	**98.300**$_{\pm 0.140}$	75.615$_{\pm 1.181}$	**83.890**$_{\pm 0.665}$
CIFAR10	Acc. (%) ↑	70.610	71.150	**72.057**$_{\pm 0.182}$	47.240$_{\pm 0.240}$	**47.520**$_{\pm 0.523}$
PATTERN	Acc. (%) ↑	85.710	86.630	**86.689**$_{\pm 0.000}$	**88.574**$_{\pm 1.907}$	86.443$_{\pm 0.527}$
CLUSTER	Acc. (%) ↑	76.900	77.530	**77.845**$_{\pm 0.140}$	48.549$_{\pm 0.359}$	**64.779**$_{\pm 0.060}$
Peptides-func	AP (%) ↑	64.400	64.900	**65.181**$_{\pm 0.240}$	47.565$_{\pm 0.854}$	**51.937**$_{\pm 0.213}$
Peptides-struct	MAE ↓	0.262	0.257	**0.253**$_{\pm 0.003}$	0.314$_{\pm 0.019}$	**0.291**$_{\pm 0.013}$
PascalVOC-SP§	Macro-F1 ↑	0.367	0.376	**0.385**$_{\pm 0.004}$	0.039$_{\pm 0.000}$	**0.094**$_{\pm 0.010}$
MalNet-Tiny	Acc. (%) ↑	91.87	91.97	**92.73**$_{\pm 0.55}$	81.150$_{\pm 0.212}$	**82.200**$_{\pm 1.414}$
ogbg-molhiv	ROC-AUC ↑	0.776	0.785	**0.792**$_{\pm 0.001}$	0.760$_{\pm 0.002}$	**0.762**$_{\pm 0.023}$
ogbg-molpcba	AP ↑	0.238	0.264	**0.280**$_{\pm 0.003}$	0.169$_{\pm 0.009}$	**0.193**$_{\pm 0.001}$
ogbg-code2	F1 ↑	0.173	0.173	**0.185**$_{\pm 0.002}$	0.176$_{\pm 0.002}$	**0.181**$_{\pm 0.001}$
COCO-SP§	Macro-F1 ↑	0.373$_{\pm 0.004}$	0.370$_{\pm 0.007}$	**0.380**$_{\pm 0.003}$	0.025$_{\pm 0.003}$	**0.077**$_{\pm 0.010}$

Attention results. On the Performer panel, AW-RoPE improves on the published WIRE means on every dataset, most strongly on CIFAR10 and CLUSTER (Table 4), and its COCO-SP FAVOR+ average exceeds the local ReLU baselines. Matched zero-field and gradient controls with sparse propagation retained (Appendix H.1) isolate the phase contribution on PascalVOC-SP, Peptides-struct and MolHIV. Figure 3 shows the matched mixing study under one backbone and training setting, with AW lower in training MSE throughout much of the trajectory.

GIN transfer results. The GIN columns of Table 4 contain all 52 runs across thirteen datasets. AW improves the mean on twelve tasks, most strongly on CLUSTER and MNIST, while PATTERN favors NoPE and MolHIV is comparable to it. Together with the synthetic GIN results, walk transport transfers to message passing without attention.

4.6 Limitations and Practical Trade-offs

Walk propagation adds computational overhead, so AW trains noticeably slower than WIRE, and *Exact* scales quadratically in memory with graph size (Appendix H). Phase utility is task-dependent (Section 4.3), the exact relative-phase reading applies to dot-product attention, and shuffled-edge controls remain future work.

5 Conclusion

Node-wise rotary phases telescope along every route and cannot distinguish how the graph connects two endpoints. AW-RoPE instead learns antisymmetric edge rotations and sums transported features over walks, letting routes reinforce or cancel. *Exact* evaluates the full walk action by a linear solve, and *Sparse* truncates it with provable forward and derivative bounds, matching *Exact* at all 30 trained checkpoints. Both variants improve on the strongest baseline across the five synthetic tasks (15–58% nRMSE reduction) and transfer to real superpixel, peptide and OGB benchmarks, attaining the best mean on twelve of thirteen GIN tasks and every Performer task. Future work includes richer fields and reducing walk-propagation complexity and runtime.

AI USE STATEMENT

ChatGPT is used for literature and proof checks, and paper proofreading. The authors are responsible for verifying all text, mathematics, code, figures and protocols.

REPRODUCIBILITY STATEMENT

The full implementation, configuration files, experiment protocols and tests are publicly available at `https://anonymous.4open.science/r/AW-RoPE-55EA`.

REFERENCES

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pp. 1352–1361. PMLR, 2021.

Dominique Beaini, Saro Passaro, Vincent Létourneau, William L. Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. In *International Conference on Machine Learning*, 2021.

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*, 2022.

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

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations*, 2023.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020. URL `https://arxiv.org/abs/2012.09699`.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a. URL `https://openreview.net/forum?id=wTTjnvGphYj`.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22326–22340, 2022b.

Florian Grötschla, Jiaqing Xie, and Roger Wattenhofer. Benchmarking positional encodings for gnns and graph transformers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations*, 2019.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/b4fd1d2cb085390fbbadae65e07876a7-Abstract.html.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Q-UHqMorzil.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip H. S. Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, 2023.

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, pp. 14501–14515, 2022.

Isaac Reid, Arijit Sehanobish, Cederik Höfs, Bruno Mlodozeniec, Leonhard Vulpius, Federico Barbero, Adrian Weller, Krzysztof Choromanski, Richard E. Turner, and Petar Veličković. Rotary position encodings for graphs. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.22259.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of NAACL-HLT (Short Papers)*, pp. 464–468, 2018. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074/.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, 2023.

Amit Singer and Hau-tieng Wu. Vector diffusion maps and the connection laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012. doi: 10.1002/cpa.21395. URL https://arxiv.org/abs/1102.0075.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

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. URL https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. MagNet: A neural network for directed graphs. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/e32084632d369461572832e6582aac36-Abstract.html.

The appendices present convergence bounds (A), operator properties with proofs (B), implementation (C) and numerical checks (D), followed by the matched synthetic study (E), scaling (F) and further ablations.

A Exact–Sparse AW-RoPE: A Step-by-Step Convergence Proof

We establish forward and parameter-derivative convergence of the finite walk sum. All comparisons use the same parameters θ, graph, and input. For one complex rotary channel, write $\mathbf{B}_\theta = z_\theta \mathbf{T}_\theta$, $\mathbf{F}_L = \sum_{k=0}^{L} \mathbf{B}_\theta^k$, and $\mathbf{F}_\infty = (\mathbf{I} - \mathbf{B}_\theta)^{-1}$ once existence is established. The proof concerns ordinary, unnormalized AW-RoPE.

A.1 Step 1: Specify the norms and contraction

For a complex vector, $\|\mathbf{x}\|_\infty = \max_u |\mathbf{x}_u|$; the induced matrix norm is $\|\mathbf{A}\|_\infty = \max_u \sum_v |\mathbf{A}_{uv}|$. For several channels we take the maximum across channels as well. These norms satisfy $\|\mathbf{A}\mathbf{x}\|_\infty \le \|\mathbf{A}\|_\infty \|\mathbf{x}\|_\infty$ and $\|\mathbf{A}\mathbf{C}\|_\infty \le \|\mathbf{A}\|_\infty \|\mathbf{C}\|_\infty$. The latter follows by expanding each row:

$$\sum_v |(\mathbf{A}\mathbf{C})_{uv}| \le \sum_v \sum_w |\mathbf{A}_{uw}||\mathbf{C}_{wv}| \le \left(\sum_w |\mathbf{A}_{uw}| \right) \|\mathbf{C}\|_\infty.$$

Since $|\mathrm{e}^{\mathrm{i}\omega a_{uv}}| = 1$, nonnegative row-substochastic $\mathbf{P}$ gives

$$\|\mathbf{T}_\theta\|_\infty \le \max_u \sum_v \mathbf{P}_{uv} \le 1, \qquad \|\mathbf{B}_\theta\|_\infty \le z_\theta. \tag{A.1}$$

Parallel edges are summed before forming a matrix entry; the first inequality remains valid by the triangle inequality. Isolated rows are zero. Assume a uniform cap $z_\theta \le r < 1$ on the parameter region of interest. It follows inductively that $\|\mathbf{B}_\theta^k\|_\infty \le r^k$.

A.2 Step 2: Show that the finite sums have a limit

For integers $L' > L$, subtracting the partial sums and using Step 1 gives

$$\|\mathbf{F}_{L'} - \mathbf{F}_L\|_\infty \le \sum_{k=L+1}^{L'} \|\mathbf{B}_\theta^k\|_\infty \le \sum_{k=L+1}^{\infty} r^k = \frac{r^{L+1}}{1-r}.$$

The right-hand side tends to zero independently of L'. Finite-dimensional matrix spaces are complete, so $(\mathbf{F}_L)$ is Cauchy and has a limit $\mathbf{F}$. Also,

$$\|\mathbf{F}_L\|_\infty \le \sum_{k=0}^{L} r^k = \frac{1 - r^{L+1}}{1-r}, \qquad \|\mathbf{F}\|_\infty \le \frac{1}{1-r}. \tag{A.2}$$

A.3 Step 3: Identify the limit with the complete linear solve

This identification does not require assuming the inverse beforehand. Multiplying the finite polynomial and canceling adjacent powers gives

$$(\mathbf{I} - \mathbf{B}_\theta)\mathbf{F}_L = (\mathbf{I} + \mathbf{B}_\theta + \cdots + \mathbf{B}_\theta^L) - (\mathbf{B}_\theta + \mathbf{B}_\theta^2 + \cdots + \mathbf{B}_\theta^{L+1}) \tag{A.3}$$

$$= \mathbf{I} - \mathbf{B}_\theta^{L+1}. \tag{A.4}$$

The same identity holds for $\mathbf{F}_L(\mathbf{I} - \mathbf{B}_\theta)$. Matrix multiplication is continuous, and $\|\mathbf{B}_\theta^{L+1}\|_\infty \le r^{L+1} \to 0$. Taking limits therefore gives $(\mathbf{I} - \mathbf{B}_\theta)\mathbf{F} = \mathbf{F}(\mathbf{I} - \mathbf{B}_\theta) = \mathbf{I}$. Thus $\mathbf{I} - \mathbf{B}_\theta$ is invertible and

$$\mathbf{F} = \mathbf{F}_\infty = (\mathbf{I} - \mathbf{B}_\theta)^{-1} = \sum_{k=0}^{\infty} \mathbf{B}_\theta^k, \qquad (\mathbf{I} - \mathbf{B}_\theta)\mathbf{Y}_\infty = \mathbf{X}. \tag{A.5}$$

The *Exact* implementation solves this finite-dimensional system. The infinitely many walks describe its meaning, not its algorithm.

A.4 Step 4: Derive the forward truncation error

Define the remainder $\mathbf{E}_L = \mathbf{F}_\infty - \mathbf{F}_L$. The finite identity above yields

$$(\mathbf{I} - \mathbf{B}_\theta)\mathbf{E}_L = \mathbf{I} - (\mathbf{I} - \mathbf{B}_\theta)\mathbf{F}_L = \mathbf{B}_\theta^{L+1}.$$

Multiplying by the inverse and applying Step 2 gives

$$\mathbf{E}_L = \mathbf{F}_\infty \mathbf{B}_\theta^{L+1}, \qquad \|\mathbf{E}_L\|_\infty \le \underbrace{\frac{r^{L+1}}{1-r}}_{\epsilon_L}. \tag{A.6}$$

Consequently $\|\mathbf{Y}_\infty - \mathbf{Y}_L\|_\infty \le \epsilon_L \|\mathbf{X}\|_\infty$. This proves convergence of the feature action as L increases.

A.5 Step 5: Differentiate a matrix power without commuting factors

Let $\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]$ denote a directional derivative in a parameter direction $\mathbf{v}$ with $\|\mathbf{v}\|_2 = 1$. Matrix products are generally noncommutative:

$$\mathrm{D}(\mathbf{B}^2)[\mathbf{v}] = \mathrm{D}\mathbf{B}[\mathbf{v}]\mathbf{B} + \mathbf{B}\mathrm{D}\mathbf{B}[\mathbf{v}], \qquad \mathrm{D}(\mathbf{B}^3)[\mathbf{v}] = \mathrm{D}\mathbf{B}[\mathbf{v}]\mathbf{B}^2 + \mathbf{B}\mathrm{D}\mathbf{B}[\mathbf{v}]\mathbf{B} + \mathbf{B}^2\mathrm{D}\mathbf{B}[\mathbf{v}].$$

Inductively, differentiating $\mathbf{B}^{k+1} = \mathbf{B}^k\mathbf{B}$ gives

$$\mathrm{D}(\mathbf{B}^k)[\mathbf{v}] = \sum_{j=0}^{k-1} \mathbf{B}^j \mathrm{D}\mathbf{B}[\mathbf{v}] \mathbf{B}^{k-1-j}. \tag{A.7}$$

The induction step appends $\mathbf{B}$ to every old term and adds $\mathbf{B}^k\mathrm{D}\mathbf{B}[\mathbf{v}]$, which gives exactly the $k+1$ required terms. Suppose $\|\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]\|_\infty \le G$. There are k summands, each bounded by $r^j G r^{k-1-j} = G r^{k-1}$, so

$$\|\mathrm{D}(\mathbf{B}^k)[\mathbf{v}]\|_\infty \le k G r^{k-1}. \tag{A.8}$$

A.6 Step 6: Differentiate the complete action and the remainder

Differentiate $(\mathbf{I} - \mathbf{B}_\theta)\mathbf{F}_\infty = \mathbf{I}$ using the product rule:

$$-\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]\mathbf{F}_\infty + (\mathbf{I} - \mathbf{B}_\theta)\mathrm{D}\mathbf{F}_\infty[\mathbf{v}] = 0.$$

Solving for the derivative gives

$$\mathrm{D}\mathbf{F}_\infty[\mathbf{v}] = \mathbf{F}_\infty \mathrm{D}\mathbf{B}_\theta[\mathbf{v}] \mathbf{F}_\infty, \qquad \|\mathrm{D}\mathbf{F}_\infty[\mathbf{v}]\|_\infty \le \frac{G}{(1-r)^2}.$$

Now differentiate the finite remainder identity from Step 4:

$$\mathrm{D}\mathbf{E}_L[\mathbf{v}] = \mathrm{D}\mathbf{F}_\infty[\mathbf{v}]\mathbf{B}_\theta^{L+1} + \mathbf{F}_\infty \mathrm{D}(\mathbf{B}_\theta^{L+1})[\mathbf{v}].$$

Applying Steps 2 and 5 to the two terms separately yields

$$\|\mathrm{D}\mathbf{E}_L[\mathbf{v}]\|_\infty \le \frac{G r^{L+1}}{(1-r)^2} + \frac{(L+1) G r^L}{1-r} \tag{A.9}$$

$$= G\, \frac{r^L\{r + (L+1)(1-r)\}}{(1-r)^2} \tag{A.10}$$

$$= \underbrace{G\, \frac{r^L((L+1) - Lr)}{(1-r)^2}}_{\eta_L}. \tag{A.11}$$

The derivative follows from the finite remainder identity and the inverse map on nonsingular matrices. For fixed $0 < r < 1$, both r^L and Lr^L tend to zero, so $\eta_L \to 0$. At $r = 0$, take the continuous limit.

A.7 Step 7: Include parameter-dependent features

In a network, $\mathbf{X} = \mathbf{X}_\theta$ may itself depend on the parameters. Writing $\mathbf{Y}_\infty - \mathbf{Y}_L = \mathbf{E}_L \mathbf{X}_\theta$ gives

$$\mathrm{D}(\mathbf{Y}_\infty - \mathbf{Y}_L)[\mathbf{v}] = \mathrm{D}\mathbf{E}_L[\mathbf{v}]\mathbf{X}_\theta + \mathbf{E}_L \mathrm{D}\mathbf{X}_\theta[\mathbf{v}].$$

If $\|\mathbf{X}_\theta\|_\infty \le C_X$ and $\|\mathrm{D}\mathbf{X}_\theta[\mathbf{v}]\|_\infty \le C_D$, then

$$\|\mathbf{Y}_\infty - \mathbf{Y}_L\|_\infty \le \epsilon_L C_X, \qquad \|\mathrm{D}(\mathbf{Y}_\infty - \mathbf{Y}_L)[\mathbf{v}]\|_\infty \le \eta_L C_X + \epsilon_L C_D. \tag{A.12}$$

The second term accounts for preceding learnable layers.

Theorem A.1 (Same-parameter forward and derivative convergence)**.** *On a parameter region with $\|\mathbf{B}_\theta\|_\infty \le r < 1$, assume $\mathbf{B}_\theta$ is differentiable at the points considered and $\|\mathrm{D}\mathbf{B}_\theta[\mathbf{v}]\|_\infty \le G$ for every unit direction $\mathbf{v}$. Then the forward and derivative remainders satisfy Eqs.* (A.6) *and* (A.11)*, uniformly wherever these assumptions hold. If the input and its derivative are bounded as in Step 7, Eq.* (A.12) *holds as well. All these errors tend to zero as $L \to \infty$.*

The result follows from Steps 1–7.

A.8 Step 8: Check what the assumptions mean for learned AW-RoPE

The implemented decay is a sigmoid multiplied by a cap below one; this supplies a uniform r, although a cap close to one makes the worst-case bound loose. The derivative assumption is additional. For fixed $\mathbf{P}$, write $\alpha_{uv,\ell} = \omega_\ell a_{uv}$. Then

$$\mathrm{D}\mathbf{B}[\mathbf{v}] = (\mathrm{D}z[\mathbf{v}])\mathbf{T} + z\,\mathrm{D}\mathbf{T}[\mathbf{v}], \qquad \mathrm{D}\alpha_{uv,\ell}[\mathbf{v}] = (\mathrm{D}\omega_\ell[\mathbf{v}])a_{uv} + \omega_\ell \mathrm{D}a_{uv}[\mathbf{v}],$$

and

$$[\mathrm{D}\mathbf{T}[\mathbf{v}]]_{uv} = \mathrm{i}\mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\alpha_{uv,\ell}}\mathrm{D}\alpha_{uv,\ell}[\mathbf{v}].$$

If $\mathbf{P}$ is learnable, add $(\mathrm{D}\mathbf{P}_{uv}[\mathbf{v}])\mathrm{e}^{\mathrm{i}\alpha_{uv,\ell}}$. These formulas identify the field, frequency, decay, and optional transition-weight derivatives that G must bound. Bounding a_{uv} alone does not bound arbitrary scorer or frequency derivatives. Smooth scorers satisfy the differentiability requirement; piecewise-smooth networks require restricting the statement to differentiability points.

A.9 Step 9: A conditional downstream loss-gradient bound

For a scalar objective $\ell_\theta(\mathbf{Y})$, use the feature norm above and its dual (the sum of two-dimensional block norms when complex channels are viewed as real pairs). Assume the downstream derivative $g_\theta(\mathbf{Y}) = \mathrm{D}_Y \ell_\theta(\mathbf{Y})$ has dual norm at most A and is C_L-Lipschitz in $\mathbf{Y}$. Assume the explicit parameter derivative $h_\theta(\mathbf{Y})[\mathbf{v}] = \partial_\theta \ell_\theta(\mathbf{Y})[\mathbf{v}]$ is C_θ-Lipschitz, and $\|\mathrm{D}\mathbf{Y}_L[\mathbf{v}]\|_\infty \le C_Y$. By the chain rule, the difference of the two total derivatives is

$$\begin{aligned}
&\mathrm{D}[\ell_\theta(\mathbf{Y}_\infty) - \ell_\theta(\mathbf{Y}_L)][\mathbf{v}] && \text{(A.13)}\\
&= h_\theta(\mathbf{Y}_\infty)[\mathbf{v}] - h_\theta(\mathbf{Y}_L)[\mathbf{v}] + g_\theta(\mathbf{Y}_\infty)\mathrm{D}\mathbf{Y}_\infty[\mathbf{v}] - g_\theta(\mathbf{Y}_L)\mathrm{D}\mathbf{Y}_L[\mathbf{v}] && \text{(A.14)}\\
&= h_\theta(\mathbf{Y}_\infty)[\mathbf{v}] - h_\theta(\mathbf{Y}_L)[\mathbf{v}] + g_\theta(\mathbf{Y}_\infty)(\mathrm{D}\mathbf{Y}_\infty[\mathbf{v}] - \mathrm{D}\mathbf{Y}_L[\mathbf{v}]) && \text{(A.15)}\\
&\quad + (g_\theta(\mathbf{Y}_\infty) - g_\theta(\mathbf{Y}_L))\mathrm{D}\mathbf{Y}_L[\mathbf{v}]. && \text{(A.16)}
\end{aligned}$$

The first term is bounded by $C_\theta \epsilon_L C_X$, because h_θ is C_θ-Lipschitz in $\mathbf{Y}$ and Eq. (A.12) gives $\|\mathbf{Y}_\infty - \mathbf{Y}_L\|_\infty \le \epsilon_L C_X$. The second is bounded by $A(\eta_L C_X + \epsilon_L C_D)$, by the dual norm A of g_θ and the derivative bound of Eq. (A.12). The third is bounded by $C_L C_Y \epsilon_L C_X$, because g_θ is C_L-Lipschitz, the same forward bound applies to $\|g_\theta(\mathbf{Y}_\infty) - g_\theta(\mathbf{Y}_L)\|$, and $\|\mathrm{D}\mathbf{Y}_L[\mathbf{v}]\|_\infty \le C_Y$. Therefore

$$|\mathrm{D}[\ell_\theta(\mathbf{Y}_\infty) - \ell_\theta(\mathbf{Y}_L)][\mathbf{v}]| \le A(\eta_L C_X + \epsilon_L C_D) + (C_L C_Y + C_\theta)\epsilon_L C_X. \tag{A.17}$$

For a multilayer model, these constants must hold for the composed downstream network, including attention normalization. The transport theorem alone does not supply them.

Scope. The bounds compare the complete and truncated operators at common parameters and inputs. They apply to zero, gradient and non-flat fields; independent optimization trajectories and test performance are evaluated experimentally.

B Operator Properties and Proofs

B.1 Problem Setup and Construction

Let $\mathcal{G} = (V, E)$ be a graph with $N = |V|$ nodes and $M = |E|$ *directed* edges. An undirected edge is stored in both orientations. The cycle-flatness statements below use an undirected support with symmetric nonnegative weights $w_{uv} = w_{vu}$. Connectivity means connectivity through positive-weight edges. The sparse walk identities also apply to directed nonnegative row-substochastic transitions, without implying that their connection Laplacian is Hermitian. The node features and paired-channel attention interface follow Section 3; the results below use the same operators with decay and frequency displayed explicitly. Each statement is followed by its proof.

B.2 Antisymmetric Rotary Edge Connections

The connection interface accepts any permutation-equivariant antisymmetric scalar field. Learned AW-RoPE constructs one from the current hidden state; AW-static and AW-flat instead use the explicitly specified fixed fields. For every directed edge $u \to v$, a shared scalar network s_θ first computes an ordered score, which we antisymmetrize explicitly:

$$c_{uv} = s_\theta([\mathbf{h}_u, \mathbf{h}_v]), \tag{B.1}$$

$$\bar{a}_{uv} = c_{uv} - c_{vu}, \tag{B.2}$$

$$a_{uv} = a_{\max} \tanh(\bar{a}_{uv}/a_{\max}). \tag{B.3}$$

Here c_{uv} is the ordered score and $\bar{a}_{uv}$ is the score difference denoted by b_{uv} in Eq. (3). The odd $\tanh$ bounds the displacement. Other supported *unclipped* fields are a potential difference $a_{uv} = g_\theta(\mathbf{h}_v) - g_\theta(\mathbf{h}_u)$ and, when coordinates are available, a geometric projection $a_{uv} = \mathbf{w}^\top(\mathbf{x}_v - \mathbf{x}_u)$. Applying the nonlinear bound to either still preserves antisymmetry but need not preserve the exact gradient property; gradient-field theory and the corresponding ablation therefore disable clipping. Learned variants use Eq. (B.3) unless stated otherwise.

Lemma B.1 (Antisymmetry and relabeling)**.** *Equation* (B.3) *satisfies* $a_{vu} = -a_{uv}$. *Because the same scorer is used on every edge, relabeling the nodes only relabels the edge field.*

Proof of Lemma B.1. Swapping the edge orientation gives

$$\bar{a}_{vu} = s_\theta([\mathbf{h}_v, \mathbf{h}_u]) - s_\theta([\mathbf{h}_u, \mathbf{h}_v]) \tag{B.4}$$

$$= -\bar{a}_{uv}. \tag{B.5}$$

Since $\tanh$ is odd,

$$a_{vu} = a_{\max} \tanh(\bar{a}_{vu}/a_{\max}) \tag{B.6}$$

$$= a_{\max} \tanh(-\bar{a}_{uv}/a_{\max}) \tag{B.7}$$

$$= -a_{uv}. \tag{B.8}$$

For a permutation π, the ordered pair associated with the relabeled edge $\pi(u) \to \pi(v)$ contains exactly the original features $[\mathbf{h}_u, \mathbf{h}_v]$. Because the scorer parameters are shared, its output is unchanged and $a'_{\pi(u)\pi(v)} = a_{uv}$. □

Let $w_{uv} \geq 0$ be an optional edge weight. For a positive weighted degree, define the row-normalized transition

$$\mathbf{P}_{uv} = \frac{w_{uv}}{\sum_{x:(u,x)\in E} w_{ux}}. \tag{B.9}$$

A zero-weight row is set to zero. Adjacent real channels are viewed as one complex channel. Channel ℓ uses

$$\omega_\ell = b^{-2\ell/d}, \qquad \ell = 0, \ldots, d/2 - 1, \qquad b = 10{,}000, \tag{B.10}$$

initialized as in RoPE (Su et al., 2021). Learned AW-RoPE may optimize these frequencies after initialization.

Proposition B.2 (Node-wise factorization obstruction)**.** *If a scalar rotary transport can be implemented by one phase per node, $\chi_\ell(u,v) = \overline{g_\ell(u)} g_\ell(v)$ with $|g_\ell(u)| = 1$, then its product around every directed cycle is one. Consequently, a transport with nontrivial cycle circulation cannot be represented exactly by standard node-wise Q/K rotations.*

Proof of Proposition B.2. Let a directed cycle be $C = (v_0, v_1, \dots, v_m)$ with $v_m = v_0$. Assume the pair transport has a scalar node factorization $\chi_\ell(u,v) = \overline{g_\ell(u)} g_\ell(v)$ and $|g_\ell(u)| = 1$. Multiplying its factors in cycle order gives

$$\prod_{j=0}^{m-1} \chi_\ell(v_j, v_{j+1}) = \prod_{j=0}^{m-1} \overline{g_\ell(v_j)} g_\ell(v_{j+1}) \tag{B.11}$$

$$= \overline{g_\ell(v_0)} g_\ell(v_1) \overline{g_\ell(v_1)} g_\ell(v_2) \cdots \overline{g_\ell(v_{m-1})} g_\ell(v_0) \tag{B.12}$$

$$= \prod_{j=0}^{m-1} |g_\ell(v_j)|^2 \tag{B.13}$$

$$= 1. \tag{B.14}$$

Every intermediate node phase cancels with its conjugate, including the initial/final node because $v_m = v_0$. Thus non-unit cycle product contradicts the assumed node factorization. This is why exact non-flat U(1) transport is necessarily relational. AW-RoPE remains node-level in its stored features by evaluating the edge operator action; it does not claim that the underlying pair kernel factorizes. □

B.3 AW-RoPE: Sparse Analytic Walk Transport

For one frequency ω, define

$$[\mathbf{T}_\omega]_{uv} = \mathbf{P}_{uv} \mathrm{e}^{\mathrm{i}\omega a_{uv}}. \tag{B.15}$$

Thus, for a complex feature $\mathbf{X}$,

$$(\mathbf{T}_\omega \mathbf{X})_u = \sum_{v:(u,v)\in E} \mathbf{P}_{uv} \mathrm{e}^{\mathrm{i}\omega a_{uv}} \mathbf{X}_v. \tag{B.16}$$

The phase is a rotation, while $\mathbf{P}_{uv}$ controls the magnitude.

Proposition B.3 (Convergence of the analytic walk)**.** *For every ω, $\|\mathbf{T}_\omega\|_\infty \le 1$. Hence for $0 \le z < 1$,*

$$\mathbf{F}_{z,\omega} = (\mathbf{I} - z\mathbf{T}_\omega)^{-1} = \sum_{k=0}^{\infty} z^k \mathbf{T}_\omega^k \tag{B.17}$$

exists and converges in the induced infinity norm.

Proposition B.3 is the fixed-parameter case of Steps 1–3 in Appendix A, with $\mathbf{B} = z\mathbf{T}_\omega$. Those steps establish contraction, convergence and invertibility before identifying the limit; no second proof is needed here.

We parameterize z as a sigmoid times $1 - \varepsilon$, with fixed $0 < \varepsilon < 1$, to keep it in the convergence interval during training.

B.4 Matrix Powers as Walk Sums

For a length-k walk $p = (v_0 = u, v_1, \dots, v_k = v)$, define

$$\pi(p) = \prod_{j=0}^{k-1} \mathbf{P}_{v_j v_{j+1}}, \qquad A(p) = \sum_{j=0}^{k-1} a_{v_j v_{j+1}}. \tag{B.18}$$

Lemma B.4 (Walk expansion)**.** *For every integer $k \ge 1$,*

$$[\mathbf{T}_\omega^k]_{uv} = \sum_{p: u \rightsquigarrow v,\, |p|=k} \pi(p) \mathrm{e}^{\mathrm{i}\omega A(p)}. \tag{B.19}$$

Consequently,

$$[\mathbf{F}_{z,\omega}]_{uv} = \sum_{p:u\rightsquigarrow v} z^{|p|}\pi(p)\mathrm{e}^{\mathrm{i}\omega A(p)}. \tag{B.20}$$

Proof of Lemma B.4. For $k = 1$, Eq. (B.19) is the definition of $\mathbf{T}_\omega$. Assume it holds for length k. Matrix multiplication gives

$$[\mathbf{T}_\omega^{k+1}]_{uv} = \sum_x [\mathbf{T}_\omega^k]_{ux}[\mathbf{T}_\omega]_{xv} \tag{B.21}$$

$$= \sum_x \sum_{p:u\rightsquigarrow x,\,|p|=k} \pi(p)\mathrm{e}^{\mathrm{i}\omega A(p)}\mathbf{P}_{xv}\mathrm{e}^{\mathrm{i}\omega a_{xv}} \tag{B.22}$$

$$= \sum_x \sum_{p:u\rightsquigarrow x,\,|p|=k} \big(\pi(p)\mathbf{P}_{xv}\big)\mathrm{e}^{\mathrm{i}\omega(A(p)+a_{xv})}. \tag{B.23}$$

Appending $x \to v$ is a bijection between the terms on the last line and all length-$(k+1)$ walks from u to v. This proves the induction step. The zero-length walk exists only for $u = v$ and has $\pi(p) = 1$, $A(p) = 0$. Absolute convergence follows from $\sum_{k\geq 0} z^k[\mathbf{P}^k]_{uv} \leq (1-z)^{-1}$. Substituting the fixed-length identity into Eq. (B.17) yields

$$[\mathbf{F}_{z,\omega}]_{uv} = \sum_{k=0}^{\infty} z^k [\mathbf{T}_\omega^k]_{uv} \tag{B.24}$$

$$= \sum_{k=0}^{\infty} \sum_{p:u\rightsquigarrow v,\,|p|=k} z^k\pi(p)\mathrm{e}^{\mathrm{i}\omega A(p)} \tag{B.25}$$

$$= \sum_{p:u\rightsquigarrow v} z^{|p|}\pi(p)\mathrm{e}^{\mathrm{i}\omega A(p)}. \tag{B.26}$$

□

Equation (B.20) makes the modeling semantics explicit. The magnitude $z^{|p|}\pi(p)$ discounts long or unlikely walks. The phase adds signed edge displacements. Different routes are summed as complex numbers, so the graph itself determines whether they reinforce or cancel.

B.5 Sparse Truncation, Error, and Stability

We approximate the resolvent with

$$\mathbf{F}_{z,\omega}^{(L)}\mathbf{X} = \sum_{k=0}^{L} z^k \mathbf{T}_\omega^k \mathbf{X}. \tag{B.27}$$

Set $\mathbf{X}^{(0)} = \mathbf{X}$ and $\mathbf{Y}^{(0)} = \mathbf{X}$. In real paired-channel form, one step is

$$\mathbf{X}_u^{(k+1)} = \sum_{v:(u,v)\in E} \mathbf{P}_{uv}\mathbf{R}(\omega a_{uv})\mathbf{X}_v^{(k)}, \tag{B.28}$$

$$\mathbf{Y}^{(k+1)} = \mathbf{Y}^{(k)} + z^{k+1}\mathbf{X}^{(k+1)}, \tag{B.29}$$

Here $\mathbf{R}(\alpha)$ is the counterclockwise rotation by angle α (Section 3.1). By induction, $\mathbf{X}^{(k)} = \mathbf{T}_\omega^k\mathbf{X}$; inserting this identity into Eq. (B.29) gives Eq. (B.27).

Proposition B.5 (Truncation and feature bounds)**.** *For* $0 \leq z < 1$*,*

$$\left\|\mathbf{F}_{z,\omega} - \mathbf{F}_{z,\omega}^{(L)}\right\|_\infty \leq \frac{z^{L+1}}{1-z}. \tag{B.30}$$

In addition,

$$\left\|\mathbf{F}_{z,\omega}^{(L)}\mathbf{X}\right\|_\infty \leq \frac{1-z^{L+1}}{1-z}\|\mathbf{X}\|_\infty. \tag{B.31}$$

For the first bound, apply Step 4 of Appendix A with $r = z$. For the feature bound, multiply the finite-sum norm bound in Eq. (A.2) by $\|\mathbf{X}\|_\infty$. These give Eqs. (B.30) and (B.31), respectively.

The bound is worst-case: it ignores cancellations between walk phases and is usually loose when cycles interfere. It nevertheless exposes why a large z should be paired with a larger L, and why normalizing the finite sum or using a zero-initialized residual gate can stabilize deep models.

B.6 Permutation Equivariance

Proposition B.6 (Permutation equivariance)**.** *Let $\mathbf{\Pi}$ be a node permutation matrix. If inputs and edges are relabeled by $\mathbf{\Pi}$, then*

$$\mathbf{T}'_\omega = \mathbf{\Pi}\mathbf{T}_\omega\mathbf{\Pi}^\top, \qquad \mathbf{F}'^{(L)}_{z,\omega}\mathbf{\Pi}\mathbf{X} = \mathbf{\Pi}\mathbf{F}^{(L)}_{z,\omega}\mathbf{X}. \tag{B.32}$$

Proof of Proposition B.6. Lemma B.1 gives $a'_{\pi(u)\pi(v)} = a_{uv}$. Row normalization commutes with the same relabeling, so $\mathbf{P}' = \mathbf{\Pi}\mathbf{P}\mathbf{\Pi}^\top$. Entrywise phase lifting gives $\mathbf{T}'_\omega = \mathbf{\Pi}\mathbf{T}_\omega\mathbf{\Pi}^\top$. Repeated multiplication yields

$$(\mathbf{T}'_\omega)^k = (\mathbf{\Pi}\mathbf{T}_\omega\mathbf{\Pi}^\top)^k = \mathbf{\Pi}\mathbf{T}^k_\omega\mathbf{\Pi}^\top, \tag{B.33}$$

because every adjacent $\mathbf{\Pi}^\top\mathbf{\Pi}$ cancels. Hence

$$\mathbf{F}'^{(L)}_{z,\omega}\mathbf{\Pi}\mathbf{X} = \sum_{k=0}^{L} z^k(\mathbf{T}'_\omega)^k\mathbf{\Pi}\mathbf{X} \tag{B.34}$$

$$= \sum_{k=0}^{L} z^k\mathbf{\Pi}\mathbf{T}^k_\omega\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{X} \tag{B.35}$$

$$= \mathbf{\Pi}\mathbf{F}^{(L)}_{z,\omega}\mathbf{X}. \tag{B.36}$$

□

B.7 Gradient Fields Recover the Sequence Phase

Suppose there is a scalar potential $s : V \to \mathbb{R}$ such that

$$a_{uv} = s_v - s_u. \tag{B.37}$$

Define the diagonal unitary matrix

$$\mathbf{G}_\omega = \operatorname{diag}\left(\mathrm{e}^{\mathrm{i}\omega s_1}, \ldots, \mathrm{e}^{\mathrm{i}\omega s_N}\right). \tag{B.38}$$

Theorem B.7 (*Exact* gradient-field factorization)**.** *For Eq.* (B.37)*,*

$$\mathbf{T}_\omega = \mathbf{G}^*_\omega\mathbf{P}\mathbf{G}_\omega. \tag{B.39}$$

For every polynomial, or power-series function $f(x) = \sum_{k\geq 0} c_k x^k$ whose matrix series converges absolutely in norm (in particular, a power series with radius greater than $\|\mathbf{P}\|_\infty$),

$$f(\mathbf{T}_\omega) = \mathbf{G}^*_\omega f(\mathbf{P})\mathbf{G}_\omega. \tag{B.40}$$

Thus every entry satisfies

$$[f(\mathbf{T}_\omega)]_{uv} = \mathrm{e}^{\mathrm{i}\omega(s_v - s_u)}[f(\mathbf{P})]_{uv}. \tag{B.41}$$

Proof of Theorem B.7. First expand one entry:

$$[\mathbf{G}^*_\omega\mathbf{P}\mathbf{G}_\omega]_{uv} = e^{-i\omega s_u}\mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega s_v} \tag{B.42}$$

$$= \mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega(s_v - s_u)} \tag{B.43}$$

$$= \mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega a_{uv}} = [\mathbf{T}_\omega]_{uv}. \tag{B.44}$$

Thus $\mathbf{T}_\omega = \mathbf{G}_\omega^* \mathbf{P} \mathbf{G}_\omega$. Since $\mathbf{G}_\omega \mathbf{G}_\omega^* = \mathbf{I}$,

$$\mathbf{T}_\omega^2 = (\mathbf{G}_\omega^* \mathbf{P} \mathbf{G}_\omega)(\mathbf{G}_\omega^* \mathbf{P} \mathbf{G}_\omega) \tag{B.45}$$

$$= \mathbf{G}_\omega^* \mathbf{P}^2 \mathbf{G}_\omega, \tag{B.46}$$

and induction gives $\mathbf{T}_\omega^k = \mathbf{G}_\omega^* \mathbf{P}^k \mathbf{G}_\omega$ for every $k \geq 0$. Therefore

$$f(\mathbf{T}_\omega) = \sum_{k \geq 0} c_k \mathbf{T}_\omega^k \tag{B.47}$$

$$= \sum_{k \geq 0} c_k \mathbf{G}_\omega^* \mathbf{P}^k \mathbf{G}_\omega \tag{B.48}$$

$$= \mathbf{G}_\omega^* \left(\sum_{k \geq 0} c_k \mathbf{P}^k \right) \mathbf{G}_\omega \tag{B.49}$$

$$= \mathbf{G}_\omega^* f(\mathbf{P}) \mathbf{G}_\omega. \tag{B.50}$$

Absolute matrix-norm convergence justifies passing the fixed similarity transform through the infinite sum. Taking entry (u, v) produces $[f(\mathbf{T}_\omega)]_{uv} = \mathrm{e}^{\mathrm{i}\omega(s_v - s_u)} [f(\mathbf{P})]_{uv}$. □

Corollary B.8 (Paths and trees)**.** *Every antisymmetric edge field on an undirected tree is a gradient. For the nonnegative-coefficient AW filters with $0 \leq z < 1$, every nonzero entry has phase $\mathrm{e}^{\mathrm{i}\omega(s_j - s_i)}$. In particular, on a path supplied with $a_{i,i+1} = 1$ and hence $s_i = i$, every nonzero coefficient of the truncated or infinite AW filter has phase $\mathrm{e}^{\mathrm{i}\omega(j-i)}$, exactly the relative phase of sequence RoPE.*

Proof of Corollary B.8. Choose a root r and set $s_r = 0$. For each node v, let $r = v_0, \ldots, v_m = v$ be the unique tree path and define

$$s_v = \sum_{j=0}^{m-1} a_{v_j v_{j+1}}. \tag{B.51}$$

For an oriented tree edge $u \to v$ away from the root, $s_v = s_u + a_{uv}$, hence $a_{uv} = s_v - s_u$. Antisymmetry gives the same identity in the reverse orientation. Theorem B.7 therefore applies. For $f(\mathbf{P}) = \sum_{k=0}^{L} z^k \mathbf{P}^k$ or $f(\mathbf{P}) = (\mathbf{I} - z\mathbf{P})^{-1}$, $[f(\mathbf{P})]_{ij}$ is real and nonnegative. Whenever it is nonzero, the only complex phase is $\mathrm{e}^{\mathrm{i}\omega(s_j - s_i)}$. Only for the supplied unit-oriented path, where $s_i = i$ up to an additive constant, does this specialize to $\mathrm{e}^{\mathrm{i}\omega(j-i)}$. □

Remark B.9 (Relative phase and feature mixing). The corollary recovers the *relative rotary character* of every transport coefficient. AW-RoPE is not identical to a node-wise sequence RoPE layer: $f(\mathbf{P})$ also mixes feature magnitudes along walks.

B.8 Gauge Covariance and Cycle Circulation

Proposition B.10 (Gauge covariance)**.** *Let f be a polynomial or an absolutely norm-convergent power series as in Theorem B.7. For arbitrary real node phases ϕ_u, define*

$$a'_{uv} = a_{uv} + \phi_v - \phi_u, \qquad \mathbf{D}_\omega = \mathrm{diag}(\mathrm{e}^{\mathrm{i}\omega\phi_u}). \tag{B.52}$$

Then

$$\mathbf{T}'_\omega = \mathbf{D}_\omega^* \mathbf{T}_\omega \mathbf{D}_\omega, \qquad f(\mathbf{T}'_\omega) = \mathbf{D}_\omega^* f(\mathbf{T}_\omega) \mathbf{D}_\omega. \tag{B.53}$$

Therefore $|[f(\mathbf{T}'_\omega)]_{uv}| = |[f(\mathbf{T}_\omega)]_{uv}|$.

Proof of Proposition B.10. Entrywise,

$$[\mathbf{T}'_\omega]_{uv} = \mathbf{P}_{uv} \mathrm{e}^{\mathrm{i}\omega(a_{uv} + \phi_v - \phi_u)} \tag{B.54}$$

$$= e^{-i\omega\phi_u} [\mathbf{T}_\omega]_{uv} \mathrm{e}^{\mathrm{i}\omega\phi_v} \tag{B.55}$$

$$= [\mathbf{D}_\omega^* \mathbf{T}_\omega \mathbf{D}_\omega]_{uv}. \tag{B.56}$$

Because $\mathbf{D}_\omega \mathbf{D}_\omega^* = \mathbf{I}$, the same cancellation as in Appendix B.7 gives $(\mathbf{T}'_\omega)^k = \mathbf{D}_\omega^* \mathbf{T}_\omega^k \mathbf{D}_\omega$ and hence $f(\mathbf{T}'_\omega) = \mathbf{D}_\omega^* f(\mathbf{T}_\omega) \mathbf{D}_\omega$. Entry (u, v) is multiplied only by $\mathrm{e}^{\mathrm{i}\omega(\phi_v - \phi_u)}$, whose modulus is one. □

Covariant feature transport uses $\mathbf{X}' = \mathbf{D}_\omega^* \mathbf{X}$, so $f(\mathbf{T}'_\omega)\mathbf{X}' = \mathbf{D}_\omega^* f(\mathbf{T}_\omega)\mathbf{X}$. This identity concerns the supplied field transformation; a learned scorer need not respect that transformation of its input features.

For a directed cycle $C = (v_0, v_1, \ldots, v_m = v_0)$, define its circulation

$$\Phi_C = \sum_{j=0}^{m-1} a_{v_j v_{j+1}}. \tag{B.57}$$

Proposition B.11 (Flatness criterion)**.** *On a connected undirected graph, a real antisymmetric field is a gradient if and only if $\Phi_C = 0$ for every cycle C.*

Proof of Proposition B.11. If $a_{uv} = s_v - s_u$, the cycle sum telescopes:

$$\Phi_C = (s_{v_1} - s_{v_0}) + (s_{v_2} - s_{v_1}) + \cdots + (s_{v_0} - s_{v_{m-1}}) \tag{B.58}$$

$$= 0. \tag{B.59}$$

Conversely, fix a root r. For any node v, choose a path $p : r \rightsquigarrow v$ and define $s_v = A(p)$. If p and q are two such paths, following p and then the reverse of q forms a closed walk. It decomposes into cycles with zero circulation, so $A(p) - A(q) = 0$ and s_v is path-independent. Extending a root-to-u path by $u \to v$ gives $s_v = s_u + a_{uv}$, hence $a_{uv} = s_v - s_u$. □

The primitive rotary transport around a cycle is

$$\prod_{(u,v)\in C} \mathrm{e}^{\mathrm{i}\omega a_{uv}} = \mathrm{e}^{\mathrm{i}\omega\Phi_C}. \tag{B.60}$$

This quantity is invariant under Eq. (B.52), because the added node potentials telescope. It is the part of the learned geometry that cannot be removed by changing node phases. At a single frequency, nonzero real circulation is observable in the rotary factor only when $\omega\Phi_C \notin 2\pi\mathbb{Z}$; real flatness and frequency-specific phase flatness must not be conflated.

B.9 Relation to magnetic operators and sheaf transport

The one-step matrix $\mathbf{T}_\theta$ is a learnable analogue of a unitary connection: its entries carry $\mathrm{U}(1)$ phases $\mathrm{e}^{\mathrm{i}\omega a_{uv}}$ on a row-stochastic base, and $\mathbf{F}_\infty = (\mathbf{I} - \mathbf{B})^{-1}$ is the resolvent of the damped connection. A magnetic Laplacian fixes these phases from a potential (edge direction times a charge, or spectral coordinates) and builds a Hermitian operator whose spectrum is then filtered (Zhang et al., 2021). AW-RoPE instead generates each phase from the two endpoint features through a shared bounded scorer, which preserves permutation equivariance, permits the zero, gradient and unrestricted field families of Appendix H.1, and keeps every phase a differentiable function of the current hidden state. Neural sheaf diffusion learns restriction maps between node and edge feature spaces (Bodnar et al., 2022): its transport mixes learned matrix-valued features rather than rotating rotary channel pairs, at a per-edge parameter cost that grows with the feature dimension, whereas AW-RoPE carries one shared scalar displacement per edge with per-channel frequencies. Learned restriction maps can in principle express the same feature-dependent nonlocal transport, so we do not claim uniqueness among learned transports. Neither family pairs the complete operator with a truncated evaluation that carries the forward and derivative bounds of Appendix A.

Two-route interference. Suppose two paths (or two groups whose members each share a phase) from u to v contribute nonnegative magnitudes c_1, c_2 and accumulated angles θ_1, θ_2. Their combined term is

$$A_{uv} = c_1 \mathrm{e}^{\mathrm{i}\theta_1} + c_2 \mathrm{e}^{\mathrm{i}\theta_2}, \qquad |A_{uv}|^2 = c_1^2 + c_2^2 + 2c_1 c_2 \cos(\theta_1 - \theta_2). \tag{B.61}$$

Equal phases maximize the magnitude; a phase gap of π makes the two terms cancel when $c_1 = c_2$. For the two paths, the gap is $\omega\Phi_C$ modulo 2π, where the closed route takes one path forward and the other backward.

The squared magnitude in Eq. (B.61) expands as

$$|A_{uv}|^2 = (c_1 \mathrm{e}^{\mathrm{i}\theta_1} + c_2 \mathrm{e}^{\mathrm{i}\theta_2})(c_1 e^{-i\theta_1} + c_2 e^{-i\theta_2}) \tag{B.62}$$

$$= c_1^2 + c_2^2 + c_1 c_2 \mathrm{e}^{\mathrm{i}(\theta_1-\theta_2)} + c_1 c_2 e^{-i(\theta_1-\theta_2)} \tag{B.63}$$

$$= c_1^2 + c_2^2 + 2c_1 c_2 \cos(\theta_1 - \theta_2), \tag{B.64}$$

where the last step uses $\mathrm{e}^{\mathrm{i}x} + e^{-ix} = 2\cos x$. Traversing the first route and the reverse of the second produces a closed route with circulation $\Phi_C = A(p_1) - A(p_2)$, so $\theta_1 - \theta_2 = \omega\Phi_C$ modulo 2π. This formulation also covers $\omega = 0$ without division by zero.

B.10 Random-Walk and Laplacian Views of AW-RoPE

The random-walk and Laplacian descriptions of AW-RoPE are two exactly equivalent views of the same online operator. Define the ordinary random-walk Laplacian and its phase-lifted counterpart by

$$L_{\mathrm{rw}} = \mathbf{I} - \mathbf{P}, \qquad \mathbf{L}_{\omega,a} = \mathbf{I} - \mathbf{T}_{\omega,a}, \qquad [\mathbf{T}_{\omega,a}]_{uv} = \mathbf{P}_{uv}\mathrm{e}^{\mathrm{i}\omega a_{uv}}. \tag{B.65}$$

Substituting $\mathbf{T}_{\omega,a} = \mathbf{I} - \mathbf{L}_{\omega,a}$ into the truncated analytic filter gives the identity

$$\mathbf{F}^{(L)}_{z,\omega}(\mathbf{T}_{\omega,a})\mathbf{X} = \sum_{k=0}^{L} z^k \mathbf{T}^k_{\omega,a}\mathbf{X} \tag{B.66}$$

$$= \sum_{k=0}^{L} z^k (\mathbf{I} - \mathbf{L}_{\omega,a})^k \mathbf{X}. \tag{B.67}$$

Hence the implementation may be read either as phase-weighted random-walk propagation or as a polynomial filter of a connection random-walk Laplacian. For the zero field, $a_{uv} = 0$, Eq. (B.65) reduces to $\mathbf{L}_{\omega,0} = L_{\mathrm{rw}}$. For a gradient field $a_{uv} = s_v - s_u$, letting $\mathbf{G}_\omega = \mathrm{diag}(\mathrm{e}^{\mathrm{i}\omega s_1}, \ldots, \mathrm{e}^{\mathrm{i}\omega s_N})$ gives, entry by entry,

$$\mathbf{T}_{\omega,a} = \mathbf{G}^*_\omega \mathbf{P} \mathbf{G}_\omega, \qquad \mathbf{L}_{\omega,a} = \mathbf{G}^*_\omega L_{\mathrm{rw}} \mathbf{G}_\omega. \tag{B.68}$$

This recovers the familiar node-factorized case while making clear what changes when the edge field has nonzero circulation.

From Eq. (B.65), $\mathbf{I} - \mathbf{L}_{\omega,a} = \mathbf{T}_{\omega,a}$. Consequently, for each nonnegative integer k,

$$\mathbf{T}^k_{\omega,a} = (\mathbf{I} - \mathbf{L}_{\omega,a})^k, \tag{B.69}$$

and multiplication by z^k, application to $\mathbf{X}$, and summation from zero to L gives Eq. (B.67). In the gradient case, $\mathbf{T}_{\omega,a} = \mathbf{G}^*_\omega \mathbf{P} \mathbf{G}_\omega$, so

$$\mathbf{L}_{\omega,a} = \mathbf{I} - \mathbf{G}^*_\omega \mathbf{P} \mathbf{G}_\omega \tag{B.70}$$

$$= \mathbf{G}^*_\omega \mathbf{I} \mathbf{G}_\omega - \mathbf{G}^*_\omega \mathbf{P} \mathbf{G}_\omega \tag{B.71}$$

$$= \mathbf{G}^*_\omega (\mathbf{I} - \mathbf{P}) \mathbf{G}_\omega \tag{B.72}$$

$$= \mathbf{G}^*_\omega L_{\mathrm{rw}} \mathbf{G}_\omega, \tag{B.73}$$

where the second line uses $\mathbf{G}^*_\omega \mathbf{G}_\omega = \mathbf{I}$. Thus the walk and Laplacian forms are not two approximations; they are the same operator written in transition and Laplacian coordinates.

Laplacian eigenpairs used by LapPE or WIRE can be cached as backbone features. AW instead applies the learned connection operator to current features; its field is recomputed once per layer call, and its L sparse steps are included in online cost.

B.11 Non-Backtracking and Multiscale Variants

Ordinary powers include immediate reversals $u \to v \to u$. We retain the row-action convention $(\mathbf{T}\mathbf{X})_u = \sum_v \mathbf{T}_{uv}\mathbf{X}_v$: the edge index $u \to v$ specifies a walk leaving u, while its feature contribution is gathered from v into row u. Define

$$m^{(1)}_{u\to v} = \mathbf{P}_{uv}\mathbf{R}(\omega a_{uv})\mathbf{X}_v, \tag{B.74}$$

$$\mathbf{S}^{(k)}_u = \sum_{v:(u,v)\in E} m^{(k)}_{u\to v}. \tag{B.75}$$

The non-backtracking update, in exactly this row convention, is

$$m_{u\to v}^{(k+1)} = \mathbf{P}_{uv}\mathbf{R}(\omega a_{uv})\left(\mathbf{S}_v^{(k)} - m_{v\to u}^{(k)}\right). \tag{B.76}$$

The subtracted term removes continuations whose first step immediately returns from v to u; it is zero when no reverse edge exists. The output is $\mathbf{X} + \sum_{k=1}^{L} z^k \mathbf{S}^{(k)}$. This matches the implementation's target gather, source-row sum, and reverse-edge subtraction; it is not an incoming-message action of $\mathbf{T}^\top$. This implementation never forms the $M \times M$ non-backtracking matrix.

A multiscale variant mixes J decay factors $0 \le z_j < 1$ with learned softmax weights $\beta_j \ge 0$, $\sum_{j=1}^{J} \beta_j = 1$. Write $\mathsf{W}_0\mathbf{X} = \mathbf{X}$; $\mathsf{W}_k\mathbf{X} = \mathbf{T}^k\mathbf{X}$ for ordinary walks, and $\mathsf{W}_k\mathbf{X} = \mathbf{S}^{(k)}$ for the non-backtracking recurrence above. Then

$$\mathbf{F}_{\mathrm{MS}}^{(L)}\mathbf{X} = \sum_{j=1}^{J}\beta_j \sum_{k=0}^{L} z_j^k \mathsf{W}_k\mathbf{X} \tag{B.77}$$

$$= \sum_{k=0}^{L}\left(\sum_{j=1}^{J}\beta_j z_j^k\right)\mathsf{W}_k\mathbf{X}. \tag{B.78}$$

The second line shows that all scales share the same propagated states $\mathsf{W}_k\mathbf{X}$; only their scalar coefficients differ. In particular, the evaluated AW-NB-MS uses non-backtracking states, not ordinary powers of $\mathbf{T}$.

B.12 Complexity

Proposition B.12 (*Sparse* cost)**.** *Given the edge field and transition weights, width d and $L \ge 1$ steps require $\mathcal{O}(L(N+M)d)$ arithmetic and $\mathcal{O}((N+M)d)$ forward working memory. When $N = O(M)$, the arithmetic simplifies to $\mathcal{O}(LMd)$. The bound is for the AW transport itself, requires no spectral decomposition, and excludes the edge scorer and subsequent attention kernel. It describes streaming forward evaluation, not peak training memory: ordinary reverse-mode autodiff can retain L steps of intermediate states.*

Proof of Proposition B.12. At one step, Eq. (B.28) reads one endpoint feature, applies a constant-size rotation to each channel pair, multiplies by $\mathbf{P}_{uv}$, and accumulates once for every directed edge. Edge work costs $\mathcal{O}(Md)$; initializing and adding node states costs $\mathcal{O}(Nd)$, including on edgeless graphs. Repeating it $L \ge 1$ times costs $\mathcal{O}(L(N+M)d)$, or $\mathcal{O}(LMd)$ when $N = O(M)$. The forward recurrence stores current and accumulated node states in $\mathcal{O}(Nd)$ memory, together with edge indices, weights, and cached edge rotations in $\mathcal{O}(Md)$ memory. It never stores $\mathbf{T}^k$ or a dense $N \times N$ positional matrix. Non-backtracking adds edge states of the same $\mathcal{O}(Md)$ order. This storage count does not include learned-field construction, attention, or a reverse-mode tape; without recomputation/checkpointing, autodiff may retain intermediate node and edge states from all L steps. □

Full attention still costs $\mathcal{O}(N^2 d)$, whereas Performer supplies its own near-linear attention approximation (Choromanski et al., 2021). AW-RoPE adds the same online sparse transport to either. Query and key tensors are concatenated only to share kernels and then split; this is a constant factor optimization, not a mathematical coupling. We account for these per-layer costs separately from one-time graph preprocessing, following the distinction in Section B.10.

C Notation and Implementation Details

Bold upright symbols denote feature vectors, matrices and linear maps. Scalar variables remain italic, while named operations and text subscripts are upright.

The complete resolvent and finite truncation below are the operators introduced in Section 3.3, with their decay and frequency dependence displayed explicitly. All operator applications to real feature matrices are interpreted independently on their paired channels.

Table C.1: Notation used in the derivations.

Symbol	Meaning
N, M	number of nodes and directed edges
$\mathbf{P}$	row-normalized nonnegative transition
a_{uv}	antisymmetric directed-edge displacement
ω	rotary frequency of one complex channel
$\mathbf{T}_\omega$	$\mathbf{P}$ lifted by $\mathrm{e}^{\mathrm{i}\omega a_{uv}}$
z	geometric walk-decay factor in $[0, 1)$
L	largest evaluated walk length
$\mathbf{F}_{z,\omega}$	infinite analytic resolvent
$\mathbf{F}^{(L)}_{z,\omega}$	degree-L truncation
$A(p), \pi(p)$	displacement and transition weight of walk p
Φ_C	signed circulation around cycle C

Algorithm 1 *Sparse* AW-RoPE forward recurrence.

Require: edge list E, row weights $\mathbf{P}$, displacement a, input $\mathbf{X} = [\mathbf{Q}; \mathbf{K}]$, frequencies ω, decay z, depth L
1: $\mathbf{S} \leftarrow \mathbf{X}$; $\mathbf{Y} \leftarrow \mathbf{X}$; $c \leftarrow 1$
2: **for** $k = 1, \ldots, L$ **do**
3: rotate each gathered endpoint pair: $m_{uv} \leftarrow \mathbf{P}_{uv}\mathbf{R}(\omega a_{uv})\mathbf{S}_v$
4: sum into row u: $\mathbf{S}_u \leftarrow \sum_{v:(u,v)\in E} m_{uv}$
5: $c \leftarrow cz$
6: $\mathbf{Y} \leftarrow \mathbf{Y} + c\mathbf{S}$
7: **end for**
8: split $\mathbf{Y}$ back into $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{K}}$

C.1 Sparse Forward Algorithm

The same graph edge list is used at every step. Query and key states are concatenated along the feature axis so the propagation kernel is launched once, but they are not mixed.

C.2 Optional Finite-Sum Normalization

When desired, the implementation divides the finite filter by its scalar mass

$$C_L(z) = \sum_{k=0}^{L} z^k = \frac{1 - z^{L+1}}{1 - z}. \tag{C.1}$$

The normalized operator is

$$\widehat{\mathbf{F}}^{(L)}_{z,\omega} = \frac{1}{C_L(z)}\mathbf{F}^{(L)}_{z,\omega}. \tag{C.2}$$

By Eq. (B.31), its induced infinity norm is at most one. This normalization changes the amplitude but not the phase in the gradient case. The Cluster ablation shows that the bound can improve stability without necessarily improving task accuracy; the main attention model therefore uses the unnormalized form unless stated otherwise.

C.3 Directed-Edge View of Non-Backtracking AW

Index states by directed edges. The phase-lifted non-backtracking matrix is

$$\mathbf{B}_{(u\to v),(x\to y)} = \mathbf{1}[x = v]\mathbf{1}[y \neq u]\mathbf{P}_{xy}\mathrm{e}^{\mathrm{i}\omega a_{xy}}. \tag{C.3}$$

The first indicator joins consecutive edges; the second forbids immediate reversal. This matrix acts on column states by gathering the next edge into the current edge row. To relate it explicitly to the node recurrence, set $e^{(0)}_{u\to v} = \mathbf{X}_v$ and $m^{(k)}_{u\to v} = \mathbf{P}_{uv}\mathbf{R}(\omega a_{uv})e^{(k-1)}_{u\to v}$. Then

$$e^{(k)}_{u\to v} = \mathbf{S}^{(k)}_v - m^{(k)}_{v\to u} \tag{C.4}$$

$$= \sum_{y\neq u} \mathbf{P}_{vy}\mathbf{R}(\omega a_{vy})e^{(k-1)}_{v\to y} = (\mathbf{B}e^{(k-1)})_{u\to v}, \tag{C.5}$$

where complex phases and real pair rotations are equivalent representations. Thus Eq. (B.76) obtains the same result without forming $\mathbf{B}$, using a target gather, source-row sum, and reverse-edge subtraction. The node output is $\mathbf{X} + \sum_{k=1}^{L} z^k \mathbf{S}^{(k)}$.

C.4 Integration with Attention and GIN

For each complex channel ℓ, the attention version computes

$$\widetilde{\mathbf{q}}^{(\ell)} = \mathbf{F}^{(L)}_{z,\omega_\ell} \mathbf{q}^{(\ell)}, \qquad \widetilde{\mathbf{k}}^{(\ell)} = \mathbf{F}^{(L)}_{z,\omega_\ell} \mathbf{k}^{(\ell)}. \tag{C.6}$$

The transformed tensors enter the usual scaled dot-product kernel or the chosen Performer feature map. Dataset encoder, value path, attention kernel, optimizer, and prediction head can therefore remain identical to the WIRE baseline.

To test the analytic operator outside attention, we also insert it as a residual GIN branch (Xu et al., 2019):

$$\mathbf{H}_{\text{in}} = \mathbf{H} + \gamma \left(\mathbf{F}^{(L)}_{z,\omega} \mathbf{H} - \mathbf{H} \right). \tag{C.7}$$

The GIN block then returns $\mathbf{H}' = \mathrm{GINBlock}(\mathbf{H}_{\text{in}})$, as in Figure 2. The legacy branch uses a sigmoid gate initialized to $\sigma(-2)$. The ReZero variant (Bachlechner et al., 2021) uses an unconstrained γ initialized exactly to zero, so the initial network is functionally identical to its paired NoPE GIN. This GIN study tests the broader transport operator and complements the Performer/WIRE positional comparison. Unless otherwise stated, the attention implementation uses the unnormalized finite sum, $L = 8$, $z = 0.8$, a learnable z, and injection at every attention layer.

C.5 Linear-attention contraction and batching

The AW action is evaluated before the attention feature map. Here node query, key and value vectors are columns; $\mathbf{o}_u$ denotes the attention output at node u, whose row is written with an explicit transpose. For one graph, write the mapped query/key features as φ_Q and φ_K, allowing for the backend's role-specific FAVOR+ stabilization. The usual factorized contraction is

$$\mathbf{o}_u^\top = \frac{\varphi_Q(\widetilde{\mathbf{q}}_u)^\top \sum_v \varphi_K(\widetilde{\mathbf{k}}_v) \mathbf{v}_v^\top}{\varphi_Q(\widetilde{\mathbf{q}}_u)^\top \sum_v \varphi_K(\widetilde{\mathbf{k}}_v)}. \tag{C.8}$$

The noncausal backend uses positive feature-map offsets for numerical stability (ReLU: 10^{-3}; FAVOR+: 10^{-4}), rather than adding a separate constant to this denominator.

In the evaluated Graph-RoPE backend, mini-batches are padded to the largest graph. Padded keys and values are zeroed before the feature map. Since $\varphi_K(0)$ need not vanish, padded positions can still contribute to the denominator while their zero values contribute nothing to the numerator. Thus the batched implementation uses the same contraction over all padded token slots, not a strict masked sum over real nodes alone. This inherited batching convention is shared by the local Graph-RoPE positional variants; AW transport itself operates only on the original graph nodes.

C.6 Fixed-field controls

The archived Static and Flat controls use the following fixed-field protocol. On the simple undirected support, let $\mathbf{P}$ be its row-normalized adjacency and form $\mathbf{r}_u = [\log(1 + d_u), (\mathbf{P}^2)_{uu}, \ldots, (\mathbf{P}^8)_{uu}]$. Each channel is standardized within the graph using population SD; channels with SD at most 10^{-8} become zero. With $\mathbf{J} = \mathrm{diag}(\mathbf{J}_2, \ldots, \mathbf{J}_2)$, $\mathbf{J}_2 = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$, the fixed displacement is

$$b_{uv} = \mathbf{r}_u^\top \mathbf{J} \mathbf{r}_v, \qquad a_{uv}^{\text{static}} = \pi \tanh\left(\frac{b_{uv}}{s + 10^{-8}} \right).$$

Here s is the median of $|b_{uv}|$ over one orientation per edge, falling back to RMS and then 1 when degenerate. Reversed edges negate the field. The field may be zero on symmetric graphs; nontrivial circulation is not guaranteed. For Flat, let $\mathbf{C}$ be the oriented incidence matrix and solve $\psi = \arg\min_\psi \|\mathbf{C}\psi - a^{\text{static}}\|_2^2$. Use $a^{\text{flat}} = \mathbf{C}\psi$ without additional clipping. This unweighted least-squares projection has zero real circulation around every cycle. Both controls use sparse AW transport; neither is the newly trained *Exact* arm.

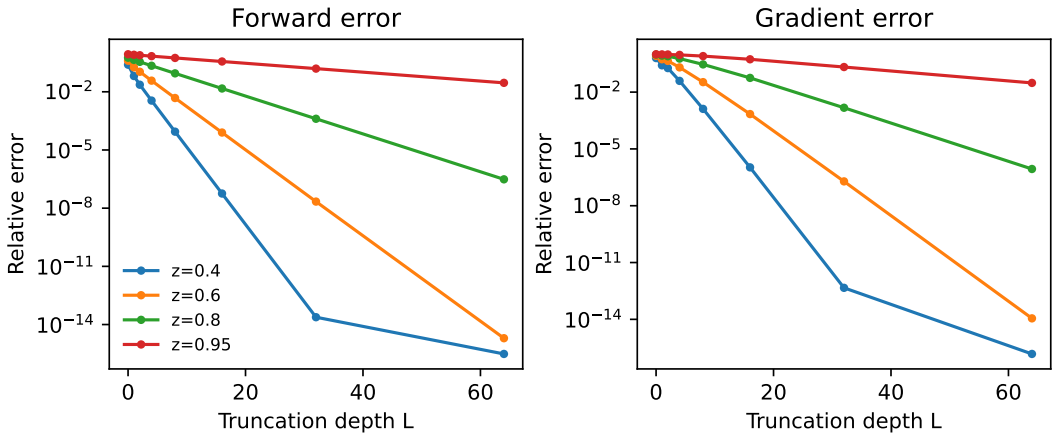

Figure D.1: Deterministic *Exact*-AW versus sparse-AW diagnostics on a fixed graph, input, and linear readout, in double precision. Left: maximum complex-feature error relative to the complete output. Right: relative Euclidean error of the readout gradient jointly with respect to inputs, edge displacements, frequencies, edge weights, and decay. Each curve changes only L at the indicated z; 32 configurations including $L = 0$ are audited. These are operator/derivative checks, not trained-model comparisons. Near-unit decay requires greater depth; numerical floors need not decrease monotonically.

D Numerical Checks of Operator Identities

The deterministic audit fixes the random generator state. A 12-node connected weighted graph is formed from a ring plus random symmetric chords. Its transition is row-normalized; the edge field is obtained by antisymmetrizing a random matrix. All calculations use complex128.

Table D.1: Maximum absolute discrepancies in the independent numerical identity audit.

Identity	Maximum error
Gradient operator factorization	1.11×10^{-16}
Gradient resolvent factorization	4.44×10^{-16}
Gauge covariance, operator	1.31×10^{-16}
Gauge covariance, resolvent	4.48×10^{-16}
Gauge-invariant magnitude	4.44×10^{-16}
Permutation equivariance	1.49×10^{-15}
Two-route expansion	2.22×10^{-16}
Path resolvent factorization	5.98×10^{-16}

These tests validate the implementation of the stated identities, not the empirical usefulness of the assumptions.

D.1 Forward and Derivative Truncation

The independent bound check at $z = 0.65$ also satisfies Eq. (B.30); its full depth sweep is retained with the artifact. Memory and timing measurements are consolidated in Appendix F.

E Matched Synthetic Study and Trained-Checkpoint Analysis

Matched predictive and efficiency protocol. The matched protocol freezes the existing five synthetic tasks' AW learning rates, initial decay, and truncation depths. Both arms learn the field, frequencies, and decay with identical full-model initialization and 250-epoch budgets over three matched runs: 15 *Exact*-AW and 15 sparse-AW trials. Checkpoints are selected on validation and evaluated at the same test epoch. Ten one-epoch probes precede training. Separate timing repeats use one trial per GPU, one warm-up and three measured training epochs, for both arms and all five

Table E.1: Every run in the completed *Exact–Sparse* study. Test nRMSE is paired with the best validation epoch (zero-based); lower is better. No run is removed. Three-decimal values are displayed here; the artifact retains full precision.

		Exact AW-RoPE		*Sparse* AW-RoPE	
Dataset	Seed	Test nRMSE	Epoch	Test nRMSE	Epoch
Monochromatic-0	0	0.040	224	0.040	224
Monochromatic-0	1	0.040	219	0.039	219
Monochromatic-0	2	0.051	212	0.055	215
Monochromatic-5	0	0.035	195	0.035	230
Monochromatic-5	1	0.034	214	0.035	208
Monochromatic-5	2	0.039	200	0.039	185
Monochromatic-10	0	0.027	219	0.027	228
Monochromatic-10	1	0.027	235	0.027	228
Monochromatic-10	2	0.027	226	0.043	200
Monochromatic-15	0	0.023	217	0.024	221
Monochromatic-15	1	0.030	224	0.031	210
Monochromatic-15	2	0.026	195	0.029	221
Watts–Strogatz SPD	0	0.016	104	0.022	74
Watts–Strogatz SPD	1	0.017	99	0.014	155
Watts–Strogatz SPD	2	0.014	187	0.015	106

Table E.2: Isolated full-model H200 cost for the matched synthetic study. Entries are three-decimal mean and sample SD over three timing repeats. L is the *Sparse* depth; *Exact* is untruncated. Smaller is better.

		Training epoch (s)		Peak allocated (MiB)	
Dataset	L	*Exact*	*Sparse*	*Exact*	*Sparse*
Monochromatic-0	8	$16.659_{\pm 0.253}$	$22.974_{\pm 0.310}$	$157.676_{\pm 0.000}$	$135.950_{\pm 0.000}$
Monochromatic-5	16	$16.572_{\pm 0.075}$	$34.561_{\pm 0.140}$	$156.801_{\pm 0.000}$	$161.184_{\pm 0.000}$
Monochromatic-10	16	$16.615_{\pm 0.190}$	$34.641_{\pm 0.920}$	$155.925_{\pm 0.000}$	$153.277_{\pm 0.000}$
Monochromatic-15	8	$17.273_{\pm 1.044}$	$22.980_{\pm 0.082}$	$155.050_{\pm 0.000}$	$123.479_{\pm 0.000}$
Watts–Strogatz SPD	8	$16.932_{\pm 0.229}$	$23.355_{\pm 0.154}$	$107.952_{\pm 0.000}$	$107.347_{\pm 0.000}$

tasks with three repeats. These measurements must be distinguished from packed concurrent training throughput. The separate size sweep is in Appendix F.

Completed paired results and provenance. The 30 formal runs and 30 isolated timing repeats passed the complete artifact audit. All formal validation/test trajectories contain epochs 0–249; selected and final checkpoints are retained. Each of the 15 dataset–run pairs has identical initialization, parameter count, split sizes, and layer configuration. The implementation solves a $2N$ real-block system with a cuSOLVER library preference; small batched kernels may use cuBLAS. The publication artifact contains the frozen configuration, per-result and trajectory hashes, all full-precision metrics, and a regenerating script. Normalized RMSE divides the logged RMSE by 25 for the monochromatic tasks and by 10 for Watts–Strogatz SPD, consistently with the earlier tables.

Measured cost and its scope. Table E.2 averages three independent timing repeats per arm and task; each repeat averages three measured training epochs after one warm-up. One trial occupies each allocated H200 during timing. Epoch time includes training and logging, but excludes validation, test evaluation, checkpoints, and cold preprocessing. Peak allocated GPU memory is the maximum over the measured epochs, summarized across repeats. Both arms contain the same 39,575 trainable parameters and dense-attention backbone. These are current-implementation small-graph costs, not the packed formal-run elapsed times. *Exact* is faster on all five tasks, and *Sparse* is not uniformly more memory-efficient. This does not contradict the sparse asymptotic bound: batched dense solves can be efficient at small sizes, while a finite recurrence incurs repeated kernel work. Appendix F supplies the separate size benchmark and measured allocation failures. Those layer-level measurements must not be conflated with the full-model epoch measurements here.

Table E.3: Relative output error at trained checkpoints. Means and sample SDs across all three run-level context averages, with three-decimal mantissas. These are approximation errors, not nRMSE or errors between independently trained models.

Task	Checkpoint origin	$L = 8$	$L = 16$	$L = 32$	$L = 64$
M0	Exact-trained	$(7.244_{\pm 3.230}) \times 10^{-3}$	$(1.971_{\pm 1.284}) \times 10^{-4}$	$(2.545_{\pm 2.514}) \times 10^{-7}$	$(8.153_{\pm 12.055}) \times 10^{-13}$
M0	Sparse-trained	$(6.738_{\pm 3.501}) \times 10^{-3}$	$(2.095_{\pm 1.904}) \times 10^{-4}$	$(4.470_{\pm 6.379}) \times 10^{-7}$	$(3.682_{\pm 6.271}) \times 10^{-12}$
M5	Exact-trained	$(6.530_{\pm 4.548}) \times 10^{-2}$	$(8.044_{\pm 6.941}) \times 10^{-3}$	$(1.399_{\pm 1.465}) \times 10^{-4}$	$(5.112_{\pm 6.865}) \times 10^{-8}$
M5	Sparse-trained	$(5.961_{\pm 4.192}) \times 10^{-2}$	$(7.044_{\pm 6.310}) \times 10^{-3}$	$(1.247_{\pm 1.407}) \times 10^{-4}$	$(6.065_{\pm 8.981}) \times 10^{-8}$
M10	Exact-trained	$(1.014_{\pm 0.126}) \times 10^{-1}$	$(1.539_{\pm 0.193}) \times 10^{-2}$	$(4.049_{\pm 0.313}) \times 10^{-4}$	$(3.794_{\pm 1.652}) \times 10^{-7}$
M10	Sparse-trained	$(1.075_{\pm 0.081}) \times 10^{-1}$	$(1.743_{\pm 0.246}) \times 10^{-2}$	$(5.857_{\pm 2.700}) \times 10^{-4}$	$(1.281_{\pm 1.553}) \times 10^{-6}$
M15	Exact-trained	$(1.214_{\pm 0.209}) \times 10^{-1}$	$(2.234_{\pm 0.911}) \times 10^{-2}$	$(1.147_{\pm 1.098}) \times 10^{-3}$	$(7.851_{\pm 12.238}) \times 10^{-6}$
M15	Sparse-trained	$(1.124_{\pm 0.173}) \times 10^{-1}$	$(1.970_{\pm 0.418}) \times 10^{-2}$	$(8.428_{\pm 2.877}) \times 10^{-4}$	$(2.789_{\pm 1.681}) \times 10^{-6}$
WS	Exact-trained	$(1.502_{\pm 0.146}) \times 10^{-1}$	$(2.938_{\pm 0.636}) \times 10^{-2}$	$(1.232_{\pm 0.614}) \times 10^{-3}$	$(2.838_{\pm 2.956}) \times 10^{-6}$
WS	Sparse-trained	$(1.460_{\pm 0.195}) \times 10^{-1}$	$(2.913_{\pm 0.731}) \times 10^{-2}$	$(1.356_{\pm 0.621}) \times 10^{-3}$	$(3.952_{\pm 3.052}) \times 10^{-6}$

E.1 Approximation at trained checkpoints

We check the operators learned in the matched synthetic study. We reuse all 30 validation-best checkpoints: five tasks, both training arms, over three matched runs. Checkpoint hashes, original commands, validation selection, and shared initialization are audited before the diagnostic.

For each checkpoint, its original *Exact* or *Sparse* network runs in evaluation mode on the first four validation examples, in two fixed batches of two. We capture H and projected $X = [Q; K]$ at all four insertion layers. A sweep then holds these local inputs, graph, scorer parameters, frequencies and decay fixed while comparing *Exact* with $L \in \{0, 1, 2, 4, 8, 16, 32, 64\}$. Each layer is checked separately: changing L does not change the inputs supplied to later-layer checks. The captured float32 weights and activations are promoted to float64 for this numerical diagnostic.

Output error is the relative Euclidean norm of $\mathcal{F}_\theta^{(L)} X - \mathcal{F}_\theta X$. For derivatives, three deterministic, unit-norm Gaussian probes R_j define the same local linear readouts $\langle R_j, Y \rangle$ for both evaluations. We compare their vector–Jacobian products (VJPs) with respect to independent local X, H, the field scorer's parameters, frequencies, and decay logit. The audit records the joint relative Euclidean error together with per-group absolute/relative errors. A zero reference gradient gives an undefined relative error, recorded as null rather than replaced with zero. These are sampled local derivatives, not the full Jacobian norm or a downstream task-loss gradient.

The complete depth curves appear in Figure 5.

Across the ten task–checkpoint-origin groups, at $L = 8$ the mean relative output errors range from 0.674% to 15.023%, and mean VJP errors from 1.345% to 25.174%. At $L = 64$, the largest group-mean output and VJP errors are 7.851×10^{-6} and 3.504×10^{-5}, respectively. Learned layer decays span 0.468–0.867. These observations support useful trained-operator approximation at sufficient depth on the sampled inputs, while showing that shallow truncation need not already be close to *Exact*. Group means are not worst-case error guarantees.

The audit contains 240 fixed layer/batch contexts, 1,920 output comparisons, and 5,760 probe-VJP comparisons. All checkpoint origins and depths are retained. *Exact* linear-system residuals and the forward tail bound are checked, and checkpoint/module/input hashes verify that the sweep does not change parameters or inputs. The diagnostic uses four validation examples per task and checkpoint. Learned decay is not reset to its initial value.

F Exact–Sparse Size Scaling and Timing Audit

The size experiment measures the production positional-attention layer, not an entire dataset training run. All methods use float32 on an isolated H200, hidden width 64 and four heads. *Exact* and *Sparse* share learned antisymmetric fields, learnable frequencies and decay initialized at $z = 0.8$, identical inputs, and identical layer initial states within each repeat. The current *Exact* implementation solves the same real-block linear system used by the completed synthetic study. Its implementation and cuSOLVER library preference are fixed across sizes; PyTorch's internal kernel dispatch can change with matrix dimensions.

Table F.1: **Matched learnable positional-layer cost on H200.** ReLU linear attention, batch size eight, width 64, four heads, $z = 0.8$. Forward/backward times and peak allocated memory are summarized over three independent repeats, each averaging ten warmed steps. *Exact* is untruncated; *Sparse* uses $L = 16$. All values have three decimals. All 72 small-graph cells (including the companion FAVOR+ table) are complete. The 256-node *Exact* timing jump is discussed below.

Nodes/graph	Method	L	Forward ms	Backward ms	Total ms	Peak MiB
64	NoPE	–	$1.430_{\pm 0.054}$	$1.276_{\pm 0.049}$	$2.706_{\pm 0.101}$	$99.006_{\pm 0.000}$
64	WIRE	–	$1.621_{\pm 0.018}$	$1.577_{\pm 0.024}$	$3.198_{\pm 0.038}$	$99.195_{\pm 0.000}$
64	Exact AW-RoPE	–	$3.133_{\pm 0.068}$	$2.460_{\pm 0.065}$	$5.593_{\pm 0.127}$	$218.241_{\pm 0.000}$
64	Sparse AW-RoPE	16	$3.495_{\pm 0.094}$	$6.096_{\pm 0.214}$	$9.591_{\pm 0.306}$	$180.102_{\pm 0.000}$
128	NoPE	–	$1.469_{\pm 0.049}$	$1.311_{\pm 0.033}$	$2.780_{\pm 0.082}$	$101.854_{\pm 0.000}$
128	WIRE	–	$1.591_{\pm 0.056}$	$1.566_{\pm 0.031}$	$3.157_{\pm 0.073}$	$102.231_{\pm 0.000}$
128	Exact AW-RoPE	–	$5.484_{\pm 0.040}$	$2.823_{\pm 0.031}$	$8.307_{\pm 0.069}$	$660.393_{\pm 0.000}$
128	Sparse AW-RoPE	16	$3.721_{\pm 0.195}$	$6.401_{\pm 0.413}$	$10.122_{\pm 0.485}$	$265.532_{\pm 0.000}$
256	NoPE	–	$1.409_{\pm 0.043}$	$1.279_{\pm 0.032}$	$2.688_{\pm 0.075}$	$107.551_{\pm 0.000}$
256	WIRE	–	$1.643_{\pm 0.037}$	$1.675_{\pm 0.076}$	$3.318_{\pm 0.100}$	$108.302_{\pm 0.000}$
256	Exact AW-RoPE	–	$433.971_{\pm 0.386}$	$5.592_{\pm 0.064}$	$439.563_{\pm 0.404}$	$2409.198_{\pm 0.000}$
256	Sparse AW-RoPE	16	$3.849_{\pm 0.552}$	$6.400_{\pm 0.230}$	$10.249_{\pm 0.780}$	$430.892_{\pm 0.000}$

For the small-graph positional-layer tables, batch size is eight and graph sizes are 64, 128 and 256. NoPE, WIRE, *Exact* and *Sparse* ($L = 16$) are measured under both ReLU and softmax FAVOR+ attention. WIRE spectral coordinates are prepared before timing. The size sweep instead uses batch size one, ReLU attention, nodes in $\{64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384\}$, degrees 6 and 18, and *Sparse* depths $L \in \{8, 16, 32\}$. The graphs are bidirectional rings with fixed-offset chords, so degree and node count are controlled separately. These are synthetic cost probes, not a real-data accuracy benchmark.

Each cell has a fresh process and one GPU. We reset the RNG before model construction and record hashes of parameters, graph and inputs. Within a repeat we run three warm-up steps and ten measured steps at fixed parameters. Each measured step computes a complete layer forward, a scalar squared-output loss, and backward; it does not update an optimizer. CUDA synchronization surrounds the forward/backward regions, and peak allocated memory is reset before the forward. Finite outputs and parameter/input gradients are checked outside the timed region. The mean and sample SD summarize the three independent repeats. Packed training throughput is not used as a timing observation.

The declared grid contains 288 cells: 72 for the small-graph layer table and 216 for the size sweep. A CUDA allocation failure is recorded as OOM; a predeclared 900-second cell limit is recorded separately as timeout. Neither is converted into an arbitrary runtime or an accuracy score. All cells and statuses are retained. *Sparse* transport error relative to *Exact* at the same parameters is also measured outside the timed region for sizes up to 1024; this is not predictive error.

The 256-node timing jump. *Exact* forward rises from 5.484 to 433.971 ms between 128 and 256 nodes, while the 256-node backward takes 5.592 ms. FAVOR+ shows the same jump (435.142 ms). The recorded PyTorch build (`2.7.0a0+ecf3bae40a.nv25.02`, CUDA 12.8) solves real blocks of twice the node count. At matrix dimension 512, its LU dispatch switches from batched cuBLAS to looped cuSOLVER; ordinary first-order backward reuses the LU factors.[1] This source-level switch is consistent with the observed jump; the timings do not isolate its runtime contribution. Peak allocation grows by a factor of 3.65, close to the quadratic growth of dense matrices. The observed crossing therefore reflects this software stack and batching configuration.

Measured capacity and approximation. All 288 cells terminate: 276 successful measurements and 12 actual CUDA OOMs, with no timeouts or software failures. The OOMs are precisely *Exact* at 8,192 and 16,384 nodes, both degrees, all three repeats. *Sparse* completes every size and depth. At degree six and $L = 16$, 16,384-node forward/backward takes 27.287 ms with 2.731 GiB peak allocated memory. *Exact* has lower measured latency at 64 nodes; this configuration-specific observation is separate from the isolated full-model training times in Table E.2.

[1] Version-matched PyTorch LU dispatch and solve backward.

Table F.2: Companion softmax FAVOR+ positional-layer timings. Protocol, hardware, initialization and reporting match the ReLU measurements in Table F.1. Entries are mean$_{\pm SD}$, three decimals; *Exact* is untruncated.

Nodes	Method	L	Forward ms	Backward ms	Total ms	Peak MiB
64	NoPE	–	$1.532_{\pm 0.035}$	$1.511_{\pm 0.037}$	$3.043_{\pm 0.068}$	$99.702_{\pm 0.000}$
64	WIRE	–	$2.009_{\pm 0.513}$	$1.753_{\pm 0.023}$	$3.762_{\pm 0.492}$	$99.891_{\pm 0.000}$
64	Exact AW-RoPE	–	$3.717_{\pm 0.825}$	$2.703_{\pm 0.018}$	$6.420_{\pm 0.834}$	$218.241_{\pm 0.000}$
64	Sparse AW-RoPE	16	$3.799_{\pm 0.213}$	$6.618_{\pm 0.467}$	$10.417_{\pm 0.678}$	$180.797_{\pm 0.000}$
128	NoPE	–	$1.565_{\pm 0.023}$	$1.543_{\pm 0.004}$	$3.108_{\pm 0.019}$	$103.246_{\pm 0.000}$
128	WIRE	–	$1.708_{\pm 0.008}$	$1.774_{\pm 0.020}$	$3.482_{\pm 0.015}$	$103.622_{\pm 0.000}$
128	Exact AW-RoPE	–	$5.598_{\pm 0.039}$	$3.044_{\pm 0.009}$	$8.642_{\pm 0.032}$	$660.393_{\pm 0.000}$
128	Sparse AW-RoPE	16	$3.694_{\pm 0.127}$	$6.524_{\pm 0.237}$	$10.219_{\pm 0.364}$	$266.923_{\pm 0.000}$
256	NoPE	–	$1.635_{\pm 0.127}$	$1.534_{\pm 0.040}$	$3.168_{\pm 0.166}$	$110.333_{\pm 0.000}$
256	WIRE	–	$1.725_{\pm 0.024}$	$1.871_{\pm 0.046}$	$3.596_{\pm 0.068}$	$111.084_{\pm 0.000}$
256	Exact AW-RoPE	–	$435.142_{\pm 0.440}$	$5.969_{\pm 0.103}$	$441.111_{\pm 0.532}$	$2409.198_{\pm 0.000}$
256	Sparse AW-RoPE	16	$3.854_{\pm 0.042}$	$6.892_{\pm 0.206}$	$10.746_{\pm 0.216}$	$434.924_{\pm 0.000}$

At 1,024 nodes and degree six, the same-input relative feature ℓ_2 errors for $L = 8, 16, 32$ average 0.079086, 0.011728, and 0.000286, respectively. These are operator checks at the initialized parameters, not errors after independently training each version. They complement the worst-case bound with the measured depth–accuracy trade-off.

G Search Spaces and Selected Hyperparameters

Table G.1 lists the shared hyperparameters of the matched synthetic study summarized in Section 4.2, following the Graph-RoPE protocol (Reid et al., 2026); *Exact* and *Sparse* use these settings with matched initializations and identical trainable-parameter counts.

Table G.1: Shared hyperparameters of the matched synthetic study (Section 4.2). Both *Exact* and *Sparse* arms use these settings with matched initializations.

Hyperparameter	Value
Backbone layers	4
Hidden width	32
Attention heads	1
Dropout	0.2
Batch size	16
Attention	full
Pooling	mean
Auxiliary PE	LapPE
Optimizer	Adam
Epochs	250
Weight decay	10^{-4}
LR schedule	cosine decay
Trainable parameters	39,575
Sparse depth L	8 or 16

The completed 110-trial GIN synthetic search uses learning rates $\{10^{-4}, 2 \times 10^{-4}\}$, $L \in \{8, 16\}$ for the family grid, and $z \in \{0.6, 0.8\}$ where applicable. NoPE has no effective L or z; the values in its configuration files are ignored. Table G.2 records the validation winner for every dataset and transport family so the ablation is reproducible.

G.1 GIN Optimization Ablation

This ablation isolates how an AW branch enters a deep network. On Cluster, all arms use four GIN layers, hidden width 32, batch size 16, $L = 16$, $z = 0.6$, learning rate 2×10^{-4}, 60 epochs, and four paired matched runs. Checkpoints are selected only by validation accuracy. Table G.4 reports mean $\pm$ sample SD of test accuracy at that epoch.

Table G.2: Validation-selected configurations in the completed GIN synthetic grid. Metric is normalized RMSE (lower is better).

Dataset	Method	LR	L	z	Validation
Mono-0	NoPE	1e−4	–	–	0.046553
Mono-0	AW	2e−4	16	0.6	0.031604
Mono-0	AW-NB	2e−4	16	0.8	0.029987
Mono-0	AW-NB-MS	2e−4	8	0.8	0.032441
Mono-5	NoPE	2e−4	–	–	0.044055
Mono-5	AW	2e−4	16	0.8	0.026159
Mono-5	AW-NB	2e−4	16	0.8	0.031583
Mono-5	AW-NB-MS	2e−4	16	0.8	0.040315
Mono-10	NoPE	2e−4	–	–	0.036058
Mono-10	AW	2e−4	8	0.8	0.020206
Mono-10	AW-NB	2e−4	8	0.6	0.022442
Mono-10	AW-NB-MS	2e−4	16	0.8	0.023225
Mono-15	NoPE	2e−4	–	–	0.026099
Mono-15	AW	2e−4	16	0.8	0.014582
Mono-15	AW-NB	2e−4	8	0.8	0.015003
Mono-15	AW-NB-MS	2e−4	16	0.8	0.016190
Watts SPD	NoPE	2e−4	–	–	0.016607
Watts SPD	AW	2e−4	8	0.8	0.006559
Watts SPD	AW-NB	2e−4	8	0.8	0.010137
Watts SPD	AW-NB-MS	1e−4	16	0.8	0.016255

Table G.3: **GIN synthetic confirmation.** Test nRMSE ↓, mean$_{\pm \mathrm{SD}}$ over two matched runs after 250 epochs, with configurations frozen from the validation-only grid. Bold marks the lowest unrounded mean.

Dataset	GIN-NoPE	GIN-AW-RoPE	GIN-AW-NB	GIN-AW-NB-MS	Winner
Monochromatic-0	$0.047_{\pm 0.000}$	$0.035_{\pm 0.001}$	$\mathbf{0.034_{\pm 0.002}}$	$0.036_{\pm 0.002}$	AW-NB
Monochromatic-5	$0.041_{\pm 0.000}$	$\mathbf{0.026_{\pm 0.001}}$	$0.031_{\pm 0.001}$	$0.032_{\pm 0.005}$	AW
Monochromatic-10	$0.034_{\pm 0.000}$	$\mathbf{0.020_{\pm 0.000}}$	$0.022_{\pm 0.001}$	$0.022_{\pm 0.000}$	AW
Monochromatic-15	$0.024_{\pm 0.000}$	$\mathbf{0.016_{\pm 0.000}}$	$0.016_{\pm 0.000}$	$0.016_{\pm 0.000}$	AW
Watts–Strogatz SPD	$0.013_{\pm 0.001}$	$\mathbf{0.006_{\pm 0.001}}$	$0.007_{\pm 0.002}$	$0.011_{\pm 0.001}$	AW

All 12 completed AW trials in this ablation beat their matched NoPE arm. The zero-initialized, per-layer ReZero branch is best; finite-sum normalization reduces its mean by 1.97 points, and a single pre-backbone injection loses a further 8.81 points. The learned ReZero gates move away from zero, while the within-graph node cosine no longer collapses near one. This supports a specific interpretation: exact identity initialization fixes an optimization problem, whereas the analytic transport supplies the eventual gain.

Table G.4: **AW-RoPE Cluster stability ablation, four matched runs.** All arms are paired and budget-matched. This table isolates *Sparse* AW-RoPE injection. Δ is relative to NoPE test accuracy.

Injection	Validation ↑	Test ↑	Δ
NoPE	0.4487±0.0286	0.4502±0.0276	–
AW-RoPE ReZero	**0.6457±0.0022**	**0.6461±0.0024**	+0.1959
AW-RoPE ReZero + norm.	0.6265±0.0057	0.6263±0.0057	+0.1762
AW-RoPE Single + norm.	0.5402±0.0128	0.5382±0.0138	+0.0880

H Completed Mechanism, Sensitivity, and Efficiency Study

The revision study is complete: 27 mechanism runs, 24 additional sensitivity runs, and 45 retained isolated timing runs. The sensitivity grid reuses the three unrestricted $(L, z) = (8, 0.8)$ first-run mechanism runs, giving 27 cells from 51 unique accuracy runs. The sensitivity grid uses the first run of each arm; additional repeats were deferred for budget. All retained results, source hashes and plot/table generation are archived with the artifact.

H.1 Frozen field controls and circulation

Monochromatic-0 uses the synthetic dense-attention backbone, 250 epochs, batch size 16, and learning rate 10^{-4}. Peptides-struct uses 200 epochs, batch size 128, and learning rate 3×10^{-4}; MolHIV uses 100 epochs, batch size 32, and learning rate 10^{-4}. Both real tasks use the frozen softmax FAVOR+ backbone. Every field arm uses three matched runs, $L = 8$, fixed $z = 0.8$, learned rotary frequencies, and all insertion layers. The backbone, optimizer settings, epoch budget, and split are shared within each task. Scorer parameters and initialization RNG consumption are matched; the zero-field scorer is inactive. The gradient scorer produces bounded node potentials whose edge differences are flat. The unrestricted scorer produces bounded antisymmetric edge values. Neither the gradient arm nor the zero arm removes sparse propagation.

Each run contributes test at its validation-best checkpoint. Synthetic RMSE is divided by 25; bars and subscripts show the sample SD across all three matched runs. The arms vary the field family, which can change both phase and transport amplitude.

For the fixed first validation batch at the selected checkpoint, we build a deterministic breadth-first spanning forest on the simple undirected support. For every non-tree edge, its discrepancy from the integrated tree potential is the signed sum along the corresponding fundamental cycle. We record all these cycle sums at each insertion layer. Across the gradient controls the largest absolute cycle sum is 7.153×10^{-7}, consistent with floating-point residuals; the zero controls are exactly zero. Eight unrestricted runs have a nonzero layer-averaged cycle RMS on this batch; one Monochromatic-0 run has zero. Cycle RMS is measured on this batch in the chosen fundamental basis, using raw Φ_C rather than frequency-scaled phases modulo 2π.

The gradient field has the best Monochromatic-0 mean. On PascalVOC-SP, the unrestricted field attains the highest mean ($0.37568_{\pm 0.01188}$, against $0.37408_{\pm 0.01999}$ for zero and $0.37228_{\pm 0.01248}$ for gradient). For Peptides-struct, unrestricted is only marginally better than zero in mean and wins on one of three individual runs. For MolHIV, zero outperforms unrestricted on all three runs, despite the latter's nonzero measured circulation. The benefit of learned phases therefore varies by task and is not monotone in circulation magnitude.

H.2 Validation sensitivity

The retained grid is $L \in \{4, 8, 16\}$ and $z \in \{0.4, 0.6, 0.8\}$ on all three tasks, with the first run fixed in advance. The field is unrestricted and the task settings match the preceding experiment. Each cell is the best validation metric over the full epoch budget. Test scores are not used to rank cells or update the headline configuration.

The best grid cells are $(8, 0.8)$ on Monochromatic-0 and Peptides-struct, and $(8, 0.6)$ on MolHIV. Their respective validation ranges are 0.039–0.042 nRMSE, 0.254–0.261 MAE, and 0.815–0.839 AUROC. The validation score varies non-monotonically with walk length.

H.3 Isolated full-model time and allocated memory

Each of the 15 retained dataset–method configurations has three independent timing runs on an isolated H200 GPU. We exclude one warm-up epoch, average three subsequent complete training epochs within a run, then report the mean and sample SD across runs. Each timing process occupies its GPU alone. The timed region includes data iteration, host-to-device transfer, full-model forward/backward, optimizer updates, and training logger updates. Validation, test evaluation, and checkpoint writes are outside this training-epoch measurement. The peak is PyTorch allocated device memory over the measured epochs.

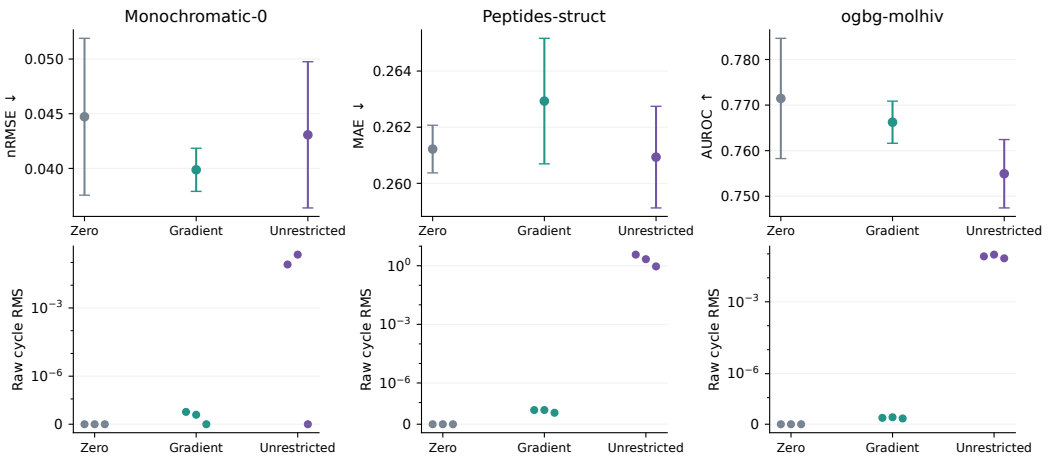

Figure H.1: **Prediction and circulation under different edge fields.** Top: paired test metrics at each validation-best checkpoint, mean and sample SD over three runs. Bottom: one point per run, obtained by averaging raw fundamental-cycle RMS across insertion layers with cycles in the first validation batch. The symmetric-log axis retains exact zeros. The one unrestricted Monochromatic-0 run is flat on this diagnostic batch.

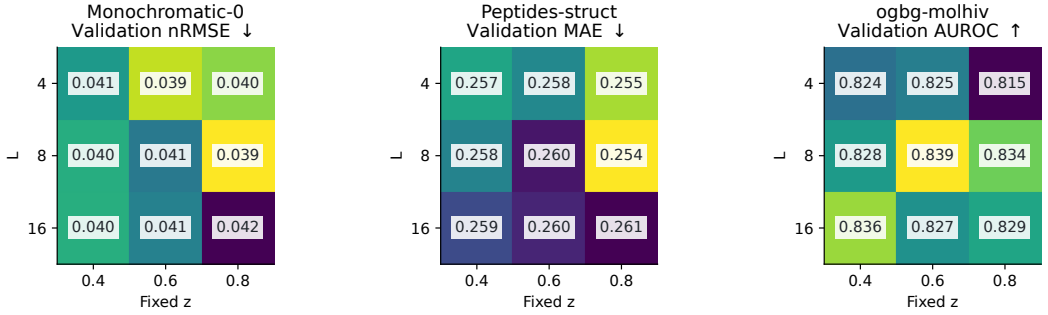

Figure H.2: **Exploratory $L \times z$ sensitivity, fixed run.** Each cell shows the best validation score of one run, to three decimals. Color scales are independent across tasks. Brighter cells indicate better scores. All 27 cells are retained, including the three reused mechanism cells.

AW time and allocated memory increase with L on all three measured tasks. At $L = 8$, AW takes $2.091\times$, $1.703\times$, and $1.693\times$ WIRE's epoch time on Monochromatic-0, Peptides-struct, and MolHIV, respectively, and allocates more memory in all three cases. Size-dependent layer timings and their backend dispatch effects are examined separately in Appendix F.

Loader initialization uses existing prepared caches. The inventory covers enumerable PyG processed files; spectral preparation and the separate AW-only ReLU replay are outside these timings.

I Multi-Route Capability and Matched Mixing Controls

This study compares learned phase transport with ordinary multi-hop mixing, using a fixed-graph diagnostic and a paired predictive experiment.

I.1 A fixed-input two-route phase intervention

For the graph in Figure I.1, a feature supported only at v reaches u for the first time at depth two. With $z = 0.8$,

$$[\mathcal{F}^{(2)}_{z,1}]_{uv} = \frac{z^2}{4}(1 + e^{i\Delta}), \qquad \frac{|[\mathcal{F}^{(2)}_{z,1}]_{uv}|^2}{z^4/4} = \cos^2(\Delta/2).$$

Table H.1: **Isolated H200 full-model cost.** Three timing runs, mean$_{\pm SD}$ to three decimals. Peak GiB is each run's maximum allocated memory over its three measured epochs. Loader seconds use existing prepared caches and are listed separately. The real tasks use FAVOR+.

Dataset	Method	Train s/epoch	Peak GiB	Loader s
Monochromatic-0	NoPE	$7.818_{\pm 0.076}$	$0.097_{\pm 0.000}$	$8.601_{\pm 0.049}$
Monochromatic-0	WIRE	$10.292_{\pm 0.535}$	$0.098_{\pm 0.000}$	$8.580_{\pm 0.047}$
Monochromatic-0	AW-RoPE, $L = 4$	$16.548_{\pm 0.202}$	$0.115_{\pm 0.000}$	$8.807_{\pm 0.102}$
Monochromatic-0	AW-RoPE, $L = 8$	$21.516_{\pm 0.115}$	$0.130_{\pm 0.000}$	$8.730_{\pm 0.295}$
Monochromatic-0	AW-RoPE, $L = 16$	$31.770_{\pm 0.365}$	$0.159_{\pm 0.000}$	$8.789_{\pm 0.196}$
Peptides-struct	NoPE	$3.493_{\pm 0.043}$	$2.801_{\pm 0.026}$	$31.090_{\pm 0.673}$
Peptides-struct	WIRE	$3.766_{\pm 0.050}$	$3.027_{\pm 0.032}$	$31.233_{\pm 0.329}$
Peptides-struct	AW-RoPE, $L = 4$	$5.213_{\pm 0.066}$	$4.740_{\pm 0.058}$	$32.311_{\pm 0.928}$
Peptides-struct	AW-RoPE, $L = 8$	$6.414_{\pm 0.024}$	$6.276_{\pm 0.110}$	$31.448_{\pm 0.460}$
Peptides-struct	AW-RoPE, $L = 16$	$8.958_{\pm 0.048}$	$9.341_{\pm 0.219}$	$30.926_{\pm 0.378}$
ogbg-molhiv	NoPE	$74.993_{\pm 0.446}$	$0.540_{\pm 0.001}$	$35.733_{\pm 0.173}$
ogbg-molhiv	WIRE	$83.769_{\pm 0.840}$	$0.549_{\pm 0.005}$	$35.583_{\pm 0.454}$
ogbg-molhiv	AW-RoPE, $L = 4$	$116.799_{\pm 1.356}$	$0.731_{\pm 0.013}$	$36.010_{\pm 0.123}$
ogbg-molhiv	AW-RoPE, $L = 8$	$141.839_{\pm 1.941}$	$0.870_{\pm 0.024}$	$35.705_{\pm 0.741}$
ogbg-molhiv	AW-RoPE, $L = 16$	$189.056_{\pm 0.652}$	$1.144_{\pm 0.036}$	$35.001_{\pm 0.391}$

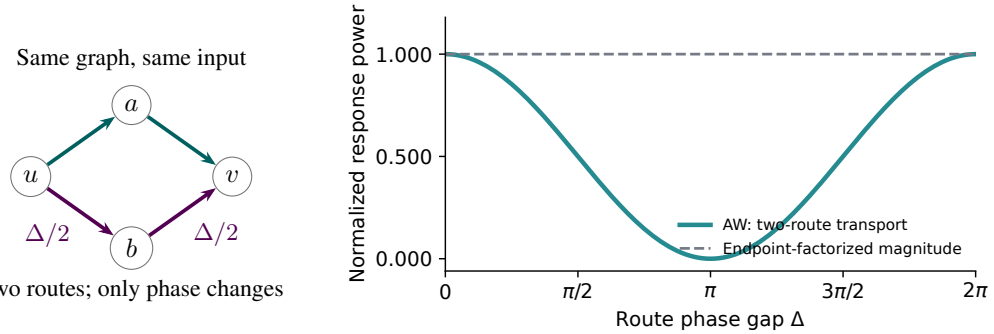

Figure I.1: **Route phase changes transport on a fixed graph.** The diamond has both edge orientations, uniform row transition weights, and a unit real input at v. Arrows show walk indices from u toward v; the row action gathers that input back to u. Only $a_{ub} = a_{bv} = \Delta/2$ and their negatives on reversed edges vary; all other displacements are zero, with $\omega = 1$. Every displacement remains in $[-\pi, \pi]$. At $L = 2$, the two routes have equal weight $1/4$. AW's normalized response power changes from reinforcement to cancellation. The dashed line is the magnitude of an endpoint-factorized coefficient with the same phase-free mixing weights, not a measured WIRE network.

We evaluate 129 uniformly spaced phases using the production real paired-channel sparse recurrence in float64 and compare against this closed form. The dashed comparison line in Figure I.1 is the magnitude of an endpoint-factorized coefficient with the same phase-free mixing weights, not a measured WIRE network. The graph, input and phase-free transition weights stay fixed; only the edge phases vary. A scalar endpoint factor $\overline{g(u)}g(v)$ has unit magnitude and cannot change the coefficient magnitude by itself. The diagnostic concerns that rotary coefficient, while a WIRE backbone can also encode graph structure in its features.

I.2 Two complementary prediction tasks

Route-interference protocol (Table 3). Two length-eight routes connect marked endpoints, forming an undirected 16-node cycle. Two separated $A \to B$ motifs occupy internal nodes. The label distinguishes both motifs on one route from one motif on each route. Moving a complete motif produces an opposite-label pair with the same endpoint features and feature multisets at each endpoint-distance shell. Six input channels encode the endpoints, motif types and two Gaussian nuisance values (SD 0.1); motif strengths vary in $[0.95, 1.05]$. Pairs share nuisance, node permutations and independently rotated/reflected bases of the first two nontrivial Laplacian eigenspaces. The latter enter WIRE rotations, not the pointwise encoder.

Table I.1: **Capability ladder on the constructed tasks.** Each row adds one ingredient. Check marks solve at high accuracy, crosses sit at chance. "(proof)" marks entries covered by an impossibility statement; the route task uses endpoint readout and full networks, the feature-conditional task uses full networks on unseen sizes. The ladder compares field classes within the walk-transport operator interface; learned restriction-map transports are discussed in Appendix B.9.

Field class	Ingredients	Route	Feature-cond.
Mixing (Zero)	transport only	×	× (proof)
WIRE	endpoint phases	×	×
Static field (random, learned)	circulation	–	× (proof)
Dynamic local gradient	adaptivity, local	×	× (proof)
Dynamic nonlocal Flat	adaptivity, nonlocal	–	✓ (unstable)
Sparse AW (ours)	circulation + adaptivity	✓	✓

Table I.2: The two constructed tasks test different capabilities under specified readouts. These are comparison axes, not a nested hierarchy of all possible graph models.

Task	Collision or empirical failure	Successful construction
Route interference, endpoint	Mixing (Zero), WIRE and pointwise gradient transport	AW route-phase interference
Feature-conditioned cycle, full block	Static shell transport and pointwise gradients; WIRE and random fixed fields fail empirically	Local edge AW or dynamically projected nonlocal Flat

The endpoint classifier applies one transport to pointwise encodings and reads the marked endpoint; the attention classifier uses a full block. Both use width 16, fixed $z = 0.8$, and $L = 8$ for Sparse. Exact replaces only the truncated transport by a complete resolvent solve. All arms have 2,618 parameters with endpoint readout or 4,810 with attention; smaller positional modules receive active pointwise adapters. Shared network initializations match within runs. There are 1,024/256/1,024 train/validation/test pairs; pairs stay within splits. Adam uses learning rate 0.003, no weight decay, 32 pairs per update and 2,000 updates. Selection minimizes validation BCE, with the earliest tie, before test evaluation. All 50 Sparse/control and ten Exact fits are retained.

For endpoint readout, $[p(P)]_{uw}$ depends only on the cycle distance from u to w. Identical shell-feature multisets therefore give identical outputs for any phase-free polynomial $p(P)$. A pointwise gradient $a_{ij} = b(x_j) - b(x_i)$ gives the gauge-conjugated action $G^* p(P) G$, where $G_{ii} = \exp(i\omega b(x_i))$, and preserves the same collision. Endpoint-factorized rotation also preserves it. AW can instead accumulate different phases on the two routes, exposing the response power in Figure I.1. With a full attention block, mixing and gradient controls succeed too. This task isolates a transport capability under its endpoint readout condition.

I.3 Feature-conditioned classification on anonymous cycles

Graph family and held-out information. On C_n, $n \geq 12$, the two color arrangements are

$$A_n = 0^{n-12}\,012001100100, \qquad B_n = 0^{n-12}\,011001200100,$$

with labels zero and one. Here 0^{n-12} denotes a run of zero-colored nodes, not a numerical power. Each node receives a constant, a three-way color indicator and two color-specific Gaussian nuisance values with SD 0.1. All nodes of a color have identical features within a graph; the nuisance draw is shared within each opposite-label pair. A pair has identical topology, size and feature multiset, but different feature arrangements. Node order is randomly permuted, and no persistent node or edge identities enter the model. Graph-only spectral bases are randomly rotated and reflected within each of the first two nontrivial eigenspaces, shared within the pair, for the WIRE arms.

Training sizes are $\{12, 16, 20, 24, 28, 32\}$ with 64 pairs per size; validation sizes are $\{14, 18, 22, 26, 30\}$ with 32 pairs per size; test sizes are $\{15, 17, 19, 21, 23, 25, 27, 29, 31\}$ with 128 pairs per size. Each split uses independent paired instances and disjoint sizes. The primary test measures unseen sizes within the training size range; separate 48/64-node diagnostics assess extrapolation beyond that range.

Eight matched arms. Let $D = I + zP + z^2P^2$, $z = 0.8$, and let h_i denote the pointwise encoded/adapted feature before transport. Mixing (Zero) uses D on Q and K. WIRE uses nodewise rotation from the spectral coordinates; WIRE + mixing applies D before this rotation. The learned static polynomial has separate arbitrary trainable matrices M_k^Q, M_k^K for $k = 0, 1, 2$: $Q' = \sum_k (P^k Q) M_k^Q$ and likewise for K. The same matrices apply at every graph size. It includes, and is stronger than, the deterministic static shell transports in Proposition I.1.

The random fixed field draws independent $N(0, 0.8^2)$ displacements on one orientation of each edge and negates them on reversal. Each new paired instance receives a new draw, shared across its labels and fixed thereafter, including at validation/test sizes. Its rotary frequencies are trainable. This supplies random symmetry breaking but no persistent edge identity. It is an empirical control, outside the deterministic static-field theorem. The topology-only Static and Flat fields of Appendix C.6 both equal zero on cycles, so their result is the Mixing (Zero) row rather than an additional independent fit.

AW uses the production local antisymmetric scorer on (h_i, h_j). Dynamic local gradient uses the same scorer parameters to form a bounded pointwise potential $b(h_i)$, then takes its exact difference $a_{ij} = b(h_j) - b(h_i)$. The dynamic nonlocal Flat arm uses the same initial edge scorer as AW and projects its output onto gradients. Choose either cycle orientation, let $\widetilde{a}_i$ be the scorer's displacement on edge $i \to i + 1$, and set

$$a_i^{\mathrm{flat}} = \widetilde{a}_i - \frac{1}{n} \sum_{j=0}^{n-1} \widetilde{a}_j, \qquad a_{i+1,i}^{\mathrm{flat}} = -a_i^{\mathrm{flat}}. \tag{I.1}$$

This is the unweighted incidence-matrix least-squares projection on a cycle. Its sum is zero, so a potential exists; both that potential and the projection depend on features beyond a single node. Reversing the chosen orientation leaves the projected directed field unchanged. It provides no canonical node identity. This dynamic Flat is distinct from the fixed Flat in Appendix C.6.

Training and selection. All arms have 4,810 trainable parameters, width 16, one full softmax attention head, residual/normalization and feedforward processing, and mean pooling with a trainable classifier. There is one graph-aware Q/K transport with $L = 2$, not a frozen endpoint readout. Smaller positional modules receive active pointwise adapters. Common network tensors have identical initial-state hashes within each run; the three dynamic-field arms have identical full initial tensors. Matrix initializations use a shared gain of 3. Adam uses learning rate 0.0003, no weight decay and 10,000 updates. Each update samples one training size uniformly and 16 pairs with replacement. With graph logits ℓ^-, ℓ^+ and ordinary binary cross-entropy $\mathcal{L}_{\mathrm{BCE}}$, every arm minimizes

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \mathbb{E}_{\mathrm{pairs}} \operatorname{softplus}\big(-100(\ell^+ - \ell^-)\big).$$

Pairing uses training labels only. The inference function processes one graph independently; neither another graph nor the pair index is an input. Every 100 updates, validation BCE is averaged equally across sizes. Its minimum selects the checkpoint, with the earliest tie; test instances are generated only after training and selection finish. The primary metric is the equally weighted size-average accuracy.

Development used fresh runs and validation data. An earlier ordinary-BCE cohort gave AW $83.91 \pm 21.35\%$ and is retained separately. The final protocol freezes fresh data and fresh runs; the two protocols are not pooled, and no run is excluded.

Proposition I.1 (Complete-block collision for static and local-potential transport)**.** *For paired A_n, B_n, a pointwise encoder followed by a degree-two static matrix-polynomial Q/K transport, one full attention block and invariant pooling has identical logits for both labels, for every choice of its parameters. The same holds for a degree-two transport whose connection is the gradient of a pointwise node potential. Consequently each such classifier has exactly 50% accuracy on the balanced paired set.*

Proof. Define a node signature as its color together with the color histograms at distances one and two. The two patterns have identical multisets of these signatures at $n = 12$. Inserting $k = n-12 \geq 1$ zeros into the common length-three zero run removes, in both patterns, one signature $(0; 2, 0, 0; 0, 2, 0)$, adds two $(0; 2, 0, 0; 1, 1, 0)$, and adds $k - 1$ $(0; 2, 0, 0; 2, 0, 0)$. Thus equality holds for all $n \geq 12$. The patterns are nonetheless nonisomorphic: the unique color-2 node fixes an isomorphism and its

Table I.3: All AW and dynamic nonlocal Flat runs, accuracy (%). Test is the primary unseen-size split; 48/64 nodes is a separate extrapolation diagnostic. Zeroing phases gives 50% for every run on the primary test.

Arm	Evaluation	20	21	22	23	24
Sparse AW	Test	100.00	100.00	100.00	93.45	99.91
Sparse AW	48/64 nodes	92.97	81.64	100.00	50.00	71.88
Dynamic nonlocal Flat	Test	99.91	100.00	99.96	74.26	100.00
Dynamic nonlocal Flat	48/64 nodes	50.78	82.81	100.00	50.00	97.27

differently colored neighbors fix the orientation, but the next four colors towards its color-0 neighbor are 0011 in A_n and 0010 in B_n.

Pointwise features depend only on color within each graph. On a cycle, Ph depends on the distance-one histogram, and P^2h on the node's color and distance-two histogram, including the backtracking term $h_i/2$. Hence each static transported query/key is a function of this signature, even with arbitrary channel matrices. For a pointwise gradient, the gauge factors in $G^*(I + zP + z^2P^2)G$ also depend only on the endpoint colors, preserving the signature dependence. The multisets of complete tuples (h_i, Q'_i, K'_i, V_i) therefore coincide. Global attention, residual/normalization and pointwise feedforward processing commute with the matching permutation, and invariant pooling gives identical graph logits. □

The static-field scope follows from cycle symmetries. A deterministic topology-only real antisymmetric displacement that is equivariant to node relabeling is constant on oriented edges by rotations/reflections, and antisymmetry makes it zero. A directly specified $U(1)$ connection also admits the constant -1 connection: Hermitian reversal requires $U = \overline{U}$, so $U = \pm 1$. Both cases give static shell polynomials covered by the proposition, including the uniform π connection on odd cycles. Random edge marks, spectral symmetry breaking, additional graph-aware layers, larger L, or arbitrary nonlocal potentials are outside the proof. WIRE's near-chance result is empirical, not this theorem applied to all spectral graph networks.

Seed variability and extrapolation. Table I.3 retains every run. Dynamic Flat's low run is a partial fit, not a collapse to chance, with 72.81% validation and 74.26% test accuracy. Its other four runs are at least 99.91%. The observed spread describes sensitivity to the shared training protocol; it establishes neither an AW–Flat accuracy advantage nor an intrinsic stability ordering. The 48/64-node results also limit the size-generalization claim to the primary within-range test. No normalization mechanism is inferred from these training outcomes.

Field diagnostics and frozen-model interventions. We inspect the first validation pair at $n = 18$ for every fixed run; Figure I.2 shows one representative. The learned AW edge field changes between feature contexts. For dynamic Flat, reconstruct the zero-mean potential by accumulating a_i^{flat} around the cycle. The maximum residual in $a_i^{\mathrm{flat}} = b_{i+1} - b_i$ is below 2×10^{-15} across the runs. Nodes with identical raw features nevertheless have potentials differing by 1.71–3.14 within a graph; the local-gradient control has no such variation up to 10^{-10}. This verifies a nonlocal potential rather than assuming one. There is no prescribed teacher potential to recover.

AW has nonzero cycle sums on these pairs (cycle RMS 1.91–2.79), but each pair's two labels have the same total circulation. The task therefore probes the arrangement of feature-conditioned transport, not a label-dependent total holonomy. Dynamic Flat's successful runs also show that nonzero circulation is not necessary here. Setting all edge displacements to zero while holding each fitted AW or dynamic Flat network fixed returns every run to 50% test accuracy. This intervention establishes reliance on phases in the fitted solution; the collision proof and independently trained controls establish the specified model-class separation. Checkpoint predictions, validation choices, source/data hashes, permutation covariance and graph-wise inference independence have been replayed.

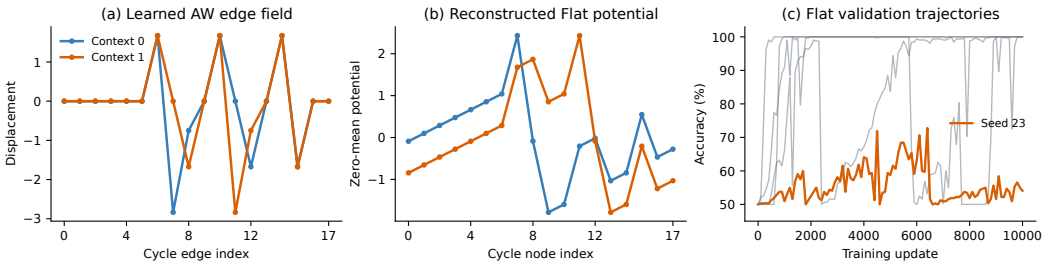

Figure I.2: **Learned fields and Flat run variability.** (a) AW edge displacements for the first paired validation example at $n = 18$. (b) The dynamic Flat control's reconstructed zero-mean potential on that pair. Cycle indices are used only for this display, not supplied as model features. (c) All five Flat validation trajectories; the partially fitted run is highlighted. No smoothing is applied.

Table I.4: Completed matched mixing controls. Test at validation-best, mean$_{\pm\text{sample SD}}$ over three matched runs, three decimals. Mixing is phase-free propagation; WIRE + mixing applies propagation before WIRE rotation. Bold marks the best unrounded mean.

Dataset	Metric	WIRE	Mixing	WIRE + mixing	*Sparse* AW-RoPE
Monochromatic-0	nRMSE ↓	$0.073_{\pm 0.008}$	$0.044_{\pm 0.005}$	$0.045_{\pm 0.006}$	$\mathbf{0.043}_{\pm 0.006}$
Peptides-struct	MAE ↓	$0.257_{\pm 0.002}$	$0.255_{\pm 0.002}$	$0.257_{\pm 0.001}$	$\mathbf{0.253}_{\pm 0.001}$

I.4 Completed matched predictive study

Monochromatic-0 and Peptides-struct each use all four arms over three matched runs. The epoch budgets, learning rates, and batch sizes are inherited from Appendix H.1. Synthetic attention remains dense; the real task uses the same generalized ReLU linear-attention kernel. The field scorer configuration is inherited without tuning.

Write $D_{L,z} = \sum_{k=0}^{L} z^k P^k$, and let R_g denote WIRE's node-coordinate rotation. With $L = 8$ and fixed $z = 0.8$, the query/key actions are:

Arm	Action on each projected query or key X
WIRE	$R_g X$
Phase-free mixing	$D_{L,z} X$
WIRE + mixing	$R_g D_{L,z} X$
AW-RoPE	$\mathcal{F}^{(L)}_{z,\omega} X$

Mixing precedes WIRE rotation in the third arm. Values, backbone widths, local message passing, and input encodings are unchanged across arms. All arms construct the same coordinate encoder to preserve shared initialization and training RNG consumption; its output is unused by AW and phase-free mixing. Positional-module initialization cannot advance the RNG stream for later shared backbone layers. Initial shared-state hashes are checked within each task and run.

WIRE and WIRE + mixing have exactly the same parameters; the latter adds parameter-free propagation. AW retains its scorer and frequency parameters. Table I.5 gives allocated, positional and unused encoder counts. Shared L and z match the propagation budget.

The protocol freezes all configurations before new results. Every epoch is checkpointed; the test value is paired with the validation-best epoch. All three runs contribute to $\text{mean}_{\pm\text{SD}}$, with three-decimal formatting.

All runs are complete and pass trajectory, checkpoint and shared initialization checks. Table I.4 retains every arm and run of the study.

AW-RoPE attains the lowest mean on both tasks. The matched arms attribute most of the gain to walk transport itself, with phase learning contributing task-dependent improvements.

Table I.5: Parameter accounting for the matched mixing study. Counts are identical across the three runs of each row. Allocated includes modules whose outputs are unused. The final column identifies unused coordinate-encoder parameters.

Dataset	Arm	Allocated	Positional	Unused encoder
Monochromatic-0	WIRE	39644	320	0
Monochromatic-0	Mixing	39324	0	85
Monochromatic-0	WIRE + mixing	39644	320	0
Monochromatic-0	Sparse AW-RoPE	39656	332	85
Peptides-struct	WIRE	506729	1920	0
Peptides-struct	Mixing	504809	0	270
Peptides-struct	WIRE + mixing	506729	1920	0
Peptides-struct	Sparse AW-RoPE	529837	25028	270

I.5 Training curves and attention on fixed validation graphs

We visualize the existing Monochromatic-0 WIRE and AW-RoPE runs from the matched mixing study above. Following the graph-based presentation of Graph-RoPE (Reid et al., 2026), we show the task inputs alongside learned attention, and plot the recorded training loss. All panels use the local logs and checkpoints.

The training panel of Figure 3 includes all 250 epochs of the three runs. Thin lines show individual runs and thick lines their arithmetic means, using recorded MSE on the unnormalized target with a logarithmic axis and no smoothing. AW-RoPE reaches lower training MSE in these runs; Table I.4 gives the validation-selected test results. Both methods use dense attention and the same backbone and training settings, with different active positional parameter counts.

For Figure I.3, we fix the first run and the first three validation graphs before inspecting attention. Each method uses its own validation-best checkpoint. We extract the final layer's single-head softmax attention in evaluation mode and display the weights from node 0 to every key node. No parameters are updated. The graph layouts and color scale are shared, and the full attention matrices are retained. These are descriptive comparisons of independently trained models.

The six complete training trajectories, checkpoint hashes, fixed graph inputs, and attention matrices are stored with the artifact. The exporter checks the original shared-initialization hashes, validation-best epoch, checkpoint loading, and unchanged model state during attention extraction.

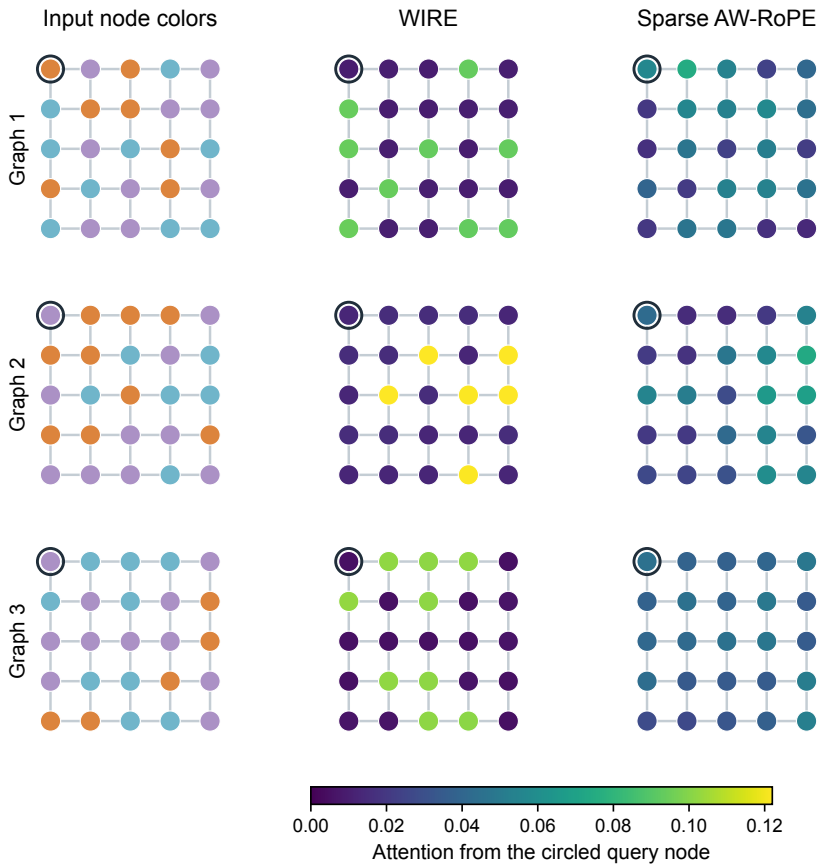

Figure I.3: **Attention on fixed Monochromatic-0 validation examples.** Left: the three categorical node colors supplied to the task. Center and right: final-layer attention probabilities from the circled node for WIRE and *Sparse* AW-RoPE. Rows are the first three validation graphs in their stored order. All six attention panels use one common color scale, without per-panel rescaling. Seed 0 and validation-best checkpoints are fixed.