LEARNING PROPAGATION GEOMETRY FROM MESSAGE-PASSING FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning local geometry enables graph neural networks (GNNs) to adapt how they compare and integrate neighborhood information. However, estimating geometry from aggregated representations can overlook variation among individual messages and dependencies across feature dimensions. We propose GeoF, a recurrent framework that jointly evolves node features and propagation geometry through message-passing feedback. Each node maintains a local symmetric positive-definite geometry, initialized from a structure-aware prototype atlas and parameterized in block log-triangular coordinates. At each step, the geometry determines neighborhood weights, while triangular frame transport maps transformed source messages into the target node's local coordinates before aggregation. Weighted second-order statistics of residuals between aligned messages and the transformed target state capture directional variation and within-block dependencies, yielding a geometric update target. A shared controller learns complementary corrections through task supervision. A bounded log-triangular update combines these corrections, the target, and the previous geometric state while preserving positive definiteness. The geometry governs subsequent propagation, closing the feedback loop. With parameters shared across recurrent steps, task-specific readouts support node classification, link prediction, and graph classification. Experiments on benchmark datasets show that GeoF consistently outperforms state-of-the-art GNN baselines.

1 INTRODUCTION

Graph neural networks (GNNs) learn representations by repeatedly transforming and aggregating neighborhood information (Gilmer et al., 2017; Xu et al., 2019). Convolution, attention, and multi-hop aggregation organize this computation through different propagation operators (Kipf & Welling, 2017; Veličković et al., 2018; Abu-El-Haija et al., 2019). Beyond selecting and weighting neighbors, message passing involves comparing features and transforming messages before they are combined. Geometry makes these operations explicit: local metrics determine how feature directions contribute to comparisons, while coordinate mappings specify how messages are expressed across local spaces (Chami et al., 2019; Bodnar et al., 2022; Wang et al., 2026). Learning propagation geometry provides a means of adapting neighborhood interactions to graph structure and the prediction task.

Existing geometric GNNs adapt propagation at two levels: the choice of representation space and the local rules for comparing and transforming messages. At the first level, curvature learning and node-specific space selection address the mismatch between prescribed geometric assumptions and graph structure by adapting the curvature or geometric family to the data (Chami et al., 2019; Fu et al., 2021; Lee et al., 2023). At the second level, feature-dependent approaches refine how neighboring information is compared and transformed by learning local maps for message transformation (Bodnar et al., 2022) and node-wise metrics for feature comparison (Wang et al., 2026). Whereas the first level concerns which geometry is appropriate, the second concerns how local interactions should adapt to the representations being processed. This makes the information used to infer those local rules consequential. ARGNN estimates its metrics from target features and neighborhood means (Wang et al., 2026), compressing the neighborhood into a summary that can conceal relationships across feature dimensions. Two neighborhoods can have identical means but different patterns of joint variation. Given identical target features, the metric estimator assigns them the same geometry, even when these differences matter for prediction. Examining individual incoming messages relative to the target state under the current geometry can expose variation obscured by the mean, providing

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.

additional information for subsequent geometric updates. This motivates our central question: ***Can feedback from message passing improve the learning of local propagation geometry for prediction?***

Leveraging this feedback presents three interconnected challenges. *First, message differences require a consistent geometric interpretation.* When nodes use different local coordinate frames, direct comparisons can conflate coordinate mismatch with variation in the messages themselves. *Second, feedback needs to capture how individual messages vary, rather than only how their average changes.* Opposing contributions can cancel during aggregation, obscuring directional variation and dependencies across feature dimensions. *Third, residual statistics describe how messages vary, but not which directions of variation matter for prediction.* Directions with large residual energy need not be those most relevant to the downstream task. Geometric updates therefore need to incorporate task supervision while remaining well-defined and bounded across repeated propagation steps.

To address these challenges, we propose GeoF, a recurrent framework that jointly evolves node features and local geometry through message-passing feedback. Each node maintains a feature state and a symmetric positive-definite geometry, initialized from a structure-aware prototype atlas and parameterized in block log-triangular coordinates (Lin, 2019). First, to resolve coordinate mismatch, *triangular frame transport* maps transformed source messages into the target node's local coordinates, establishing a common reference for comparison and geometry-weighted aggregation. Second, to capture variation obscured by averaging, *second-order residual feedback* accumulates weighted outer products of residuals between individual aligned messages and the transformed target state. These statistics capture directional variation and within-block dependencies, yielding a geometric update target. Third, *task-guided geometry evolution* complements this target with corrections learned by a shared controller through end-to-end task supervision. A bounded log-triangular update combines the target, corrections, and previous geometric state, preserving positive definiteness across recurrent steps. The updated geometry determines neighborhood weights and frame transformations for subsequent propagation, closing the feedback loop. All recurrent parameters are shared across steps, and task-specific readouts support node classification, link prediction, and graph classification.

Our contributions are summarized as follows: (1) We investigate message-passing feedback as an information source for local geometry learning, formulating propagation geometry as a persistent node-wise state that evolves jointly with node features. (2) We develop a closed-loop mechanism in which local geometry governs neighborhood weighting and frame-aligned message propagation, while weighted second-order residual statistics and task-supervised corrections drive its evolution. (3) We evaluate GeoF on node classification, link prediction, and graph classification, where it demonstrates consistent gains over general, manifold-based, and adaptive GNN baselines.

2 Related Work

Operator and Structure Adaptation in GNNs. GNN adaptation has been explored through propagation operator design and graph structure learning. Higher-order neighborhood mixing integrates information across hop distances, while adaptive propagation adjusts the contributions of different propagation scales (Abu-El-Haija et al., 2019; Xu et al., 2018; Gasteiger et al., 2018; Liu et al., 2020). To adapt connectivity to downstream tasks, graph structure learning constructs meta-path graphs or iteratively refines graph structure together with node representations (Yun et al., 2019; Chen et al., 2020). Related approaches refine noisy adjacency matrices for robust prediction (Jin et al., 2020) or learn latent edges that support message passing beyond observed connectivity (Wu et al., 2022). These studies adapt how messages are aggregated and which nodes communicate. GeoF learns node-wise propagation geometry that explicitly governs message comparison and transport.

Geometric Modeling and Adaptive Propagation Spaces. Geometric GNNs introduce inductive biases through representation spaces and local maps for message passing. Hyperbolic methods exploit negative curvature to represent hierarchical and scale-free structures (Liu et al., 2019; Chami et al., 2019), while sheaf-based diffusion learns local linear maps that relate node spaces for feature comparison and propagation (Bodnar et al., 2022). Geometric adaptation further includes learning graph-dependent curvature (Fu et al., 2021) and selecting between Euclidean and hyperbolic spaces at the node level (Lee et al., 2023). Adaptive Riemannian models learn anisotropic node-wise metric tensors from node features and neighborhood means, adapting geometry to individual nodes (Wang et al., 2026). GeoF maintains local geometry as a persistent state across propagation steps and updates it through second-order statistics of aligned message residuals and task-guided corrections.

3 METHODOLOGY

Problem Setup. Let $G = (V, E, X)$ be an undirected attributed graph with $n = |V|$ nodes, adjacency matrix A, and node features $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times F_0}$. Each node has a structural signature $u_i \in \mathbb{R}^q$, and $U = [u_1, \ldots, u_n]^\top$ collects these signatures. We construct u_i from normalized degree and random-walk return probabilities and keep it fixed during propagation. Let $\mathcal{N}(i)$ denote the neighbors of node i and $\widetilde{\mathcal{N}}(i) = \mathcal{N}(i) \cup \{i\}$ include the self-loop. We consider three prediction tasks: node classification predicts node labels $y_i \in \{1, \ldots, C\}$, link prediction estimates whether a candidate node pair (u, v) is connected, and graph classification predicts a graph label $Y_G \in \{1, \ldots, C\}$. Here, C denotes the number of classes for the corresponding classification task.

Overview. GeoF couples node features and local geometry through three recurrent components. We initialize feature states and construct local geometries from a structure-aware prototype atlas in block log-triangular coordinates. The recurrent computation then comprises: (i) *Triangular Frame Transport*, which uses the current geometry to weight neighbors and align transformed messages with the target node's local frame for aggregation and feature updates; (ii) *Second-Order Residual Feedback*, which summarizes aligned message residuals to capture directional variation and within-block dependencies, producing a target metric on the node's local coordinates; and (iii) *Task-Guided Geometry Evolution*, which combines this target with task-supervised corrections and the previous geometric state to update the geometry. Together, these components form a closed feedback loop, with parameters shared across steps. Task-specific readouts produce node, link, and graph predictions, and the corresponding learning objectives train the framework end to end.

3.1 GEOMETRIC STATES AND INITIALIZATION

Learning propagation geometry through recurrent feedback requires a compact parameterization that preserves positive definiteness under repeated updates and an initialization informed by local structure. Inspired by log-Cholesky parameterizations (Lin, 2019), we use block log-triangular coordinates and a shared prototype atlas. The former models within-block interactions, while the latter uses structural signatures to initialize node-specific geometries.

At step l, node i maintains a local feature state $\xi_i^l \in \mathbb{R}^d$ and geometric coordinates $Z_i^l = \{z_{i,b}^l\}_{b=1}^B$. We partition the feature space into B blocks of size m, with $d = Bm$. Each block is parameterized as $z_{i,b}^l = [a_{i,b}^l, \ell_{i,b}^l]^\top \in \mathbb{R}^{m(m+1)/2}$, where $a_{i,b}^l \in \mathbb{R}^{m(m-1)/2}$ stores strictly lower-triangular entries and $\ell_{i,b}^l \in \mathbb{R}^m$ stores log-diagonal scales. The corresponding triangular frame and local metric are:

$$L_{i,b}^l = \mathrm{mat}_{\mathrm{sl}}(a_{i,b}^l) + \mathrm{Diag}\big(\exp(\ell_{i,b}^l)\big), \qquad g_{i,b}^l = (L_{i,b}^l)^\top L_{i,b}^l, \tag{1}$$

where $\mathrm{mat}_{\mathrm{sl}}$ reconstructs a strictly lower-triangular matrix and the exponential acts elementwise. The positive diagonal makes $L_{i,b}^l$ invertible and $g_{i,b}^l$ positive definite. $L_{i,b}^l$ maps local coordinates into a shared environment, where $g_{i,b}^l$ measures lengths with $\|L_{i,b}^l v\|_2^2 = v^\top g_{i,b}^l v$. Combining the blocks:

$$L_i^l = \bigoplus_{b=1}^B L_{i,b}^l, \qquad g_i^l = \bigoplus_{b=1}^B g_{i,b}^l = (L_i^l)^\top L_i^l, \tag{2}$$

where $\bigoplus$ denotes the block-diagonal direct sum. This representation uses $d(m+1)/2$ geometric coordinates per node while retaining off-diagonal interactions within each block. To bound coordinate magnitudes during initialization and recurrent updates, we define the admissible set $\mathcal{Z} = [-a_{\max}, a_{\max}]^{m(m-1)/2} \times [\ell_{\min}, \ell_{\max}]^m$, where $a_{\max} > 0$ and $\ell_{\min} < \ell_{\max}$ are finite bounds. We denote coordinatewise clipping to this set by $\Pi_{\mathcal{Z}}$. This constraint bounds the geometric coordinates, while positive definiteness follows from the log-triangular reconstruction.

Before propagation feedback becomes available, we initialize geometry from the structural signatures u_i. Let $\{z_{k,b}^{\mathrm{proto}}\}_{k=1}^K$ denote K learnable prototype coordinates for block b. A shared gate assigns each node a distribution over the prototypes, yielding

$$\xi_i^0 = W_{\mathrm{in}} x_i, \qquad \alpha_i = \mathrm{softmax}(W_g u_i + b_g), \qquad z_{i,b}^0 = \Pi_{\mathcal{Z}}\left(\sum\nolimits_{k=1}^K \alpha_{ik} z_{k,b}^{\mathrm{proto}}\right), \tag{3}$$

where W_{in}, W_g, and b_g are learnable parameters. The atlas shares geometric templates across nodes, while the structural gate determines their initial mixtures. These initial states provide the starting point for frame-aligned propagation and subsequent feedback-driven geometric updates.

Lemma 1 (Well-Posed Geometric Parameterization) *Let $m \geq 1$ and $z, z' \in \mathcal{Z}$. The frame $L(z)$ is lower triangular with $\det L(z) = \exp(\sum_t \ell_t) > 0$, so it is invertible and $g(z) \in \mathrm{SPD}(m)$. There are constants $\sigma_+ \geq \sigma_- > 0$ and $\kappa_L > 0$, depending only on m, $a_{\max}$, $\ell_{\min}$, and $\ell_{\max}$, such that*

$$\|L(z)\|_2 \leq \sigma_+, \qquad \|L(z)^{-1}\|_2 \leq \sigma_-^{-1}, \qquad \sigma_-^2\, I_m \preceq g(z) \preceq \sigma_+^2\, I_m,$$

and the reconstruction is Lipschitz: $\|L(z) - L(z')\|_F \leq \kappa_L \|z - z'\|_2$ and $\|L(z)^{-1} - L(z')^{-1}\|_2 \leq \sigma_-^{-2}\kappa_L \|z - z'\|_2$. All statements carry over to the block-diagonal frames $L_i = \bigoplus_b L(z_{i,b})$ and $g_i = L_i^\top L_i$, with $\|z - z'\|_2$ replaced by $\|Z_i - Z_i'\|_F := (\sum_b \|z_{i,b} - z'_{i,b}\|_2^2)^{1/2}$.

Lemma 1 makes the recurrence well-posed: the triangular solves of Eq. (5) and the projected update of Eq. (12) are defined at every admissible state, and the bounds supply the constants for Propositions 1 and 3. The proof, with explicit expressions for $\sigma_\pm$ and κ_L, is in Appendix B.

3.2 Triangular Frame Transport

Message passing across node-specific frames requires a common coordinate reference for comparison and aggregation. GCN (Kipf & Welling, 2017) and GAT (Veličković et al., 2018) transform neighboring features in shared coordinates and weight them through degree normalization or learned attention. However, scalar reweighting alone cannot align distinct frames. We introduce *triangular frame transport* to map transformed source messages through a shared environment into the target node's local coordinates before aggregation. The same geometric state determines neighborhood weights, coupling weighting with alignment and providing a consistent reference for residual computation.

Specifically, at step l, we construct environment representations from the current feature and geometric states and compute neighborhood weights from their pairwise distances:

$$h_i^l = L_i^l \xi_i^l, \qquad e_{ij}^l = \|h_i^l - h_j^l\|_2^2, \qquad \omega_{ij}^l = \frac{\exp(-e_{ij}^l/\tau)}{\sum_{k \in \widetilde{\mathcal{N}}(i)} \exp(-e_{ik}^l/\tau)}, \tag{4}$$

where $j \in \widetilde{\mathcal{N}}(i)$, $\|\cdot\|_2$ denotes the Euclidean norm, and $\tau > 0$ controls the concentration of the weights. The weights satisfy $\omega_{ij}^l \geq 0$ and $\sum_{j \in \widetilde{\mathcal{N}}(i)} \omega_{ij}^l = 1$. To align messages before aggregation, we apply a shared feature transformation $\phi(\xi) = W_\phi \xi$. For each block b, the transformed source message is mapped through the source frame and pulled back into the target frame:

$$m_{j \to i, b}^l = \mathrm{TriSolve}\left(L_{i,b}^l,\, L_{j,b}^l \big(\phi(\xi_j^l)\big)_b\right), \tag{5}$$

where $(\cdot)_b$ extracts the b-th feature block and $\mathrm{TriSolve}(L, y)$ solves $Lw = y$ without explicitly forming L^{-1}. Lemma 1 ensures this solution is unique for every admissible geometric state. The transported message satisfies $L_{i,b}^l m_{j \to i, b}^l = L_{j,b}^l (\phi(\xi_j^l))_b$: both sides represent the same transformed message in the shared environment, while $m_{j \to i, b}^l$ expresses it in the target node's coordinates. We concatenate the blockwise messages and aggregate them using the geometry-conditioned weights:

$$m_{j \to i}^l = \mathrm{concat}\left(m_{j \to i, 1}^l, \ldots, m_{j \to i, B}^l\right), \qquad \bar{\xi}_i^l = \sum\nolimits_{j \in \widetilde{\mathcal{N}}(i)} \omega_{ij}^l m_{j \to i}^l. \tag{6}$$

All terms in $\bar{\xi}_i^l$ are expressed in the target frame. A feature gate balances the state against the aggregated message:

$$r_i^l = \sigma\left(W_r[\xi_i^l \| \bar{\xi}_i^l] + b_r\right), \qquad \xi_i^{l+1} = (\mathbf{1} - r_i^l) \odot \xi_i^l + r_i^l \odot \bar{\xi}_i^l, \tag{7}$$

where $W_r \in \mathbb{R}^{d \times 2d}$ and $b_r \in \mathbb{R}^d$ are learnable parameters, σ is the elementwise sigmoid, $\|$ denotes concatenation, $\odot$ denotes elementwise multiplication, and $\mathbf{1} \in \mathbb{R}^d$ is the all-ones vector. The parameters W_ϕ, W_r, and b_r are shared across nodes and recurrent steps.

Proposition 1 (Frame-Aligned Transport) *Fix a recurrent step l with admissible geometric states, and let $T_{j \to i, b}^l = (L_{i,b}^l)^{-1} L_{j,b}^l$ and $M_j^l = L_j^l W_\phi (L_j^l)^{-1}$. Then $m_{j \to i, b}^l = T_{j \to i, b}^l (\phi(\xi_j^l))_b$ with $\|T_{j \to i, b}^l\|_2 \leq \sigma_+/\sigma_-$, and the frame maps satisfy $T_{i \to i, b}^l = I_m$ and $T_{j \to i, b}^l T_{k \to j, b}^l = T_{k \to i, b}^l$, so their product along any walk depends only on its endpoints. In environment coordinates, the aggregate and the residuals $\delta_{ij}^l := m_{j \to i}^l - \phi(\xi_i^l)$ read*

$$L_i^l \bar{\xi}_i^l = \sum\nolimits_{j \in \widetilde{\mathcal{N}}(i)} \omega_{ij}^l\, M_j^l h_j^l, \qquad L_i^l \delta_{ij}^l = M_j^l h_j^l - M_i^l h_i^l,$$

where each M_j^l is similar to W_ϕ with $\|M_j^l\|_2 \le (\sigma_+/\sigma_-)\|W_\phi\|_2$, and $M_j^l = cI_d$ when $W_\phi = cI_d$.

In environment coordinates, one recurrent step is thus Gaussian-kernel attention with temperature τ, where the frames enter through the energies e_{ij}^l, the transforms M_j^l, and the pull-back $(L_i^l)^{-1}$ that makes the residuals of Sec. 3.3 comparable across neighbors. The proof is in Appendix C.

3.3 Second-Order Residual Feedback

The transport step aligns messages and updates node features under the current geometry, and the aligned messages give a common reference for local comparison. ARGNN estimates local metrics from target features and neighborhood means (Wang et al., 2026), but identical summaries can conceal directional variation and cross-feature dependence. We therefore examine the residuals of individual aligned messages from the transformed target state, not their aggregate, and accumulate their weighted outer products into a geometric target that captures directional variation and within-block dependence. Interactions under the current geometry thereby become feedback for its evolution.

Specifically, for node i and block b, we compute the residual between each aligned message and the transformed target state, $\delta_{ij,b}^l = m_{j\to i,b}^l - \big(\phi(\xi_i^l)\big)_b$, which by Proposition 1 are expressed in the same frame. To capture directional variation beyond the mean, we form a second-moment matrix whose quadratic form measures residual energy along any direction, reusing ω_{ij}^l to preserve aggregation weights:

$$C_{i,b}^l = \sum\nolimits_{j\in\widetilde{\mathcal{N}}(i)} \omega_{ij}^l \delta_{ij,b}^l (\delta_{ij,b}^l)^\top + \epsilon_s I_m, \tag{8}$$

where $\epsilon_s > 0$ provides a positive spectral lower bound. The diagonal entries combine weighted squared residuals with ϵ_s, while the off-diagonal entries capture weighted cross-products within each block. Since residuals are measured relative to the transformed target state rather than centered at their neighborhood mean, $C_{i,b}^l$ is a regularized second-moment matrix. Writing $\bar\delta_{i,b}^l = \sum_j \omega_{ij}^l \delta_{ij,b}^l$ for the weighted mean residual, the classical mean–covariance decomposition gives, for every $v \in \mathbb{R}^m$,

$$v^\top C_{i,b}^l v = \epsilon_s \|v\|_2^2 + \big(v^\top \bar\delta_{i,b}^l\big)^2 + \sum\nolimits_{j\in\widetilde{\mathcal{N}}(i)} \omega_{ij}^l \left[v^\top\big(\delta_{ij,b}^l - \bar\delta_{i,b}^l\big)\right]^2. \tag{9}$$

The last term is the weighted residual covariance: residuals of opposite sign cancel in the mean yet still contribute to $C_{i,b}^l$, which is exactly what mean-based estimators discard. Since the geometric state evolves in log-triangular coordinates, we convert this statistic into a coordinate target for updates:

$$R_{i,b}^l = \mathrm{SChol}(C_{i,b}^l), \qquad \hat z_{i,b}^l = \left[\mathrm{Svec}_{\mathrm{sl}}(R_{i,b}^l)^\top,\quad \log\Big(\mathrm{diag}(R_{i,b}^l)\Big)^\top\right]^\top, \tag{10}$$

where $\mathrm{SChol}(C)$ returns the positive-diagonal lower-triangular factor R with $RR^\top = C + \eta I_m$, $\mathrm{Svec}_{\mathrm{sl}}$ stacks the strictly lower-triangular entries in fixed order, and $\log$ is elementwise. Since Cholesky factorization is bijective on $\mathrm{SPD}(m)$, $\hat z_{i,b}^l$ encodes $C_{i,b}^l + \eta I_m$ losslessly, and $R_{i,b}^l$ maps the local unit ball onto the residual second-moment ellipsoid, stretching high-energy residual directions.

Proposition 2 (Second-Order Refinement) *Fix a step l. Write $\mathcal{R}_{i,b}^l = \{(\omega_{ij}^l, \delta_{ij,b}^l)\}_{j\in\widetilde{\mathcal{N}}(i)}$ for the residual configuration of node i in block b, with weighted mean $\bar\delta_{i,b}^l$, and call an update* first-order *if it depends on $\mathcal{R}_{i,b}^l$ only through the weights and $\bar\delta_{i,b}^l$. Let i, i' be nodes with $\xi_i^l = \xi_{i'}^l$, $Z_i^l = Z_{i'}^l$, $u_i = u_{i'}$, and $\bar\delta_{i,b}^l = \bar\delta_{i',b}^l$ for all b.*

(i) Every first-order update, including the mean-only target $C_{i,b}^{l,\mathrm{mean}} = \bar\delta_{i,b}^l(\bar\delta_{i,b}^l)^\top + \epsilon_s I_m$, the controller of Eq. (11), and the feature update of Eq. (7), returns identical outputs for i and i'; in particular $\xi_i^{l+1} = \xi_{i'}^{l+1}$ and $\Delta z_{i,b}^l = \Delta z_{i',b}^l$.

(ii) The second-order target is injective in the statistic: $C_{i,b}^l \ne C_{i',b}^l$ implies $\hat z_{i,b}^l \ne \hat z_{i',b}^l$. If $\lambda \in (0,1]$ and Eq. (12) is not clipped in block b, then $Z_i^{l+1} \ne Z_{i'}^{l+1}$, and $h_i^{l+1} \ne h_{i'}^{l+1}$ unless $\xi_i^{l+1} \in \ker(L_i^{l+1} - L_{i'}^{l+1})$, a proper subspace.

(iii) If i has at least two neighbors and $\mathcal{R}_{i',b}^l$ ranges over all configurations with the same weights and first moment, those with $C_{i',b}^l = C_{i,b}^l$ form a Lebesgue-null set.

(iv) The target depends on $\mathcal{R}^l_{i,b}$ only through $C^l_{i,b}$ and is therefore strictly weaker than an injective aggregator: for $m = 1$ there are distinct three-point residual sets with equal weights, first, and second moments. Conversely, for $m \geq 2$ the sets $\{0, \pm v\}$ and $\{0, \pm w\}$ with $v = (1, 1, 0, \ldots, 0)^\top$, $w = (1, -1, 0, \ldots, 0)^\top$ and equal weights on $\pm v$, $\pm w$ share the mean-only and the diagonal target $C^{l,\mathrm{diag}}_{i,b} = \mathrm{Diag}\big(\sum_j \omega^l_{ij}\, \delta^l_{ij,b} \odot \delta^l_{ij,b}\big) + \epsilon_s I_m$, yet $C^l_{i,b} \neq C^l_{i',b}$.

Proposition 2 places the feedback of GeoF in a moment hierarchy: first-order updates, including the controller and mean-based metric estimation (Wang et al., 2026), cannot register variation that cancels in the mean, the second-order target registers it for almost every configuration, and the mean-only and diagonal targets sit strictly below it. The proof is in Appendix D.

3.4 Task-Guided Geometry Evolution

The residual target captures local variation but does not identify which differences matter for prediction. Rather than directly estimating local metrics from feature summaries (Wang et al., 2026), we learn a complementary correction through task supervision. This correction, the residual target, and the previous geometric state are combined through a bounded log-triangular update.

The geometric correction is predicted from the current local features, aligned neighborhood aggregate, and structural signature. For node i and block b, these quantities are concatenated into $q^l_{i,b} = [(\xi^l_i)_b \| (\bar{\xi}^l_i)_b \| u_i]$. A shared MLP Γ_θ processes this input using a single hidden layer: $\Gamma_\theta(q^l_{i,b}) = W_{\Gamma,2}\,\mathrm{SiLU}\left(W_{\Gamma,1} q^l_{i,b} + b_{\Gamma,1}\right) + b_{\Gamma,2}$, where θ collects trainable parameters and $\mathrm{SiLU}(x) = x/(1 + e^{-x})$. The output is reshaped into $U^l_{i,b}$, $V^l_{i,b}$ and a log-diagonal correction $d^l_{i,b}$. Following Sec. 3.1, the strictly lower-triangular entries of $U^l_{i,b}(V^l_{i,b})^\top$ correct $a^l_{i,b}$ and $d^l_{i,b}$ corrects $\ell^l_{i,b}$. Concatenating them gives

$$\Delta z^l_{i,b} = \left[\mathrm{Svec}_{\mathrm{sl}}\left(U^l_{i,b}(V^l_{i,b})^\top\right)^\top \quad (d^l_{i,b})^\top\right]^\top \in \mathbb{R}^{m(m+1)/2}. \tag{11}$$

The controller is trained using task objectives, with parameters shared across nodes, blocks, and recurrent steps. The geometric update combines the previous state, residual target, and correction:

$$z^{l+1}_{i,b} = \Pi_{\mathcal{Z}}\left((1-\lambda) z^l_{i,b} + \lambda \hat{z}^l_{i,b} + \gamma \Delta z^l_{i,b}\right), \tag{12}$$

where $\lambda \in (0, 1]$ and $\gamma \geq 0$. The target supplies a reference derived from local interactions, while the learned correction allows the update to depart from that reference under task supervision.

Proposition 3 (Stability and Convergence) *Fix $\tau > 0$, $\epsilon_s > 0$, $\eta \geq 0$ and a feature bound Ξ. Let $\mathcal{D} = \{(\xi, Z) : \|\xi_i\|_2 \leq \Xi,\ z_{i,b} \in \mathcal{Z}\}$, let $F : (\xi^l, Z^l) \mapsto (\xi^{l+1}, Z^{l+1})$ denote one recurrent step, and let $d_\infty(S, S') = \max_i \left(\|\xi_i - \xi'_i\|_2 + \|Z_i - Z'_i\|_F\right)$.*

(i) Eq. (12) is a projected proximal step: $\Pi_{\mathcal{Z}}$ is the Euclidean projection onto $\mathcal{Z}$, and $z^{l+1}_{i,b}$ is the unique minimizer over $\mathcal{Z}$ of

$$\tfrac{1-\lambda}{2}\|z - z^l_{i,b}\|_2^2 + \tfrac{\lambda}{2}\|z - \hat{z}^l_{i,b}\|_2^2 - \gamma\langle \Delta z^l_{i,b},\, z - z^l_{i,b}\rangle.$$

It differs from the uncorrected update $z^{l+1,0}_{i,b} = \Pi_{\mathcal{Z}}\big((1-\lambda) z^l_{i,b} + \lambda \hat{z}^l_{i,b}\big)$ by at most $\gamma\|\Delta z^l_{i,b}\|_2$.

(ii) One step is Lipschitz on $\mathcal{D}$: $d_\infty\big(F(S), F(S')\big) \leq K\, d_\infty(S, S')$ with $K = c_\xi + (1-\lambda) + \lambda c_{\hat{Z}} + \gamma c_\Delta$, where the constants depend only on m, B, $\sigma_\pm$, κ_L, ϵ_s, η, τ, Ξ, λ, γ, and the norms of W_ϕ, W_r, and Γ_θ, but not on node degrees.

(iii) For fixed ξ, let $\Phi_\xi(Z) = \Pi_{\mathcal{Z}}\big((1-\lambda) Z + \lambda \hat{Z}(Z; \xi) + \gamma \Delta Z(Z; \xi)\big)$. If $Z^\star$ is an unclipped fixed point of Φ_ξ whose Jacobian $J = \partial \Phi_\xi(Z^\star)$ has spectral radius $\rho(J) < 1$, then $Z^\star$ is locally attracting with rate $\rho(J)$: for every $\varepsilon > 0$ there are $c > 0$ and a neighborhood $\mathcal{U}$ of $Z^\star$ such that $\|Z^l - Z^\star\|_F \leq c\,(\rho(J) + \varepsilon)^l\, \|Z^0 - Z^\star\|_F$ for all $Z^0 \in \mathcal{U}$.

(iv) If moreover $\hat{Z}(\cdot; \xi)$ and $\Delta Z(\cdot; \xi)$ are $L_{\hat{z}}$- and L_Δ-Lipschitz on $\mathcal{Z}^{nB}$ and $\rho := (1-\lambda) + \lambda L_{\hat{z}} + \gamma L_\Delta < 1$, the fixed point is unique and $\|Z^{l+1} - Z^l\|_F \leq \rho^l \|Z^1 - Z^0\|_F$ from every initialization.

Proposition 3 characterizes the update as a projected proximal step, bounds the single-step effect of the learned correction, gives a one-step Lipschitz constant that does not depend on node degrees, and states conditions under which the geometry converges. The proof is in Appendix E.

Table 1: Performance comparisons (in %) between baselines and GeoF for Node Classification (NC) and Link Prediction (LP) on different datasets. **Bold** indicates the best performance.

Model	CiteSeer		PubMed		CS		Physics		Photo		Computers	
	NC	LP	NC	LP	NC	LP	NC	LP	NC	LP	NC	LP
GCN	$71.5_{\pm 1.7}$	$92.3_{\pm 0.9}$	$87.6_{\pm 0.5}$	$92.9_{\pm 0.6}$	$93.8_{\pm 0.4}$	$92.7_{\pm 0.6}$	$93.5_{\pm 0.2}$	$92.6_{\pm 1.3}$	$92.1_{\pm 0.5}$	$86.1_{\pm 0.7}$	$87.8_{\pm 0.7}$	$86.9_{\pm 0.8}$
GIN	$70.6_{\pm 1.2}$	$93.0_{\pm 1.0}$	$86.6_{\pm 0.6}$	$89.5_{\pm 0.7}$	$91.3_{\pm 0.6}$	$93.3_{\pm 0.7}$	$94.2_{\pm 0.5}$	$92.1_{\pm 0.8}$	$92.7_{\pm 0.5}$	$87.7_{\pm 0.4}$	$87.1_{\pm 0.8}$	$84.1_{\pm 1.2}$
ML^2-GCL	$73.7_{\pm 2.0}$	$93.8_{\pm 1.0}$	$87.9_{\pm 0.5}$	$95.8_{\pm 0.3}$	$92.1_{\pm 0.5}$	$97.1_{\pm 0.1}$	$93.4_{\pm 1.6}$	$97.2_{\pm 0.1}$	$92.8_{\pm 0.8}$	$96.7_{\pm 0.1}$	$87.4_{\pm 0.8}$	$95.8_{\pm 0.3}$
AMPs	$74.4_{\pm 1.8}$	$93.5_{\pm 0.9}$	$88.9_{\pm 0.3}$	$96.6_{\pm 0.3}$	$94.6_{\pm 0.3}$	$97.1_{\pm 0.2}$	$96.1_{\pm 0.1}$	$96.6_{\pm 0.6}$	$94.6_{\pm 0.8}$	$96.8_{\pm 0.2}$	$90.2_{\pm 0.6}$	$96.0_{\pm 0.1}$
WaveGC	$75.4_{\pm 1.9}$	$92.8_{\pm 0.7}$	$87.6_{\pm 0.5}$	$97.5_{\pm 0.3}$	$94.4_{\pm 0.3}$	$97.1_{\pm 0.2}$	$96.2_{\pm 0.1}$	$97.2_{\pm 0.2}$	$94.6_{\pm 0.8}$	$97.7_{\pm 0.4}$	$90.2_{\pm 0.8}$	$97.5_{\pm 0.4}$
SPARROW	$73.1_{\pm 2.1}$	$92.9_{\pm 0.9}$	$85.5_{\pm 0.6}$	$97.0_{\pm 0.7}$	$92.0_{\pm 0.3}$	$97.0_{\pm 0.1}$	$94.5_{\pm 0.3}$	$97.2_{\pm 0.3}$	$93.4_{\pm 1.0}$	$97.8_{\pm 0.1}$	$88.2_{\pm 0.5}$	$97.3_{\pm 0.2}$
G^2Former	$72.7_{\pm 1.7}$	$93.0_{\pm 0.6}$	$88.6_{\pm 0.5}$	$97.1_{\pm 0.2}$	$93.7_{\pm 0.4}$	$96.4_{\pm 0.9}$	$95.5_{\pm 0.2}$	$97.4_{\pm 0.4}$	$94.1_{\pm 0.7}$	$97.9_{\pm 0.3}$	$90.4_{\pm 0.4}$	$97.1_{\pm 0.1}$
HGCN	$74.6_{\pm 1.8}$	$92.1_{\pm 1.0}$	$84.8_{\pm 0.7}$	$91.3_{\pm 0.3}$	$91.9_{\pm 0.4}$	$93.2_{\pm 0.4}$	$92.0_{\pm 3.1}$	$95.7_{\pm 0.4}$	$89.5_{\pm 0.8}$	$92.6_{\pm 0.9}$	$85.9_{\pm 0.7}$	$88.1_{\pm 0.2}$
D-GCN	$74.9_{\pm 1.1}$	$92.0_{\pm 0.6}$	$86.9_{\pm 0.9}$	$93.1_{\pm 0.4}$	$93.8_{\pm 0.2}$	$96.2_{\pm 0.1}$	$93.6_{\pm 0.2}$	$97.1_{\pm 0.3}$	$93.7_{\pm 0.6}$	$97.1_{\pm 0.1}$	$87.1_{\pm 0.5}$	$94.1_{\pm 0.6}$
SPDGNN	$74.9_{\pm 1.7}$	$92.7_{\pm 0.8}$	$87.2_{\pm 0.6}$	$93.8_{\pm 0.7}$	$92.6_{\pm 0.4}$	$94.9_{\pm 0.2}$	$94.1_{\pm 0.2}$	$94.5_{\pm 0.6}$	$93.0_{\pm 0.8}$	$95.7_{\pm 0.5}$	$88.7_{\pm 1.2}$	$93.1_{\pm 0.6}$
ACE-HGNN	$74.0_{\pm 1.5}$	$92.9_{\pm 0.6}$	$86.8_{\pm 0.5}$	$96.9_{\pm 0.1}$	$92.5_{\pm 0.3}$	$97.0_{\pm 0.1}$	$93.9_{\pm 0.2}$	$97.3_{\pm 0.2}$	$94.0_{\pm 1.0}$	$97.2_{\pm 0.1}$	$89.5_{\pm 1.1}$	$97.3_{\pm 0.5}$
BEC-GNN	$74.0_{\pm 1.6}$	$94.0_{\pm 1.3}$	$87.3_{\pm 0.3}$	$97.4_{\pm 0.2}$	$93.3_{\pm 0.4}$	$97.2_{\pm 0.2}$	$96.3_{\pm 0.2}$	$97.2_{\pm 0.2}$	$94.1_{\pm 0.5}$	$97.1_{\pm 0.1}$	$89.8_{\pm 0.8}$	$96.7_{\pm 0.1}$
GNRF	$74.2_{\pm 1.7}$	$93.6_{\pm 0.5}$	$87.4_{\pm 0.6}$	$95.8_{\pm 0.2}$	$93.7_{\pm 0.6}$	$97.0_{\pm 0.1}$	$95.4_{\pm 1.0}$	$97.3_{\pm 0.3}$	$93.9_{\pm 0.9}$	$96.8_{\pm 0.4}$	$89.0_{\pm 0.9}$	$96.3_{\pm 0.3}$
ARGNN	$75.6_{\pm 1.2}$	$94.1_{\pm 0.6}$	$88.8_{\pm 0.2}$	$97.6_{\pm 0.4}$	$94.4_{\pm 0.4}$	$96.2_{\pm 0.3}$	$96.3_{\pm 0.2}$	$97.4_{\pm 0.2}$	$94.9_{\pm 0.5}$	$97.0_{\pm 0.3}$	$90.4_{\pm 0.4}$	$97.4_{\pm 0.5}$
GeoF	**77.7**$_{\pm 1.2}$	**94.9**$_{\pm 0.8}$	**89.6**$_{\pm 0.3}$	**98.5**$_{\pm 0.2}$	**95.8**$_{\pm 0.3}$	**97.7**$_{\pm 0.3}$	**97.2**$_{\pm 0.1}$	**97.9**$_{\pm 0.3}$	**96.0**$_{\pm 0.4}$	**98.4**$_{\pm 0.2}$	**91.2**$_{\pm 0.5}$	**98.0**$_{\pm 0.4}$

3.5 Learning Objectives

The prediction losses depend on the final geometric coordinates $\{z_{i,b}^L\}_{b=1}^B$ through the environment representation $h_i^L = L_i^L \xi_i^L$, where L_i^L is assembled from the block factors reconstructed using Eq. (1). The intermediate updates $z_{i,b}^l \mapsto z_{i,b}^{l+1}$ also shape subsequent message passing by determining the neighborhood weights and frame transformations. The task objectives below train the geometric controller Γ_θ, propagation parameters, and task-specific readouts jointly through these dependencies.

Node Classification. A shared linear classifier maps h_i^L to class logits, followed by softmax normalization. Training minimizes cross-entropy over the labeled nodes $V_{\text{tr}} \subseteq V$:

$$s_i = W_o h_i^L + b_o, \quad p_i = \text{softmax}(s_i), \quad \mathcal{L}_{\text{node}} = -\frac{1}{|V_{\text{tr}}|} \sum_{i \in V_{\text{tr}}} \sum_{c=1}^{C} \mathbb{I}[y_i = c] \log p_{i,c}, \tag{13}$$

where W_o, b_o are learnable classifier parameters, C is the number of node classes. s_i, p_i denote the logits and class probabilities. y_i is the ground-truth, and $\mathbb{I}[y_i = c] = 1$ when $y_i = c$ and 0 otherwise.

Graph Classification. For a graph G with node set V_G, we mean-pool the final environment representations in their shared coordinate space and apply a shared MLP to obtain class probabilities:

$$h_G = \frac{1}{|V_G|} \sum\nolimits_{i \in V_G} h_i^L, \; p_G = \text{softmax}(\text{MLP}(h_G)), \; \mathcal{L}_{\text{graph}} = \mathbb{E}_{G \sim \mathcal{D}_{\text{tr}}^{\text{graph}}} \left[\ell_{\text{CE}}(p_G, Y_G) \right], \tag{14}$$

where h_G is the graph representation. The vector p_G contains the class probabilities, with $p_{G,c}$ denoting the probability of class c, and Y_G is the ground-truth label. The cross-entropy loss is $\ell_{\text{CE}}(p_G, Y_G) = -\sum_{c=1}^{C} \mathbb{I}[Y_G = c] \log p_{G,c}$, where $\mathbb{I}[Y_G = c] = 1$ when $Y_G = c$ and 0 otherwise.

Link Prediction. For a candidate pair (u, v), we construct a symmetric pair representation $f_{uv} = [h_u^L \odot h_v^L \| \, |h_u^L - h_v^L|] \in \mathbb{R}^{2d}$ from the final environment representations. A shared MLP predicts whether the nodes are connected. Let $\mathcal{E}_{\text{tr}} = E^+ \cup E^-$ contain positive training edges E^+ and sampled negative pairs E^-. The edge probability and training objective are

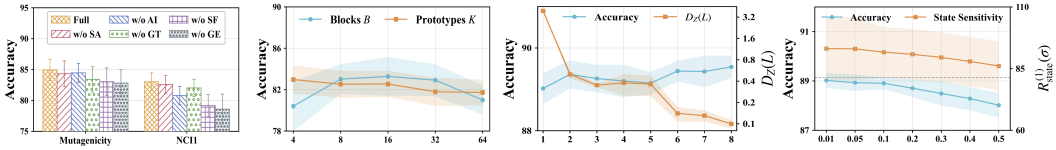
$$p_{uv} = \sigma(\text{MLP}_{\text{link}}(f_{uv})), \qquad \mathcal{L}_{\text{link}} = \mathbb{E}_{(u,v) \sim \mathcal{E}_{\text{tr}}} \left[\ell_{\text{BCE}}(p_{uv}, y_{uv}) \right], \tag{15}$$

where MLP_{link} is a learnable decoder, $\sigma(t)$ is the sigmoid function, and p_{uv} is the predicted edge probability. The label $y_{uv} = 1$ for positive edges and 0 for negative pairs.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate GeoF on node classification, link prediction, and graph classification. Node classification uses 60%/20%/20% splits for training, validation, and testing, and link prediction uses 80%/5%/15% (Wang et al., 2026; Pei et al., 2020), both on CiteSeer, PubMed (Sen et al., 2008),



(a) Ablation Study (b) Sensitivity Analysis (c) Evolution Depth (d) Perturbation Stability

Figure 1: (a) Ablations on Mutagenicity and NCI1. (b) Sensitivity to block count B and atlas size K on NCI1. (c), (d) Evolution depth and perturbation stability on PubMed, respectively.

CS, Physics, Photo, and Computers (Shchur et al., 2018). Graph classification uses 10-fold cross-validation on PROTEINS, Mutagenicity, NCI1, FRANKENSTEIN, BBBP, and ogbg-molhiv (Hong et al., 2024; Wei et al., 2023). We report accuracy for node and graph classification, and ROC-AUC for link prediction and for BBBP and ogbg-molhiv. More details are provided in Appendix G.

Baselines. We compare GeoF with a comprehensive set of baselines, including: (1) general graph neural networks (GNNs), such as GCN (Kipf & Welling, 2017), GIN (Xu et al., 2019), ML^2-GCL (Liang et al., 2025), AMPs (Errica et al., 2025), WaveGC (Liu et al., 2025), SPARROW (Lin et al., 2025), and G^{2}Former (Zhang et al., 2025); (2) manifold-based GNNs, including HGCN (Chami et al., 2019), D-GCN (Sun et al., 2024), and SPDGNN (Wang & Chang, 2025); and (3) adaptive GNNs, including ACE-HGNN (Fu et al., 2021), GNRF (Chen et al., 2025), BEC-GNN (Hevapathige et al., 2025), and ARGNN (Wang et al., 2026). More details of baselines can be found in Appendix H.

4.2 Performance Comparisons

Tables 1 and 2 compare GeoF with the baselines on node classification (NC), link prediction (LP), and graph classification (GC). Three observations follow. (1) Advanced architectures such as AMPs and WaveGC improve clearly over GCN and GIN, which underlines the value of multi-scale propagation. (2) Manifold-based and adaptive GNNs are competitive through richer geometric priors and curvature-aware designs, but they treat geometry as a fixed manifold or a locally adaptive parameter rather than an evolving state, which limits how far the propagation space can follow task-specific requirements. (3) GeoF is best on every dataset and task. This matches its design. It keeps propagation geometry as a persistent node-wise state that governs neighborhood weighting and frame-aligned propagation, and it updates this state at every step from the previous geometry, a target built from second-order statistics of aligned residuals, and task-supervised corrections, so directional variation and within-block dependencies enter the geometry that governs subsequent propagation.

Table 2: Performance comparisons (in %) between baselines and GeoF for Graph Classification.

Model	PROTEINS	Mutag	NCI1	FRANK	BBBP	molhiv
GCN	$75.3_{\pm 1.9}$	$79.8_{\pm 1.8}$	$76.0_{\pm 1.0}$	$63.3_{\pm 2.2}$	$87.4_{\pm 2.0}$	$75.8_{\pm 2.0}$
GIN	$76.7_{\pm 1.7}$	$80.1_{\pm 1.9}$	$78.0_{\pm 1.2}$	$68.9_{\pm 1.7}$	$89.5_{\pm 2.1}$	$77.3_{\pm 2.0}$
ML^2-GCL	$77.9_{\pm 1.8}$	$81.9_{\pm 1.9}$	$80.7_{\pm 1.3}$	$70.7_{\pm 1.8}$	$90.9_{\pm 2.2}$	$81.3_{\pm 1.6}$
AMPs	$78.8_{\pm 2.0}$	$83.4_{\pm 1.6}$	$82.0_{\pm 1.1}$	$72.4_{\pm 1.8}$	$92.5_{\pm 2.3}$	$82.0_{\pm 2.4}$
WaveGC	$79.0_{\pm 1.8}$	$82.3_{\pm 2.3}$	$81.6_{\pm 1.8}$	$72.6_{\pm 1.9}$	$92.6_{\pm 1.8}$	$82.0_{\pm 1.9}$
SPARROW	$78.7_{\pm 1.5}$	$83.7_{\pm 1.9}$	$80.9_{\pm 1.2}$	$72.3_{\pm 2.2}$	$91.8_{\pm 1.9}$	$81.5_{\pm 2.3}$
G^{2}Former	$78.9_{\pm 2.0}$	$82.7_{\pm 2.4}$	$82.1_{\pm 1.5}$	$71.9_{\pm 1.9}$	$92.4_{\pm 2.4}$	$81.8_{\pm 1.6}$
HGCN	$77.3_{\pm 1.6}$	$80.2_{\pm 2.1}$	$78.3_{\pm 1.4}$	$69.4_{\pm 2.3}$	$89.4_{\pm 2.2}$	$76.0_{\pm 1.6}$
D-GCN	$78.4_{\pm 2.2}$	$82.2_{\pm 1.8}$	$79.1_{\pm 1.3}$	$69.8_{\pm 2.2}$	$90.7_{\pm 1.7}$	$79.0_{\pm 2.2}$
SPDGNN	$78.5_{\pm 1.9}$	$81.6_{\pm 1.7}$	$79.6_{\pm 1.6}$	$71.4_{\pm 2.1}$	$91.5_{\pm 1.7}$	$81.3_{\pm 2.2}$
ACE-HGNN	$77.5_{\pm 1.7}$	$80.7_{\pm 1.8}$	$81.6_{\pm 1.2}$	$71.7_{\pm 2.2}$	$91.6_{\pm 1.2}$	$79.1_{\pm 2.2}$
BEC-GNN	$77.5_{\pm 1.9}$	$81.0_{\pm 1.5}$	$79.4_{\pm 2.2}$	$70.0_{\pm 2.3}$	$91.9_{\pm 1.8}$	$80.3_{\pm 1.3}$
GNRF	$78.5_{\pm 1.7}$	$82.4_{\pm 2.3}$	$81.9_{\pm 2.1}$	$72.1_{\pm 1.7}$	$92.2_{\pm 1.8}$	$81.5_{\pm 1.7}$
ARGNN	$78.0_{\pm 1.8}$	$83.4_{\pm 1.5}$	$81.7_{\pm 1.6}$	$72.2_{\pm 2.0}$	$92.3_{\pm 1.7}$	$81.1_{\pm 2.2}$
GeoF	**79.4**$_{\pm 1.6}$	**84.9**$_{\pm 1.7}$	**83.0**$_{\pm 1.5}$	**73.3**$_{\pm 1.8}$	**93.4**$_{\pm 1.8}$	**82.9**$_{\pm 1.1}$

4.3 Ablation Study

We evaluate five ablations on Mutagenicity and NCI1. Three remove a component: the structural signatures in the prototype gate and the correction (w/o SA), the structure-aware prototype atlas for initialization (w/o AI), and the triangular frame transport into the target frame (w/o GT). Two alter the geometry update, either replacing the second-order residual target by the current geometry while keeping the correction (w/o SF) or freezing the initialized geometry across steps (w/o GE). Figure 1(a) shows that the full model attains the highest accuracy on both datasets. Freezing the geometry costs the most, confirming the benefit of adapting propagation geometry during message passing, and removing second-order feedback costs the second most, so residual statistics contribute beyond the correction. Disabling frame transport also lowers accuracy, consistent with the role of coordinate alignment in aggregation and residual computation. Removing the atlas hurts more on NCI1, and removing structural signatures causes modest decreases. Additional results are in Appendix K.2.

4.4 Sensitivity Analysis

Hyperparameters Analysis. We vary the number of geometric blocks B, atlas prototypes K, and evolution depth L, which set the block partition, atlas size, and recurrent depth. On NCI1, Figure 1(b) shows that accuracy improves as B increases from 4 to 8, stays stable for $B \in \{8, 16, 32\}$, and drops at $B = 64$, while it changes little for $K \in \{4, 8, 16\}$ and decreases slightly at $K = 32$ and 64; moderate values therefore suffice. On PubMed, Figure 1(c) shows stable accuracy for L from 1 to 8, while the terminal geometric change $D_Z(L) = \frac{1}{nB} \sum_{i,b} \|z_{i,b}^L - z_{i,b}^{L-1}\|_2$ decreases with depth, with a plateau between $L = 2$ and 5. Successive geometric updates thus shrink with depth without affecting accuracy, consistent with the stability of Proposition 3. More results are provided in Appendix K.3.

Perturbation Stability. We perturb the initial features $\{\xi_i^0\}_{i=1}^n$ on PubMed along a random Gaussian direction with relative norm σ, keeping the graph, initial geometry, and trained parameters fixed, and report accuracy together with the one-step sensitivity $R_{\text{state}}^{(1)}(\sigma) = d(S_\sigma^1, S^1)/d(S_\sigma^0, S^0)$, where $S^l = (\Xi^l, Z^l)$ stacks the feature states and geometric coordinates of all nodes and $d(S, S') = \|\Xi - \Xi'\|_F + \sum_b \|Z_b - Z_b'\|_F$. Figure 1(d) shows that $R_{\text{state}}^{(1)}(\sigma)$ is nearly constant at small σ, indicating a proportional response to small perturbations, and decreases steadily by 7.6% from $\sigma = 0.01$ to 0.5, while accuracy drops by only 1.1% over the same range. Larger perturbations thus reduce accuracy mildly without amplifying the one-step sensitivity of the coupled update.

4.5 Analysis of Second-Order Feedback

Features versus Feedback. To test whether the gains come from second-order information alone, we give GCN, GIN, and ARGNN the same weighted second-order residual statistics as additional features, with parameter counts matched to GeoF within 0.5% (Table 3). The augmentation helps GIN in all four settings and GCN in three, but hurts GCN on BBBP and ARGNN on CiteSeer and BBBP. GeoF exceeds the strongest augmented variant in every column, by 0.7 to 3.3 points. At comparable parameter budgets, the difference is how the statistics are used. The baselines consume them as features, whereas GeoF uses them as feedback that changes the geometry governing subsequent weighting and transport.

Table 3: Performance comparison (in %) between baselines and their second-order (SO) variants. **Bold** indicates the best performance.

Model	CiteSeer	Photo	PROTEINS	BBBP
GCN	$71.5_{\pm 1.7}$	$92.1_{\pm 0.5}$	$75.3_{\pm 1.9}$	$87.4_{\pm 2.0}$
GCN w/ SO	$72.3_{\pm 1.9}$	$92.4_{\pm 1.1}$	$77.0_{\pm 2.4}$	$86.7_{\pm 2.5}$
GIN	$70.6_{\pm 1.2}$	$92.7_{\pm 0.5}$	$76.7_{\pm 1.7}$	$89.5_{\pm 2.1}$
GIN w/ SO	$73.3_{\pm 1.0}$	$93.6_{\pm 1.4}$	$78.5_{\pm 1.5}$	$91.3_{\pm 2.2}$
ARGNN	$75.6_{\pm 1.2}$	$94.9_{\pm 0.5}$	$78.0_{\pm 1.8}$	$92.3_{\pm 1.7}$
ARGNN w/ SO	$74.4_{\pm 2.3}$	$95.3_{\pm 0.6}$	$78.3_{\pm 2.2}$	$91.9_{\pm 2.1}$
GeoF	$\mathbf{77.7}_{\pm 1.2}$	$\mathbf{96.0}_{\pm 0.4}$	$\mathbf{79.4}_{\pm 1.6}$	$\mathbf{93.4}_{\pm 1.8}$

Feedback Structure and Schedule. Table 4 varies the feedback with the rest of the model unchanged. Replacing the full statistic $C_{i,b}^l$ by the mean-only target $C_{i,b}^{l,\text{mean}}$ or the diagonal target $C_{i,b}^{l,\text{diag}}$ of Proposition 2 lowers accuracy on all four datasets, consistent with Proposition 2; the diagonal target loses most on NCI1 and Mutag. The two restricted variants rank differently across datasets, so the mean and the marginal second moments capture different aspects of local variation. Updating the geometry only once, after the first step, also lowers accuracy on all four datasets, by 1.4 and 1.0 points on Photo and CS and by 1.8 and 3.2 points on Mutag and NCI1.

Table 4: Performance comparison (in %) among different residual feedback statistics. **Bold** indicates the best performance.

Feedback	Photo	CS	Mutag.	NCI1
Mean-only	$94.9_{\pm 0.6}$	$95.1_{\pm 0.3}$	$83.7_{\pm 1.7}$	$81.9_{\pm 1.6}$
Diagonal	$95.1_{\pm 0.3}$	$95.2_{\pm 0.4}$	$83.2_{\pm 2.0}$	$80.4_{\pm 1.0}$
One-shot	$94.6_{\pm 0.5}$	$94.8_{\pm 0.3}$	$83.1_{\pm 1.8}$	$79.8_{\pm 2.7}$
Full	$\mathbf{96.0}_{\pm 0.4}$	$\mathbf{95.8}_{\pm 0.3}$	$\mathbf{84.9}_{\pm 1.7}$	$\mathbf{83.0}_{\pm 1.5}$

5 Conclusion

To capture message-level variation obscured by aggregation, we proposed GeoF, which jointly evolves node features and propagation geometry through recurrent message-passing feedback. The geometry governs neighborhood weighting and triangular frame transport for consistent message comparison, second-order residual statistics capture directional variation and within-block dependencies to define geometric targets, and bounded log-triangular updates combine these targets with task-supervised corrections while preserving positive definiteness. Experiments on node, link, and graph prediction show consistent gains over competitive baselines, supporting learning propagation geometry from the interactions it induces and motivating future work on adaptive-depth inference and evolving graphs.

AI USE STATEMENT

Large language models, including ChatGPT, were only used during manuscript revision for language polishing. The authors reviewed and revised the AI-assisted content and take full responsibility for the research ideas, experimental design, data analysis, and scientific conclusions.

REFERENCES

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *Proceedings of the International Conference on Machine Learning*, pp. 21–29, 2019.

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

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Proceedings of the Conference on Neural Information Processing Systems*, 32, 2019.

Jialong Chen, Bowen Deng, Zhen Wang, Chuan Chen, and Zibin Zheng. Graph neural ricci flow: Evolving feature from a curvature perspective. In *Proceedings of the International Conference on Learning Representations*, volume 2025, pp. 31083–31111, 2025.

Yu Chen, Lingfei Wu, and Mohammed Zaki. Iterative deep graph learning for graph neural networks: Better and robust node embeddings. *Advances in neural information processing systems*, 33: 19314–19326, 2020.

Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

Federico Errica, Henrik Christiansen, Viktor Zaverkin, Takashi Maruyama, Mathias Niepert, and Francesco Alesiani. Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching. In *Proceedings of the International Conference on Machine Learning*, pp. 15490–15515. PMLR, 2025.

Xingcheng Fu, Jianxin Li, Jia Wu, Qingyun Sun, Cheng Ji, Senzhang Wang, Jiajun Tan, Hao Peng, and Philip S Yu. Ace-hgnn: Adaptive curvature exploration hyperbolic graph neural network. In *2021 IEEE international conference on data mining (ICDM)*, pp. 111–120. IEEE, 2021.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning*, pp. 1263–1272, 2017.

Asela Hevapathige, Ahad N Zehmakan, and Qing Wang. Depth-adaptive graph neural networks via learnable bakry-émery curvature. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 944–955, 2025.

Xiaobin Hong, Wenzhong Li, Chaoqun Wang, Mingkai Lin, and Sanglu Lu. Label attentive distillation for gnn-based graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8499–8507, 2024.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

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. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1):312–320, 2005.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*, 2017.

See Hian Lee, Feng Ji, and Wee Peng Tay. Node-specific space selection via localized geometric hyperbolicity in graph neural networks. *arXiv preprint arXiv:2303.01724*, 2023.

Jianqing Liang, Zhiqiang Li, Xinkai Wei, Yuan Liu, and Zhiqiang Wang. Ml ^{2}-gcl: Manifold learning inspired lightweight graph contrastive learning. In *Proceedings of the International Conference on Machine Learning*, 2025.

Yuena Lin, Gengyu Lyu, Haichun Cai, Deng-Bao Wang, Haobo Wang, and Zhen Yang. Simplified graph contrastive learning model without augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 338–348, 2020.

Nian Liu, Xiaoxin He, Thomas Laurent, Francesco Di Giovanni, Michael M Bronstein, and Xavier Bresson. A general graph spectral wavelet convolution via chebyshev order decomposition. In *Proceedings of the International Conference on Machine Learning*, pp. 38598–38622. PMLR, 2025.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 43–52, 2015.

Francesco Orsini, Paolo Frasconi, and Luc De Raedt. Graph invariant kernels. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2015.

James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Li Sun, Zhenhao Huang, Zixi Wang, Feiyang Wang, Hao Peng, and Philip S Yu. Motif-aware riemannian graph neural network with generative-contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9044–9052, 2024.

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

Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375, 2008.

Xudong Wang, Chris Ding, Tongxin Li, and Jicong Fan. Adaptive riemannian graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26606–26614, 2026.

Yu Wang and Yi Chang. Enhancing graph neural networks on spd manifolds via cholesky decomposition. *Pattern Recognition*, pp. 112763, 2025.

Lanning Wei, Huan Zhao, Zhiqiang He, and Quanming Yao. Neural architecture search for gnn-based graph classification. *ACM Transactions on Information Systems*, 42(1):1–29, 2023.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Proceedings of the Conference on Neural Information Processing Systems*, 35:27387–27401, 2022.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the International Conference on Machine Learning*, pp. 5453–5462. pmlr, 2018.

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

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Proceedings of the Conference on Neural Information Processing Systems*, 32, 2019.

Jingyuan Zhang, Xin Wang, Lei Yu, Zhirong Huang, Li Yang, and Fengjun Zhang. Restricted global-aware graph filters bridging gnns and transformer for node classification. In *Proceedings of the Conference on Neural Information Processing Systems*, 2025.

A Notation Summary

As shown in the Table 5, we summarize the key notations of this paper.

Table 5: Summary of key notations.

Symbol	Description
$G = (V, E, X), A$	Attributed graph with node set V, edge set E, feature matrix X, and adjacency matrix A.
n, F_0, C	Numbers of nodes, input features, and classes, respectively.
u_i, U, q	Fixed structural signature of node i, signature matrix, and signature dimension.
$\mathcal{N}(i), \widetilde{\mathcal{N}}(i)$	Neighborhood of node i and its extension with a self-loop.
d, B, m	Hidden dimension, number of geometric blocks, and block size, with $d = Bm$.
K, L	Numbers of geometric prototypes and recurrent evolution steps.
ξ_i^l, Z_i^l	Local feature state and node-wise geometric coordinates at step l.
$z_{i,b}^l = [a_{i,b}^l, \ell_{i,b}^l]^\top$	Log-triangular coordinates of block b: strictly lower-triangular entries and log-diagonal scales.
$L_{i,b}^l, g_{i,b}^l$	Block triangular frame and corresponding symmetric positive-definite geometry.
L_i^l, g_i^l	Node-wise frame and local geometry assembled as block-diagonal matrices.
$\mathcal{Z}, \Pi_{\mathcal{Z}}$	Admissible coordinate domain with bounds $a_{\max}, \ell_{\min}, \ell_{\max}$, and coordinatewise clipping onto it.
$\sigma_+, \sigma_-, \kappa_L$	Spectral bounds of admissible frames and Lipschitz constant of the frame reconstruction.
$z_{k,b}^{\text{proto}}, \alpha_{ik}$	Coordinates of prototype k in block b and its structure-conditioned weight for node i.
h_i^l	Shared-environment representation $h_i^l = L_i^l \xi_i^l$.
$e_{ij}^l, \omega_{ij}^l, \tau$	Squared environment-space distance, normalized neighborhood weight, and temperature.
ϕ, W_ϕ	Shared linear message transformation and its learnable weight matrix.
$T_{j\to i,b}^l$	Blockwise source-to-target frame map $(L_{i,b}^l)^{-1} L_{j,b}^l$.
$m_{j\to i,b}^l, m_{j\to i}^l$	Transported block message and the full message concatenated across blocks.
$\bar{\xi}_i^l, r_i^l$	Aligned neighborhood aggregate and elementwise gate for feature updates.
$\delta_{ij,b}^l, \bar{\delta}_{i,b}^l$	Residual between an aligned source message and the transformed target state, and its weighted mean.
$C_{i,b}^l, \epsilon_s$	Regularized weighted second-moment matrix of residuals and its spectral regularization.
$R_{i,b}^l, \widehat{z}_{i,b}^l, \eta$	Stabilized Cholesky target frame, its log-triangular coordinates, and the numerical jitter.
$q_{i,b}^l, \Gamma_\theta$	Controller input and shared controller predicting task-supervised geometric corrections.
$U_{i,b}^l, V_{i,b}^l, d_{i,b}^l$	Controller factors for lower-triangular corrections and the log-diagonal correction vector.
$\Delta z_{i,b}^l$	Learned correction to the block log-triangular coordinates.
λ, γ	Mixing weight for the residual target and strength of the learned correction.

B Proof of Lemma 1

Lemma 1 (Well-Posed Geometric Parameterization) *Let $m \geq 1$ and $z, z' \in \mathcal{Z}$. The frame $L(z)$ is lower triangular with $\det L(z) = \exp(\sum_t \ell_t) > 0$, so it is invertible and $g(z) \in \mathrm{SPD}(m)$. There are constants $\sigma_+ \geq \sigma_- > 0$ and $\kappa_L > 0$, depending only on m, $a_{\max}$, $\ell_{\min}$, and $\ell_{\max}$, such that*

$$\|L(z)\|_2 \leq \sigma_+, \qquad \|L(z)^{-1}\|_2 \leq \sigma_-^{-1}, \qquad \sigma_-^2 \, I_m \preceq g(z) \preceq \sigma_+^2 \, I_m,$$

and the reconstruction is Lipschitz: $\|L(z) - L(z')\|_F \leq \kappa_L \|z - z'\|_2$ and $\|L(z)^{-1} - L(z')^{-1}\|_2 \leq \sigma_-^{-2} \kappa_L \|z - z'\|_2$. All statements carry over to the block-diagonal frames $L_i = \bigoplus_b L(z_{i,b})$ and $g_i = L_i^\top L_i$, with $\|z - z'\|_2$ replaced by $\|Z_i - Z_i'\|_F := (\sum_b \|z_{i,b} - z_{i,b}'\|_2^2)^{1/2}$.

Proof 1 *Throughout, $\|\cdot\|_2$ denotes the Euclidean norm of a vector and the spectral norm of a matrix, $\|\cdot\|_F$ the Frobenius norm, and $\|\cdot\|_1$, $\|\cdot\|_\infty$ the induced max-column-sum and max-row-sum norms. Recall from Eq.* (1) *that a block coordinate $z = [a^\top, \ell^\top]^\top \in \mathcal{Z}$, with $a \in \mathbb{R}^{m(m-1)/2}$ and $\ell \in \mathbb{R}^m$,*

is reconstructed as

$$L(z) = N + D, \qquad N := \mathrm{mat}_{\mathrm{sl}}(a), \qquad D := \mathrm{Diag}(e^{\ell}), \tag{16}$$

so that $L(z)_{rr} = e^{\ell_r}$ *and* $L(z)_{rs} = a_{rs}$ *for* $r > s$*, where* N *is strictly lower triangular with* $|N_{rs}| \le a_{\max}$ *and* $\ell_t \in [\ell_{\min}, \ell_{\max}]$ *for all* $r > s$ *and* t*. With*

$$\nu := a_{\max}\sqrt{\tfrac{m(m-1)}{2}}, \qquad \bar{a} := a_{\max}e^{-\ell_{\min}}, \qquad \kappa_0 := \min\Big\{\sqrt{m}\,(1+\bar{a})^{m-1},\ \sum_{k=0}^{m-1}\big(\nu e^{-\ell_{\min}}\big)^k\Big\}, \tag{17}$$

we set

$$\sigma_+ := e^{\ell_{\max}} + \nu, \qquad \sigma_- := e^{\ell_{\min}}/\kappa_0, \qquad \kappa_L := 1 + e^{\ell_{\max}}. \tag{18}$$

These constants depend only on m*,* $a_{\max}$*,* $\ell_{\min}$*, and* $\ell_{\max}$*. Since* $\kappa_0 \ge 1$ *and* $e^{\ell_{\max}} \ge e^{\ell_{\min}}$*, we have* $\sigma_+ \ge \sigma_- > 0$*.*

Invertibility. *$L(z)$ is lower triangular with diagonal entries $e^{\ell_t} \in [e^{\ell_{\min}}, e^{\ell_{\max}}]$, so $\det L(z) = \prod_t e^{\ell_t} = \exp(\sum_t \ell_t) > 0$ and $L(z)$ is invertible. The matrix $g(z) = L(z)^\top L(z)$ is symmetric and, for $v \neq 0$, $v^\top g(z) v = \|L(z)v\|_2^2 > 0$; hence $g(z) \in \mathrm{SPD}(m)$.*

Upper bound. *Since $\|D\|_2 = \max_t e^{\ell_t} \le e^{\ell_{\max}}$ and N has $m(m-1)/2$ entries of magnitude at most $a_{\max}$,*

$$\|L(z)\|_2 \le \|D\|_2 + \|N\|_2 \le e^{\ell_{\max}} + \|N\|_F \le e^{\ell_{\max}} + a_{\max}\sqrt{\tfrac{m(m-1)}{2}} = \sigma_+. \tag{19}$$

Lower bound. *Factor the diagonal out of Eq. (16): $L(z) = D(I_m + \tilde{N})$ with $\tilde{N} := D^{-1}N$, so that $L(z)^{-1} = (I_m + \tilde{N})^{-1}D^{-1}$ and $\|D^{-1}\|_2 = \max_t e^{-\ell_t} \le e^{-\ell_{\min}}$. Left multiplication by D^{-1} scales row r by $e^{-\ell_r}$, so $\tilde{N}$ is strictly lower triangular with $|\tilde{N}_{rs}| = e^{-\ell_r}|a_{rs}| \le \bar{a}$ and $\|\tilde{N}\|_F \le e^{-\ell_{\min}}\|N\|_F \le \nu e^{-\ell_{\min}}$. It suffices to show $\|(I_m + \tilde{N})^{-1}\|_2 \le \kappa_0$. We bound this norm in two ways and take the smaller.*

(i) Forward substitution. *For $y \in \mathbb{R}^m$ let $w = (I_m + \tilde{N})^{-1}y$, i.e. $w_r = y_r - \sum_{s<r}\tilde{N}_{rs}w_s$ for $r = 1, \dots, m$. We claim $|w_r| \le \|y\|_\infty(1+\bar{a})^{r-1}$. This holds for $r = 1$ since $w_1 = y_1$. If it holds for all $s < r$, then*

$$|w_r| \le \|y\|_\infty + \bar{a}\sum_{s<r}|w_s| \le \|y\|_\infty\Big(1 + \bar{a}\sum_{s=1}^{r-1}(1+\bar{a})^{s-1}\Big) = \|y\|_\infty(1+\bar{a})^{r-1},$$

where the last equality uses the geometric sum $\bar{a}\sum_{s=1}^{r-1}(1+\bar{a})^{s-1} = (1+\bar{a})^{r-1} - 1$. Hence $\|(I_m+\tilde{N})^{-1}\|_\infty \le (1+\bar{a})^{m-1}$. For any $A \in \mathbb{R}^{m\times m}$, $\|A\|_1 \le m\|A\|_\infty$ and $\|A\|_2 \le \sqrt{\|A\|_1\|A\|_\infty}$, so $\|(I_m + \tilde{N})^{-1}\|_2 \le \sqrt{m}\,(1+\bar{a})^{m-1}$.

(ii) Neumann series. *Since $\tilde{N}$ is strictly lower triangular, $\tilde{N}^m = 0$ and $(I_m+\tilde{N})^{-1} = \sum_{k=0}^{m-1}(-\tilde{N})^k$. With $\|\tilde{N}\|_2 \le \|\tilde{N}\|_F \le \nu e^{-\ell_{\min}}$,*

$$\|(I_m + \tilde{N})^{-1}\|_2 \le \sum_{k=0}^{m-1}\|\tilde{N}\|_2^k \le \sum_{k=0}^{m-1}\big(\nu e^{-\ell_{\min}}\big)^k.$$

Combining (i) and (ii) gives $\|(I_m + \tilde{N})^{-1}\|_2 \le \kappa_0$, hence $\|L(z)^{-1}\|_2 \le \|(I_m + \tilde{N})^{-1}\|_2\|D^{-1}\|_2 \le \kappa_0 e^{-\ell_{\min}} = \sigma_-^{-1}$.

Metric bounds. *The eigenvalues of $g(z) = L(z)^\top L(z)$ are the squared singular values of $L(z)$. Since $\sigma_{\max}(L(z)) = \|L(z)\|_2 \le \sigma_+$ and $\sigma_{\min}(L(z)) = \|L(z)^{-1}\|_2^{-1} \ge \sigma_-$, we obtain $\sigma_-^2 I_m \preceq g(z) \preceq \sigma_+^2 I_m$.*

Lipschitz reconstruction. *Write $z' = [a'^\top, \ell'^\top]^\top$, $D' = \mathrm{Diag}(e^{\ell'})$ and $N' = \mathrm{mat}_{\mathrm{sl}}(a')$. By Eq. (16),*

$$L(z) - L(z') = (N - N') + (D - D').$$

Since $\mathrm{mat}_{\mathrm{sl}}$ *is an isometry onto the strictly lower entries,* $\|N - N'\|_F = \|a - a'\|_2$*. By the mean value theorem on* $[\ell_{\min}, \ell_{\max}]$*,* $|e^{\ell_t} - e^{\ell'_t}| \le e^{\ell_{\max}}|\ell_t - \ell'_t|$ *for every* t*, so* $\|D - D'\|_F = \big(\sum_t (e^{\ell_t} - e^{\ell'_t})^2\big)^{1/2} \le e^{\ell_{\max}}\|\ell - \ell'\|_2$*. Using the triangle inequality and* $\|a - a'\|_2,\ \|\ell - \ell'\|_2 \le \|z - z'\|_2$*,*

$$\|L(z) - L(z')\|_F \le \|a - a'\|_2 + e^{\ell_{\max}}\|\ell - \ell'\|_2 \le \big(1 + e^{\ell_{\max}}\big)\|z - z'\|_2 = \kappa_L\|z - z'\|_2.$$

For the inverses, $L(z)^{-1} - L(z')^{-1} = L(z)^{-1}\big(L(z') - L(z)\big)L(z')^{-1}$*, so by the spectral bounds and* $\|\cdot\|_2 \le \|\cdot\|_F$*,*

$$\|L(z)^{-1} - L(z')^{-1}\|_2 \le \|L(z)^{-1}\|_2\,\|L(z') - L(z)\|_F\,\|L(z')^{-1}\|_2 \le \sigma_-^{-2}\kappa_L\|z - z'\|_2.$$

Block-diagonal frames. L_i *is block diagonal with lower-triangular blocks* $L(z_{i,b})$*, hence lower triangular with* $\det L_i = \prod_b \det L(z_{i,b}) > 0$*, and* $g_i = \bigoplus_b g(z_{i,b})$ *is SPD. The singular values of a block-diagonal matrix are the union of those of its blocks, so the spectral bounds hold for* L_i *and* g_i *with the same* $\sigma_\pm$*. For the Lipschitz bounds,*

$$\|L_i - L'_i\|_F^2 = \sum_b \|L(z_{i,b}) - L(z'_{i,b})\|_F^2 \le \kappa_L^2 \sum_b \|z_{i,b} - z'_{i,b}\|_2^2 = \kappa_L^2\|Z_i - Z'_i\|_F^2,$$

and, since the spectral norm of a block-diagonal matrix is the maximum over its blocks,

$$\|L_i^{-1} - L_i'^{-1}\|_2 = \max_b \|L(z_{i,b})^{-1} - L(z'_{i,b})^{-1}\|_2 \le \sigma_-^{-2}\kappa_L \max_b \|z_{i,b} - z'_{i,b}\|_2 \le \sigma_-^{-2}\kappa_L\|Z_i - Z'_i\|_F.$$

C Proof of Proposition 1

Proposition 1 (Frame-Aligned Transport) *Fix a recurrent step* l *with admissible geometric states, and let* $T^l_{j\to i,b} = (L^l_{i,b})^{-1}L^l_{j,b}$ *and* $M^l_j = L^l_j W_\phi (L^l_j)^{-1}$*. Then* $m^l_{j\to i,b} = T^l_{j\to i,b}(\phi(\xi^l_j))_b$ *with* $\|T^l_{j\to i,b}\|_2 \le \sigma_+/\sigma_-$*, and the frame maps satisfy* $T^l_{i\to i,b} = I_m$ *and* $T^l_{j\to i,b}T^l_{k\to j,b} = T^l_{k\to i,b}$*, so their product along any walk depends only on its endpoints. In environment coordinates, the aggregate and the residuals* $\delta^l_{ij} := m^l_{j\to i} - \phi(\xi^l_i)$ *read*

$$L^l_i\bar{\xi}^l_i = \sum\nolimits_{j\in\tilde{\mathcal{N}}(i)} \omega^l_{ij}\, M^l_j h^l_j, \qquad L^l_i\delta^l_{ij} = M^l_j h^l_j - M^l_i h^l_i,$$

where each M^l_j *is similar to* W_ϕ *with* $\|M^l_j\|_2 \le (\sigma_+/\sigma_-)\|W_\phi\|_2$*, and* $M^l_j = cI_d$ *when* $W_\phi = cI_d$*.*

Proof 2 *Throughout,* $\|\cdot\|_2$ *denotes the Euclidean norm of a vector and the spectral norm of a matrix. The geometric states are admissible, so Lemma 1 applies to every block frame* $L^l_{i,b}$ *and to every block-diagonal frame* L^l_i*.*

Weights. *For finite* ξ^l *and admissible* Z^l*,* $h^l_i = L^l_i\xi^l_i$ *and* $e^l_{ij} = \|h^l_i - h^l_j\|_2^2$ *are finite, so every term* $\exp(-e^l_{ij}/\tau)$ *in Eq.* (4) *is strictly positive and the normalizer is finite. Hence* $\omega^l_{ij} > 0$ *and* $\sum_{j\in\tilde{\mathcal{N}}(i)} \omega^l_{ij} = 1$*.*

Transport. *By Lemma 1,* $L^l_{i,b}$ *is invertible, so the triangular solve in Eq.* (5) *returns the unique* $m^l_{j\to i,b}$ *with* $L^l_{i,b}m^l_{j\to i,b} = L^l_{j,b}(\phi(\xi^l_j))_b$*, i.e.* $m^l_{j\to i,b} = (L^l_{i,b})^{-1}L^l_{j,b}(\phi(\xi^l_j))_b = T^l_{j\to i,b}(\phi(\xi^l_j))_b$*. Submultiplicativity and the spectral bounds of Lemma 1 give* $\|T^l_{j\to i,b}\|_2 \le \|(L^l_{i,b})^{-1}\|_2\,\|L^l_{j,b}\|_2 \le \sigma_-^{-1}\sigma_+$*.*

Composition. *Directly from the definition,* $T^l_{i\to i,b} = (L^l_{i,b})^{-1}L^l_{i,b} = I_m$ *and*

$$T^l_{j\to i,b}\,T^l_{k\to j,b} = (L^l_{i,b})^{-1}L^l_{j,b}(L^l_{j,b})^{-1}L^l_{k,b} = (L^l_{i,b})^{-1}L^l_{k,b} = T^l_{k\to i,b}.$$

For a walk $i_0 \to i_1 \to \cdots \to i_k$*, induction on* k *with this rule gives* $T^l_{i_{k-1}\to i_k,b}\cdots T^l_{i_0\to i_1,b} = T^l_{i_0\to i_k,b} = (L^l_{i_k,b})^{-1}L^l_{i_0,b}$*, which involves only the frames of the two endpoints; for a closed walk it equals* I_m*.*

Environment-coordinate form. *Taking the b-th block of Eq.* (6)*, multiplying by $L^l_{i,b}$ and using linearity,*

$$L^l_{i,b}(\bar{\xi}^l_i)_b = \sum_{j\in\tilde{\mathcal{N}}(i)} \omega^l_{ij} L^l_{i,b} m^l_{j\to i,b} = \sum_{j\in\tilde{\mathcal{N}}(i)} \omega^l_{ij} L^l_{j,b}\big(\phi(\xi^l_j)\big)_b.$$

All nodes share the block partition and L^l_i is block diagonal, so stacking this identity over $b = 1,\dots,B$ yields $L^l_i\bar{\xi}^l_i = \sum_j \omega^l_{ij} L^l_j W_\phi \xi^l_j$. Since $\xi^l_j = (L^l_j)^{-1}h^l_j$, each summand equals $\omega^l_{ij} L^l_j W_\phi (L^l_j)^{-1}h^l_j = \omega^l_{ij} M^l_j h^l_j$, which is the stated form. By construction $M^l_j = L^l_j W_\phi (L^l_j)^{-1}$ is similar to W_ϕ, and $\|M^l_j\|_2 \le \|L^l_j\|_2\,\|W_\phi\|_2\,\|(L^l_j)^{-1}\|_2 \le (\sigma_+/\sigma_-)\|W_\phi\|_2$ by Lemma 1.

Residuals in environment coordinates. *For $j\in\tilde{\mathcal{N}}(i)$, stacking the blocks of Eq.* (5) *as above gives $L^l_i m^l_{j\to i} = L^l_j W_\phi \xi^l_j$, while $L^l_i\phi(\xi^l_i) = L^l_i W_\phi (L^l_i)^{-1}h^l_i = M^l_i h^l_i$. Hence*

$$L^l_i\,\delta^l_{ij} = L^l_i m^l_{j\to i} - L^l_i\phi(\xi^l_i) = L^l_j W_\phi (L^l_j)^{-1}h^l_j - M^l_i h^l_i = M^l_j h^l_j - M^l_i h^l_i.$$

Scalar transform. *If $W_\phi = cI_d$, then $M^l_j = cL^l_j(L^l_j)^{-1} = cI_d$.*

D Proof of Proposition 2

Proposition 2 (Second-Order Refinement) *Fix a step l. Write $\mathcal{R}^l_{i,b} = \{(\omega^l_{ij}, \delta^l_{ij,b})\}_{j\in\tilde{\mathcal{N}}(i)}$ for the residual configuration of node i in block b, with weighted mean $\bar{\delta}^l_{i,b}$, and call an update* first-order *if it depends on $\mathcal{R}^l_{i,b}$ only through the weights and $\bar{\delta}^l_{i,b}$. Let i, i' be nodes with $\xi^l_i = \xi^l_{i'}$, $Z^l_i = Z^l_{i'}$, $u_i = u_{i'}$, and $\bar{\delta}^l_{i,b} = \bar{\delta}^l_{i',b}$ for all b.*

(i) Every first-order update, including the mean-only target $C^{l,\mathrm{mean}}_{i,b} = \bar{\delta}^l_{i,b}(\bar{\delta}^l_{i,b})^\top + \epsilon_s I_m$, the controller of Eq. (11)*, and the feature update of Eq.* (7)*, returns identical outputs for i and i'; in particular $\xi^{l+1}_i = \xi^{l+1}_{i'}$ and $\Delta z^l_{i,b} = \Delta z^l_{i',b}$.*

(ii) The second-order target is injective in the statistic: $C^l_{i,b} \ne C^l_{i',b}$ implies $\hat{z}^l_{i,b} \ne \hat{z}^l_{i',b}$. If $\lambda \in (0,1]$ and Eq. (12) *is not clipped in block b, then $Z^{l+1}_i \ne Z^{l+1}_{i'}$, and $h^{l+1}_i \ne h^{l+1}_{i'}$ unless $\xi^{l+1}_i \in \ker(L^{l+1}_i - L^{l+1}_{i'})$, a proper subspace.*

(iii) If i has at least two neighbors and $\mathcal{R}^l_{i',b}$ ranges over all configurations with the same weights and first moment, those with $C^l_{i',b} = C^l_{i,b}$ form a Lebesgue-null set.

(iv) The target depends on $\mathcal{R}^l_{i,b}$ only through $C^l_{i,b}$ and is therefore strictly weaker than an injective aggregator: for $m = 1$ there are distinct three-point residual sets with equal weights, first, and second moments. Conversely, for $m \ge 2$ the sets $\{0, \pm v\}$ and $\{0, \pm w\}$ with $v = (1,1,0,\dots,0)^\top$, $w = (1,-1,0,\dots,0)^\top$ and equal weights on $\pm v, \pm w$ share the mean-only and the diagonal target $C^{l,\mathrm{diag}}_{i,b} = \mathrm{Diag}\big(\sum_j \omega^l_{ij}\,\delta^l_{ij,b}\odot\delta^l_{ij,b}\big) + \epsilon_s I_m$, yet $C^l_{i,b} \ne C^l_{i',b}$.

Proof 3 Preliminaries. *Since the weights sum to one (Proposition 1),*

$$\bar{\delta}^l_{i,b} = \sum_{j\in\tilde{\mathcal{N}}(i)} \omega^l_{ij}\, m^l_{j\to i,b} - \big(\phi(\xi^l_i)\big)_b = (\bar{\xi}^l_i)_b - \big(\phi(\xi^l_i)\big)_b. \tag{20}$$

Hence, under $\xi^l_i = \xi^l_{i'}$, the conditions $\bar{\delta}^l_{i,b} = \bar{\delta}^l_{i',b}$ for all b and $\bar{\xi}^l_i = \bar{\xi}^l_{i'}$ are equivalent. Moreover, $Z^l_i = Z^l_{i'}$ implies $L^l_i = L^l_{i'}$, and the self residual vanishes, $\delta^l_{ii,b} = 0$, because $T^l_{i\to i,b} = I_m$. By Eq. (8)*, $C^l_{i,b} \succeq \epsilon_s I_m \succ 0$, so the target of Eq.* (10) *exists for every fixed jitter $\eta \ge 0$. Throughout, the statistics are regarded as functions of the residual configuration $\mathcal{R}^l_{i,b}$.*

(i) First-order invariance. *By definition, a first-order update depends on $\mathcal{R}^l_{i,b}$ only through the weights and $\bar{\delta}^l_{i,b}$, which coincide for i and i'; hence its outputs coincide. This covers the three updates listed in the statement:*

- *the mean-only target is a function of $\bar{\delta}^l_{i,b}$ alone;*

- *the controller input* $q^l_{i,b} = [(\xi^l_i)_b \| (\bar{\xi}^l_i)_b \| u_i]$ *coincides for the two nodes, so* $\Gamma_\theta(q^l_{i,b}) = \Gamma_\theta(q^l_{i',b})$ *and, by Eq.* (11), $\Delta z^l_{i,b} = \Delta z^l_{i',b}$*;*
- *the feature update of Eq.* (7) *is a function of* $(\xi^l_i, \bar{\xi}^l_i)$ *only, so* $\xi^{l+1}_i = \xi^{l+1}_{i'}$.

(ii) Injectivity and propagation. *For a fixed jitter,* $C \mapsto R = \mathrm{SChol}(C)$ *is the Cholesky factor of* $C + \eta I_m$*, a bijection between* $\mathrm{SPD}(m)$ *and lower-triangular matrices with positive diagonal (Lin, 2019), hence injective. The map*

$$R \;\mapsto\; \hat{z} = \left[\, \mathrm{Svec}_{\mathrm{sl}}(R)^\top,\; \log(\mathrm{diag}\, R)^\top \right]^\top \tag{21}$$

of Eq. (10) *is injective because* R *is recovered from* $\hat{z}$ *as* $\mathrm{mat}_{\mathrm{sl}}(\hat{a}) + \mathrm{Diag}(e^{\hat{\ell}})$*. Therefore* $C^l_{i,b} \neq C^l_{i',b}$ *implies* $\hat{z}^l_{i,b} \neq \hat{z}^l_{i',b}$*.*

When Eq. (12) *is not clipped in block* b*, the projection acts as the identity. Since* $z^l_{i,b} = z^l_{i',b}$ *and* $\Delta z^l_{i,b} = \Delta z^l_{i',b}$ *by (i),*

$$z^{l+1}_{i,b} - z^{l+1}_{i',b} = \lambda\big(\hat{z}^l_{i,b} - \hat{z}^l_{i',b}\big) \neq 0 \qquad \text{for } \lambda > 0, \tag{22}$$

so $Z^{l+1}_i \neq Z^{l+1}_{i'}$*. The map* $z \mapsto L(z)$ *of Eq.* (16) *is injective, since* $\ell_r = \log L_{rr}$ *and* $a_{rs} = L_{rs}$ *recover* z *from* $L(z)$*; hence* $L^{l+1}_{i,b} \neq L^{l+1}_{i',b}$ *and, as block-diagonal matrices,* $L^{l+1}_i \neq L^{l+1}_{i'}$*. With* $\xi^{l+1}_i = \xi^{l+1}_{i'}$*,*

$$h^{l+1}_i - h^{l+1}_{i'} = \big(L^{l+1}_i - L^{l+1}_{i'}\big)\, \xi^{l+1}_i, \tag{23}$$

which vanishes only if ξ^{l+1}_i *lies in the kernel of the nonzero matrix* $L^{l+1}_i - L^{l+1}_{i'}$*, a proper subspace of* $\mathbb{R}^d$*.*

(iii) Genericity. *Let* $j = 1, \ldots, k$ *with* $k \geq 2$ *index the neighbors of* i' *other than itself, fix their weights* $(\omega_j)_{j=1}^k$*, all positive by Proposition 1, and parametrize the residuals of* i' *in block* b *by* $\delta = (\delta_1, \ldots, \delta_k) \in \mathbb{R}^{mk}$*; the self residual is zero and does not enter. Equal first moment is the affine constraint*

$$\sum_{j=1}^{k} \omega_j \delta_j = \bar{\delta}^l_{i,b}, \tag{24}$$

which defines an affine subspace $\mathcal{A} \subset \mathbb{R}^{mk}$ *of dimension* $m(k-1)$*. On* $\mathcal{A}$ *the map* $C(\delta) = \sum_j \omega_j \delta_j \delta_j^\top + \epsilon_s I_m$ *is polynomial, and so is* $p(\delta) := C(\delta)_{aa} - (C^l_{i,b})_{aa}$ *for any fixed coordinate* a*.*

The polynomial p *is not identically zero on* $\mathcal{A}$*. For* $\delta \in \mathcal{A}$ *and* $t \in \mathbb{R}$*, the perturbation* $\delta_1 \mapsto \delta_1 + te_a$*,* $\delta_2 \mapsto \delta_2 - (\omega_1/\omega_2)te_a$ *keeps* $\sum_j \omega_j \delta_j$ *fixed, hence stays in* $\mathcal{A}$*, and changes* $C(\delta)_{aa}$ *by*

$$\omega_1\big[(\delta_{1,a} + t)^2 - \delta^2_{1,a}\big] + \omega_2\Big[\big(\delta_{2,a} - \tfrac{\omega_1}{\omega_2}t\big)^2 - \delta^2_{2,a}\Big] = 2\omega_1 t\,(\delta_{1,a} - \delta_{2,a}) + \Big(\omega_1 + \frac{\omega_1^2}{\omega_2}\Big)t^2, \tag{25}$$

a polynomial in t *with positive leading coefficient, hence nonconstant along the line. The zero set of a nonzero polynomial on an affine space has Lebesgue measure zero, so*

$$\{\delta \in \mathcal{A} : C(\delta) = C^l_{i,b}\} \subseteq \{\delta \in \mathcal{A} : p(\delta) = 0\} \tag{26}$$

is a Lebesgue-null subset of $\mathcal{A}$*.*

(iv) Limits of second-order separation. *The target* $\hat{z}^l_{i,b}$ *is a function of* $C^l_{i,b}$ *alone, so any two residual configurations with equal weighted second moments yield equal targets.*

The case $m = 1$. *Take equal weights* $1/3$*,* $q \in (0,1)$*, and* $p, r = \frac{-q \pm \sqrt{4-3q^2}}{2}$*, which are real since* $4 - 3q^2 > 0$*. Then* p *and* r *are the roots of* $t^2 + qt + (q^2 - 1) = 0$*, so* $p + r = -q$*,* $pr = q^2 - 1$*, and* $p^2 + r^2 = (p+r)^2 - 2pr = 2 - q^2$*. The set* $\{q, p, r\}$ *therefore has first moment* $(q + p + r)/3 = 0$ *and second moment* $(q^2 + p^2 + r^2)/3 = 2/3$*, as does* $\{-1, 0, 1\}$*, while its third moment is*

$$\tfrac{1}{3}\big(q^3 + p^3 + r^3\big) = \tfrac{1}{3}\big(q^3 + (p+r)^3 - 3pr(p+r)\big) = \tfrac{1}{3}\big(q^3 - q^3 + 3q(q^2-1)\big) = q(q^2-1) \neq 0. \tag{27}$$

The two sets are distinct multisets, so an injective aggregator separates them, whereas the second-order target does not.

The case $m \geq 2$. *Consider the configurations* $\{0, \pm v\}$ *and* $\{0, \pm w\}$ *with weights* $(\omega_0, \omega, \omega)$, $\omega > 0$. *In both, the first moment is* $\omega_0 \cdot 0 + \omega v - \omega v = 0$ *(respectively with* w*), so the mean-only targets are both* $\epsilon_s I_m$. *The full statistics are*

$$C^l_{i,b} = 2\omega\, vv^\top + \epsilon_s I_m, \qquad C^l_{i',b} = 2\omega\, ww^\top + \epsilon_s I_m, \tag{28}$$

whose $(1,2)$ *entries are* $+2\omega$ *and* -2ω, *so they differ. The diagonal statistics use* $v \odot v = w \odot w = (1, 1, 0, \ldots, 0)^\top$ *and therefore coincide,* $\mathrm{Diag}\big(2\omega(1, 1, 0, \ldots, 0)^\top\big) + \epsilon_s I_m$. *Such configurations arise, for instance, from two neighbors whose transformed features differ from* $\phi(\xi^l_i)$ *by* $\pm v$ *(respectively* $\pm w$*) and share the same frame and environment distance to the target.*

E Proof of Proposition 3

Proposition 3 (Stability and Convergence) *Fix* $\tau > 0$, $\epsilon_s > 0$, $\eta \geq 0$ *and a feature bound* Ξ. *Let* $\mathcal{D} = \{(\xi, Z) : \|\xi_i\|_2 \leq \Xi,\ z_{i,b} \in \mathcal{Z}\}$, *let* $F : (\xi^l, Z^l) \mapsto (\xi^{l+1}, Z^{l+1})$ *denote one recurrent step, and let* $d_\infty(S, S') = \max_i \big(\|\xi_i - \xi'_i\|_2 + \|Z_i - Z'_i\|_F\big)$.

(i) Eq. (12) *is a projected proximal step:* $\Pi_{\mathcal{Z}}$ *is the Euclidean projection onto* $\mathcal{Z}$, *and* $z^{l+1}_{i,b}$ *is the unique minimizer over* $\mathcal{Z}$ *of*

$$\tfrac{1-\lambda}{2}\|z - z^l_{i,b}\|_2^2 + \tfrac{\lambda}{2}\|z - \hat{z}^l_{i,b}\|_2^2 - \gamma\langle \Delta z^l_{i,b},\, z - z^l_{i,b}\rangle.$$

It differs from the uncorrected update $z^{l+1,0}_{i,b} = \Pi_{\mathcal{Z}}\big((1-\lambda)z^l_{i,b} + \lambda\hat{z}^l_{i,b}\big)$ *by at most* $\gamma\|\Delta z^l_{i,b}\|_2$.

(ii) One step is Lipschitz on $\mathcal{D}$: $d_\infty\big(F(S), F(S')\big) \leq K\, d_\infty(S, S')$ *with* $K = c_\xi + (1-\lambda) + \lambda c_{\hat{Z}} + \gamma c_\Delta$, *where the constants depend only on* m, B, $\sigma_\pm$, κ_L, ϵ_s, η, τ, Ξ, λ, γ, *and the norms of* W_ϕ, W_r, *and* Γ_θ, *but not on node degrees.*

(iii) For fixed ξ, *let* $\Phi_\xi(Z) = \Pi_{\mathcal{Z}}\big((1-\lambda)Z + \lambda\hat{Z}(Z;\xi) + \gamma\Delta Z(Z;\xi)\big)$. *If* $Z^\star$ *is an unclipped fixed point of* Φ_ξ *whose Jacobian* $J = \partial\Phi_\xi(Z^\star)$ *has spectral radius* $\rho(J) < 1$, *then* $Z^\star$ *is locally attracting with rate* $\rho(J)$: *for every* $\varepsilon > 0$ *there are* $c > 0$ *and a neighborhood* $\mathcal{U}$ *of* $Z^\star$ *such that* $\|Z^l - Z^\star\|_F \leq c\,(\rho(J) + \varepsilon)^l\, \|Z^0 - Z^\star\|_F$ *for all* $Z^0 \in \mathcal{U}$.

(iv) If moreover $\hat{Z}(\cdot;\xi)$ *and* $\Delta Z(\cdot;\xi)$ *are* $L_{\hat{z}}$- *and* L_Δ-*Lipschitz on* $\mathcal{Z}^{nB}$ *and* $\rho := (1-\lambda) + \lambda L_{\hat{z}} + \gamma L_\Delta < 1$, *the fixed point is unique and* $\|Z^{l+1} - Z^l\|_F \leq \rho^l \|Z^1 - Z^0\|_F$ *from every initialization.*

Proof 4 *Throughout,* $\|\cdot\|$ *denotes the Euclidean norm of a vector and the spectral norm of a matrix,* $\|\cdot\|_F$ *the Frobenius norm, and* $\|\cdot\|_1$, $\|\cdot\|_\infty$ *the vector* ℓ_1 *and* ℓ_∞ *norms. The constants* $\sigma_\pm$ *and* κ_L *are those of Lemma 1.*

(i) Projected proximal step. $\mathcal{Z}$ *is a product of closed intervals, and the objective* $\frac{1}{2}\|z - y\|^2$ *separates over coordinates, so its minimizer over* $\mathcal{Z}$ *is obtained by clipping each coordinate of* y *to its interval; hence* $\Pi_{\mathcal{Z}}$ *is the Euclidean projection. Let* $u = (1-\lambda)z^l_{i,b} + \lambda\hat{z}^l_{i,b}$ *and* $c = \Delta z^l_{i,b}$. *Expanding the squares and using* $(1-\lambda) + \lambda = 1$,

$$\tfrac{1-\lambda}{2}\|z - z^l_{i,b}\|^2 + \tfrac{\lambda}{2}\|z - \hat{z}^l_{i,b}\|^2 - \gamma\langle c, z - z^l_{i,b}\rangle = \tfrac{1}{2}\|z - (u + \gamma c)\|^2 + \mathrm{const}, \tag{29}$$

so the unique minimizer over $\mathcal{Z}$ *is* $\Pi_{\mathcal{Z}}(u + \gamma c) = z^{l+1}_{i,b}$ *by Eq.* (12). *The projection onto a closed convex set is non-expansive, whence*

$$\|z^{l+1}_{i,b} - z^{l+1,0}_{i,b}\| = \|\Pi_{\mathcal{Z}}(u + \gamma c) - \Pi_{\mathcal{Z}}(u)\| \leq \gamma\|c\|. \tag{30}$$

(ii) One-step Lipschitz bound. *Let* $S = (\xi, Z)$ *and* $S' = (\xi', Z')$ *lie in* $\mathcal{D}$, *write* $d := d_\infty(S, S')$, *and let primes denote quantities computed from* S'. *Set* $w_\phi := \|W_\phi\|$, $w_r := \|W_r\|$, $\bar{U} := \max_i \|u_i\|$,

$\ell_\Gamma := \mathrm{Lip}(\Gamma_\theta) \le 1.1\|W_{\Gamma,2}\|\|W_{\Gamma,1}\|$ *(since* $\sup|\mathrm{SiLU}'| < 1.1$*), and*

$$H := \sigma_+\Xi, \qquad c_h := \max(\kappa_L\Xi, \sigma_+), \qquad c_\omega := 16Hc_h/\tau, \qquad \bar{M} := \sigma_+\sigma_-^{-1}w_\phi\Xi,$$
$$c_m := w_\phi\big(\sigma_-^{-2}\sigma_+\kappa_L\Xi + \sigma_-^{-1}\kappa_L\Xi + \sigma_-^{-1}\sigma_+\big), \qquad c_{\bar\xi} := \bar{M}c_\omega + c_m,$$
$$c_\xi := 2 + c_{\bar\xi} + \tfrac{w_r}{4}(1 + c_{\bar\xi})(\bar{M} + \Xi), \qquad \bar{D} := \bar{M} + w_\phi\Xi, \qquad \bar{C} := \bar{D}^2 + \epsilon_s,$$
$$c_C := \bar{D}^2c_\omega + 2\bar{D}(c_m + w_\phi), \qquad L_{\mathrm{chol}} := \frac{(\bar{C}+\eta)^{1/2}}{\epsilon_s+\eta}, \qquad c_z := \max\big\{1, (\epsilon_s+\eta)^{-1/2}\big\},$$
$$c_{\hat{Z}} := \sqrt{B}\, c_z L_{\mathrm{chol}} c_C, \qquad \bar{G} := \sup_{\|q\| \le \Xi + \bar{M} + \bar{U}} \|\Gamma_\theta(q)\|, \qquad c_\Delta := \sqrt{3B}\, \max(\bar{G}, 1)\, \ell_\Gamma(1 + c_{\bar\xi}).$$

Each bound below is derived for an arbitrary node i; since d majorizes the perturbation of i and of every neighbor, the same d appears throughout.

Frames. *By Lemma 1,* $\|L_i\| \le \sigma_+$, $\|L_i^{-1}\| \le \sigma_-^{-1}$, $\|L_i - L_i'\|_F \le \kappa_L\|Z_i - Z_i'\|_F \le \kappa_L d$ *and* $\|L_i^{-1} - L_i'^{-1}\| \le \sigma_-^{-2}\kappa_L d$.

Environment representations and weights. $\|h_i\| = \|L_i\xi_i\| \le \sigma_+\Xi = H$ *and*

$$\|h_i - h_i'\| \le \|L_i - L_i'\|\,\|\xi_i\| + \|L_i'\|\,\|\xi_i - \xi_i'\| \le \kappa_L\Xi\|Z_i - Z_i'\|_F + \sigma_+\|\xi_i - \xi_i'\| \le c_h d. \quad (31)$$

Since $\|h_i - h_j\| \le 2H$ *and* $|\,\|x\|^2 - \|y\|^2| \le (\|x\| + \|y\|)\|x - y\|$, $|e_{ij} - e_{ij}'| \le 4H\big(\|h_i - h_i'\| + \|h_j - h_j'\|\big) \le 8Hc_hd$. *The softmax map is* 2*-Lipschitz from ℓ_∞ to ℓ_1: its Jacobian at ω is* $\mathrm{Diag}(\omega) - \omega\omega^\top$*, and for any v,*

$$\big\|(\mathrm{Diag}(\omega) - \omega\omega^\top)v\big\|_1 = \sum_k \omega_k\,|v_k - \langle\omega, v\rangle| \le \max_k v_k - \min_k v_k \le 2\|v\|_\infty, \quad (32)$$

because $\langle\omega, v\rangle$ is a convex combination of the v_k; the bound extends to finite differences by the integral form of the mean value theorem. Applying it to Eq. (4)*,* $\|\omega_{i\cdot} - \omega_{i\cdot}'\|_1 \le \frac{2}{\tau}\max_j|e_{ij} - e_{ij}'| \le c_\omega d$.

Messages and aggregate. *By Proposition 1,* $m_{j\to i} = L_i^{-1}L_jW_\phi\xi_j$*, so* $\|m_{j\to i}\| \le \bar{M}$. *Writing*

$$m_{j\to i} - m_{j\to i}' = (L_i^{-1} - L_i'^{-1})L_jW_\phi\xi_j + L_i'^{-1}(L_j - L_j')W_\phi\xi_j + L_i'^{-1}L_j'W_\phi(\xi_j - \xi_j') \quad (33)$$

and bounding the three terms by $\sigma_-^{-2}\kappa_L d\cdot\sigma_+w_\phi\Xi$, $\sigma_-^{-1}\cdot\kappa_L d\cdot w_\phi\Xi$ and $\sigma_-^{-1}\sigma_+w_\phi d$ gives $\|m_{j\to i} - m_{j\to i}'\| \le c_m d$. *Since the weights are nonnegative and sum to one,*

$$\begin{aligned}\|\bar\xi_i - \bar\xi_i'\| &\le \sum_j |\omega_{ij} - \omega_{ij}'|\,\|m_{j\to i}\| + \sum_j \omega_{ij}'\|m_{j\to i} - m_{j\to i}'\| \\ &\le \bar{M}\|\omega_{i\cdot} - \omega_{i\cdot}'\|_1 + \max_j\|m_{j\to i} - m_{j\to i}'\| \le c_{\bar\xi}d,\end{aligned} \quad (34)$$

and $\|\bar\xi_i\| \le \bar{M}$.

Feature update. *The sigmoid is $\frac{1}{4}$-Lipschitz with values in* $[0,1]$*, so the gate of Eq.* (7) *satisfies* $\|r_i - r_i'\| \le \frac{w_r}{4}\big(\|\xi_i - \xi_i'\| + \|\bar\xi_i - \bar\xi_i'\|\big) \le \frac{w_r}{4}(1 + c_{\bar\xi})d$. *Writing* $\xi_i^{l+1} = \xi_i + r_i \odot (\bar\xi_i - \xi_i)$*,*

$$\xi_i^{l+1} - \xi_i'^{\,l+1} = (\xi_i - \xi_i') + (r_i - r_i')\odot(\bar\xi_i - \xi_i) + r_i'\odot\big[(\bar\xi_i - \bar\xi_i') - (\xi_i - \xi_i')\big], \quad (35)$$

and using $\|a\odot b\| \le \|a\|\,\|b\|$, $\|r_i'\odot b\| \le \|b\|$ *and* $\|\bar\xi_i - \xi_i\| \le \bar{M} + \Xi$*,*

$$\|\xi_i^{l+1} - \xi_i'^{\,l+1}\| \le d + \tfrac{w_r}{4}(1 + c_{\bar\xi})(\bar{M} + \Xi)\,d + (c_{\bar\xi} + 1)\,d = c_\xi d. \quad (36)$$

Residual statistic. $\delta_{ij} = m_{j\to i} - W_\phi\xi_i$ *satisfies* $\|\delta_{ij}\| \le \bar{D}$ *and* $\|\delta_{ij} - \delta_{ij}'\| \le (c_m + w_\phi)d$*, and the same bounds hold for each block $\delta_{ij,b}$. Since* $\|\delta\delta^\top - \delta'\delta'^\top\|_F \le 2\bar{D}\|\delta - \delta'\|$ *and* $\|\delta\delta^\top\|_F = \|\delta\|^2 \le \bar{D}^2$*, the statistic of Eq.* (8) *satisfies*

$$\|C_{i,b} - C_{i,b}'\|_F \le \bar{D}^2\|\omega_{i\cdot} - \omega_{i\cdot}'\|_1 + 2\bar{D}\max_j\|\delta_{ij,b} - \delta_{ij,b}'\| \le c_C d, \quad (37)$$

and $\epsilon_sI_m \preceq C_{i,b} \preceq \bar{C}I_m$ *because* $\delta\delta^\top \preceq \|\delta\|^2 I_m$.

Cholesky target. *Let $\mathcal{K} = \{C : \epsilon_s I_m \preceq C \preceq \bar{C} I_m\}$, a compact convex subset of $\mathrm{SPD}(m)$, and let $R(C)$ denote the lower Cholesky factor of $C + \eta I_m$. We bound the two maps $C \mapsto R$ and $R \mapsto \hat{z}$ in turn.*

The map $C \mapsto R(C)$ is real-analytic on $\mathrm{SPD}(m)$ (Lin, 2019). Differentiating $RR^\top = C + \eta I_m$ gives $R^{-1}\, dC\, R^{-\top} = X + X^\top$ with $X = R^{-1} dR$ lower triangular, hence

$$dR = R\,\Phi\big(R^{-1}\, dC\, R^{-\top}\big), \tag{38}$$

where Φ keeps the strictly lower part of its argument and halves the diagonal, so that $\|\Phi(Y)\|_F \le \|Y\|_F$. On $\mathcal{K}$, $\|R\|^2 = \lambda_{\max}(C + \eta I_m) \le \bar{C} + \eta$ and $\|R^{-1}\|^2 = \lambda_{\min}(C + \eta I_m)^{-1} \le (\epsilon_s + \eta)^{-1}$, so

$$\|dR\|_F \le \|R\|\, \|R^{-1}\|^2\, \|dC\|_F \le L_{\mathrm{chol}} \|dC\|_F. \tag{39}$$

Since $\mathcal{K}$ is convex, the mean value theorem gives $\|R(C) - R(C')\|_F \le L_{\mathrm{chol}} \|C - C'\|_F$.

For the coordinates of Eq. (10)*, the strictly lower coordinates are the entries R_{rs} themselves. The diagonal entries satisfy $R_{tt}^2 = C_{tt} - C_{t,<t}\, C_{<t,<t}^{-1}\, C_{<t,t}$, a Schur complement of $C + \eta I_m$, hence $R_{tt}^2 \ge \epsilon_s + \eta$ and*

$$|\log R_{tt} - \log R'_{tt}| \le (\epsilon_s + \eta)^{-1/2}\, |R_{tt} - R'_{tt}|. \tag{40}$$

Collecting all entries in ℓ_2,

$$\|\hat{z}_{i,b} - \hat{z}'_{i,b}\| \le c_z \|R - R'\|_F \le c_z L_{\mathrm{chol}}\, c_C\, d, \tag{41}$$

and stacking the B blocks gives $\|\hat{Z}_i - \hat{Z}'_i\|_F \le \sqrt{B}\, c_z L_{\mathrm{chol}} c_C d = c_{\hat{Z}} d$.

Correction. *The controller input $q_{i,b} = [(\xi_i)_b \| (\bar{\xi}_i)_b \| u_i]$ satisfies $\|q_{i,b}\| \le \Xi + \bar{M} + \bar{U}$ and, the signature being fixed, $\|q_{i,b} - q'_{i,b}\| \le \|\xi_i - \xi'_i\| + \|\bar{\xi}_i - \bar{\xi}'_i\| \le (1 + c_{\bar{\xi}}) d$. Γ_θ is ℓ_Γ-Lipschitz, and its output on this ball is bounded by $\bar{G}$; the reshaped factors therefore satisfy $\|U\|_F, \|V\|_F, \|d_\ell\| \le \bar{G}$. Using $\|UV^\top - U'V'^\top\|_F \le \bar{G}(\|U - U'\|_F + \|V - V'\|_F)$, the correction of Eq.* (11) *satisfies*

$$\begin{aligned} \|\Delta z_{i,b} - \Delta z'_{i,b}\| &\le \|UV^\top - U'V'^\top\|_F + \|d_\ell - d'_\ell\| \\ &\le \max(\bar{G}, 1)\big(\|U - U'\|_F + \|V - V'\|_F + \|d_\ell - d'_\ell\|\big) \\ &\le \sqrt{3}\max(\bar{G}, 1)\, \|\Gamma_\theta(q_{i,b}) - \Gamma_\theta(q'_{i,b})\| \le \sqrt{3}\max(\bar{G}, 1)\, \ell_\Gamma (1 + c_{\bar{\xi}})\, d, \end{aligned} \tag{42}$$

and stacking the B blocks gives $\|\Delta Z_i - \Delta Z'_i\|_F \le c_\Delta d$.

Composition. *$\Pi_{\mathcal{Z}}$ acts coordinatewise and is non-expansive in the Frobenius norm, so by Eq.* (12)

$$\|Z_i^{l+1} - Z_i'^{\,l+1}\|_F \le (1-\lambda)\|Z_i - Z'_i\|_F + \lambda\|\hat{Z}_i - \hat{Z}'_i\|_F + \gamma\|\Delta Z_i - \Delta Z'_i\|_F \le \big((1-\lambda) + \lambda c_{\hat{Z}} + \gamma c_\Delta\big) d. \tag{43}$$

Adding the feature bound and taking the maximum over i, $d_\infty(F(S), F(S')) \le \big(c_\xi + (1 - \lambda) + \lambda c_{\hat{Z}} + \gamma c_\Delta\big) d = Kd$. None of the bounds involves the number of neighbors, so K is independent of node degrees.

(iii) Local convergence. *Write $\Phi_\xi = \Pi_{\mathcal{Z}} \circ G$ with*

$$G(Z) = (1 - \lambda) Z + \lambda \hat{Z}(Z; \xi) + \gamma \Delta Z(Z; \xi). \tag{44}$$

The argument has three steps.

Reduction to G. *Since $Z^\star$ is unclipped, $G(Z^\star) = Z^\star$ lies in the interior of $\mathcal{Z}^{nB}$. As G is continuous, $G(Z)$ stays in the interior for all Z in a neighborhood $\mathcal{V}$ of $Z^\star$, so $\Phi_\xi = G$ on $\mathcal{V}$.*

Differentiability. *On $\mathcal{V}$ the map G is continuously differentiable: it is a composition of the frame reconstruction of Eq.* (16)*, the softmax, the triangular solves, which are rational in the entries of invertible frames, the residual second moment, which is polynomial, the Cholesky factorization and elementwise logarithm on $\mathcal{K}$, and the SiLU network. Let $J = \partial G(Z^\star)$.*

Contraction near $Z^\star$. *Fix $\varepsilon > 0$ with $\rho(J) + \varepsilon < 1$. There is a vector norm $\|\cdot\|_\varepsilon$ whose induced matrix norm satisfies $\|J\|_\varepsilon \le \rho(J) + \varepsilon/2$ (Horn & Johnson, 2012, Lemma 5.6.10). By differentiability, there is a neighborhood $\mathcal{U} \subseteq \mathcal{V}$ of $Z^\star$ on which*

$$\|G(Z) - Z^\star\|_\varepsilon \le \big(\rho(J) + \varepsilon\big)\|Z - Z^\star\|_\varepsilon. \tag{45}$$

Hence G maps $\mathcal{U}$ into itself, and iterating gives

$$\|Z^l - Z^\star\|_\varepsilon \le \big(\rho(J) + \varepsilon\big)^l \|Z^0 - Z^\star\|_\varepsilon \qquad \text{for all } Z^0 \in \mathcal{U}. \tag{46}$$

Equivalence of $\|\cdot\|_\varepsilon$ and $\|\cdot\|_F$ on the finite-dimensional space $\mathbb{R}^{nB\cdot m(m+1)/2}$ yields the constant c (Ortega & Rheinboldt, 2000).

(iv) Global convergence. *With ξ fixed, $\hat{Z}(\cdot;\xi)$ and $\Delta Z(\cdot;\xi)$ are Lipschitz on $\mathcal{Z}^{nB}$ by part (ii); let $L_{\hat{z}}$ and L_Δ be their constants. Then*

$$\|G(Z) - G(Z')\|_F \le \big((1-\lambda) + \lambda L_{\hat{z}} + \gamma L_\Delta\big)\|Z - Z'\|_F = \rho\|Z - Z'\|_F, \tag{47}$$

and non-expansiveness of $\Pi_{\mathcal{Z}}$ gives $\mathrm{Lip}(\Phi_\xi) \le \rho$. The set $\mathcal{Z}^{nB}$ is a nonempty closed subset of a Euclidean space, hence complete, and Φ_ξ maps it into itself. If $\rho < 1$, the Banach fixed-point theorem yields a unique $Z^\star$ with $\Phi_\xi(Z^\star) = Z^\star$, and

$$\|Z^{l+1} - Z^l\|_F = \|\Phi_\xi(Z^l) - \Phi_\xi(Z^{l-1})\|_F \le \rho\|Z^l - Z^{l-1}\|_F \le \rho^l\|Z^1 - Z^0\|_F. \tag{48}$$

F Numerical Operators for Log-Triangular Targets

This section defines the operators used to construct the geometric update target in Eq. (10).

F.1 Strictly Lower-Triangular Vectorization

For $A \in \mathbb{R}^{m\times m}$, the operator $\mathrm{Svec}_{\mathrm{sl}}$ extracts the strictly lower-triangular entries in row-major order, so that $[\mathrm{Svec}_{\mathrm{sl}}(A)]_{\kappa(r,s)} = A_{rs}$ with $\kappa(r,s) = (r-1)(r-2)/2 + s$ for $1 \le s < r \le m$:

$$\mathrm{Svec}_{\mathrm{sl}}(A) = [A_{21},\, A_{31},\, A_{32},\, \ldots,\, A_{m1},\, \ldots,\, A_{m,m-1}]^\top \in \mathbb{R}^{m(m-1)/2}. \tag{49}$$

The reconstruction operator $\mathrm{mat}_{\mathrm{sl}}$ of Eq. (1) inverts this ordering: for $a \in \mathbb{R}^{m(m-1)/2}$,

$$\big[\mathrm{mat}_{\mathrm{sl}}(a)\big]_{rs} = \begin{cases} a_{\kappa(r,s)}, & r > s, \\ 0, & r \le s. \end{cases} \tag{50}$$

The two operators satisfy $\mathrm{Svec}_{\mathrm{sl}}\big(\mathrm{mat}_{\mathrm{sl}}(a)\big) = a$ and $\mathrm{mat}_{\mathrm{sl}}\big(\mathrm{Svec}_{\mathrm{sl}}(A)\big) = \mathrm{tril}(A,-1)$, where $\mathrm{tril}(A,-1)$ retains only the strictly lower-triangular entries of A. In particular, $\mathrm{mat}_{\mathrm{sl}}$ is an isometry from $\mathbb{R}^{m(m-1)/2}$ onto the strictly lower-triangular matrices, which is used in the proof of Lemma 1.

F.2 Stabilized Cholesky Factorization

For a symmetric matrix $C \in \mathbb{R}^{m\times m}$, let $\epsilon_0 > 0$ be the initial diagonal jitter, $\beta > 1$ its growth factor, and $T_{\max} \in \{0, 1, \ldots\}$ the maximum retry index. Define the jitter schedule

$$\epsilon_t = \beta^t \epsilon_0, \qquad C_t = C + \epsilon_t I_m, \qquad t = 0, \ldots, T_{\max}, \tag{51}$$

and, whenever the set is nonempty, the first admissible index

$$t^* = \min\{t \in \{0, \ldots, T_{\max}\} : C_t \in \mathrm{SPD}(m)\}. \tag{52}$$

The stabilized factorization returns the Cholesky factor of the first admissible candidate,

$$\mathrm{SChol}(C; \epsilon_0, \beta, T_{\max}) = \mathrm{chol}(C_{t^*}), \qquad RR^\top = C + \epsilon_{t^*} I_m, \tag{53}$$

where chol returns the unique lower-triangular factor R with positive diagonal. The main text abbreviates this operator as $\mathrm{SChol}(C)$ and denotes the selected jitter ϵ_{t^*} by η. The schedule contains $T_{\max} + 1$ candidate factorizations, including the initial attempt.

For the residual second-moment matrix, Eq. (8) gives $C^l_{i,b} \succeq \epsilon_s I_m$ with $\epsilon_s > 0$, hence $C^l_{i,b} + \epsilon_t I_m \succeq (\epsilon_s + \epsilon_t) I_m \succ 0$ for every t. The factorization is therefore well-defined, with $t^* = 0$ in exact arithmetic; in floating-point arithmetic, retries are triggered by numerical factorization failure. The jitter ϵ_t is a numerical safeguard and is separate from the regularization ϵ_s that enters the statistic itself.

Table 6: Statistics of the experimental datasets.

Datasets	Graphs	Avg. Nodes	Avg. Edges	Classes
CiteSeer	-	3,327	9,104	6
PubMed	-	19,717	88,648	3
CS	-	18,333	163,788	15
Physics	-	34,493	495,924	5
Photo	-	7,650	238,162	8
Computers	-	13,752	491,722	10
PROTEINS	1,113	39.10	72.80	2
NCI1	4,110	29.87	32.30	2
Mutagenicity	4,337	30.32	30.77	2
FRANKENSTEIN	4,337	16.90	17.88	2
BBBP	2,050	23.90	51.60	2
ogbg-molhiv	41,127	25.50	27.50	2

F.3 Construction of the Coordinate Target

Given $R^l_{i,b} = \mathrm{SChol}(C^l_{i,b})$, the target coordinates are

$$\widehat{z}^l_{i,b} = \begin{bmatrix} \widehat{a}^l_{i,b} \\ \widehat{\ell}^l_{i,b} \end{bmatrix} = \begin{bmatrix} \mathrm{Svec}_{\mathrm{sl}}(R^l_{i,b}) \\ \log\Big(\mathrm{diag}(R^l_{i,b})\Big) \end{bmatrix} \in \mathbb{R}^{m(m+1)/2}, \tag{54}$$

where diag extracts the diagonal as a column vector and the logarithm acts elementwise; the positive diagonal of $R^l_{i,b}$ makes $\widehat{\ell}^l_{i,b}$ well-defined. With the coordinate ordering of Eq. (1),

$$\mathrm{mat}_{\mathrm{sl}}(\widehat{a}^l_{i,b}) + \mathrm{Diag}\Big(\exp(\widehat{\ell}^l_{i,b})\Big) = R^l_{i,b}, \tag{55}$$

so $\widehat{z}^l_{i,b}$ encodes $R^l_{i,b}$ in frame coordinates. Since Cholesky factorization is a bijection between $\mathrm{SPD}(m)$ and lower-triangular matrices with positive diagonals, the encoding is lossless: $R^l_{i,b}(R^l_{i,b})^\top = C^l_{i,b} + \epsilon_{t^*} I_m$.

The target frame has a direct geometric meaning. For $h = R^l_{i,b} x$,

$$\|x\|_2^2 = \big\|(R^l_{i,b})^{-1} h\big\|_2^2 = h^\top \big(C^l_{i,b} + \epsilon_{t^*} I_m\big)^{-1} h, \tag{56}$$

so $R^l_{i,b}$ maps the local unit ball onto the residual second-moment ellipsoid in the shared environment,

$$\big\{R^l_{i,b} x : \|x\|_2 \le 1\big\} = \Big\{h : \; h^\top \big(C^l_{i,b} + \epsilon_{t^*} I_m\big)^{-1} h \le 1\Big\}. \tag{57}$$

The singular values of $R^l_{i,b}$ are the square roots of the eigenvalues of $C^l_{i,b} + \epsilon_{t^*} I_m$. Local directions mapped onto high-energy residual axes are therefore stretched more strongly in the environment, where neighborhood weights are computed.

G Datasets

G.1 Dataset Description

We conduct node classification, link prediction, and graph classification on a variety of datasets. The statistics of the datasets are summarized in Table 6. The detailed descriptions of these dataset are provided as follows:

(1) For node classification and link prediction:

- **CiteSeer**: The CiteSeer dataset Sen et al. (2008) is a widely used benchmark citation network comprising 3,327 nodes and 9,104 edges, designed for multi-class node classification with 6 categories. In this graph, nodes represent scientific publications, while edges correspond to citation relationships between documents, capturing the underlying structure of the citation network. Each node is assigned a class label that indicates the research topic of the corresponding paper.

- **PubMed**: The PubMed dataset Sen et al. (2008) is a large-scale citation network benchmark comprising 19,717 nodes and 88,648 edges, designed for multi-class node classification with 3 categories. In this graph, nodes represent scientific publications related to diabetes research, while edges denote citation relationships between documents, capturing the structural dependencies within the citation network. Each node is assigned a class label corresponding to the type of diabetes discussed in the paper.
- **CS**: The CS dataset Shchur et al. (2018) is a co-authorship network derived from the Microsoft Academic Graph, comprising 18,333 nodes and 163,788 edges for multi-class node classification with 15 categories. In this graph, nodes represent authors in the field of computer science, while edges indicate co-authorship relationships between them, capturing the collaboration structure of the research community. Each node is assigned a class label corresponding to the primary research field of the author.
- **Physics**: The Physics dataset Shchur et al. (2018) is a co-authorship network also derived from the Microsoft Academic Graph, consisting of 34,493 nodes and 495,924 edges for multi-class node classification with 5 categories. In this graph, nodes denote authors in the field of physics, and edges represent co-authorship relationships, reflecting the collaboration patterns within the physics community. Each node is labeled according to the primary research area of the author.
- **Photo**: The Photo dataset Shchur et al. (2018); McAuley et al. (2015) is an Amazon co-purchase network comprising 7,650 nodes and 238,162 edges for multi-class node classification with 8 categories. In this graph, nodes represent products from the Amazon Photo category, while edges indicate that two products are frequently purchased together, capturing the co-purchasing patterns among items. Each node is assigned a class label corresponding to the product category.
- **Computers**: The Computers dataset Shchur et al. (2018); McAuley et al. (2015) is another Amazon co-purchase network consisting of 13,752 nodes and 491,722 edges for multi-class node classification with 10 categories. In this graph, nodes denote products in the Amazon Computers category, and edges represent co-purchase relationships between items, reflecting user purchasing behavior. Each node is labeled according to its product category.

(2) For graph classification:

- **PROTEINS**: The PROTEINS dataset Dobson & Doig (2003) is a benchmark graph classification dataset consisting of 1,113 protein graphs. In this dataset, each graph represents a protein, where nodes correspond to secondary structure elements (e.g., helices and sheets), and edges indicate spatial or sequential adjacency between these elements, capturing the structural organization of proteins. Each graph is assigned a class label indicating whether the protein belongs to a specific functional class.
- **NCI1**: The NCI1 dataset Wale et al. (2008) is a widely used benchmark for graph classification, consisting of 4,110 molecular graphs. In this dataset, each graph represents a chemical compound, where nodes correspond to atoms and edges denote chemical bonds, capturing the molecular structure. Each graph is labeled according to its activity against non-small cell lung cancer, indicating whether the compound is active or inactive.
- **Mutagenicity**: The Mutagenicity dataset Kazius et al. (2005) is a benchmark graph classification dataset comprising 4,337 molecular graphs. In this dataset, each graph represents a chemical compound, where nodes correspond to atoms and edges denote chemical bonds, capturing the molecular structure. Each graph is labeled according to its mutagenic effect on a biological system, indicating whether the compound is mutagenic or non-mutagenic.
- **FRANKENSTEIN**: The FRANKENSTEIN dataset Orsini et al. (2015) is a molecular graph classification benchmark derived from Mutagenicity, consisting of 4,337 graphs. Nodes correspond to atoms whose symbols are replaced by 780-dimensional MNIST digit images as node attributes, and edges denote chemical bonds. Each graph is labeled according to whether the compound is mutagenic.
- **BBBP**: The BBBP dataset Wu et al. (2018) is a molecular graph classification benchmark consisting of 2,050 compounds. In this dataset, each graph represents a molecule, where nodes correspond to atoms and edges denote chemical bonds, capturing the molecular

structure. Each graph is labeled according to its ability to penetrate the blood–brain barrier, indicating whether the compound is permeable or non-permeable.

- **ogbg-molhiv**: The ogbg-molhiv dataset Hu et al. (2020) is a large-scale molecular graph benchmark from the Open Graph Benchmark (OGB), consisting of 41,127 molecules. In this dataset, each graph represents a molecule, where nodes correspond to atoms and edges denote chemical bonds, capturing the molecular structure. Each graph is labeled according to its ability to inhibit HIV replication, indicating whether the compound is HIV active or inactive.

G.2 Data Processing

For node classification and link prediction, we evaluate on standard benchmarks including citation networks (CiteSeer and PubMed), Amazon co-purchasing graphs (Computers and Photo), and Coauthor networks (CS and Physics). All datasets are preprocessed following PyTorch Geometric [1] , where graphs are converted to undirected forms and node features are row-normalized. To enhance structural awareness, we further construct structural signatures from normalized degree information and Random Walk Structural Encodings (RWSE), which enter the prototype gate and the geometric controller. Specially, we adopt the standard random edge split protocol to construct positive and negative evaluation edges for link prediction. For graph classification, we consider bioinformatics datasets from TUDataset (e.g., PROTEINS and NCI1) and molecular graphs (e.g., BBBP and ogbg-molhiv).

H Baselines

H.1 Baseline Description

In this part, we introduce the details of the compared baselines as follows:

(1) **General Graph Neural Networks (GNNs).** We compare GeoF with seven general GNNs:

- **GCN** Kipf & Welling (2017): GCN is a graph neural network that propagates and transforms node features through normalized neighborhood aggregation, enabling the learning of expressive node representations via layer-wise message passing and smoothing.
- **GIN** Xu et al. (2019): GIN is a graph neural network that employs injective aggregation functions to maximally preserve structural information, enabling the learning of highly expressive node representations with discriminative power comparable to the Weisfeiler–Lehman test.
- **ML^2-GCL** Liang et al. (2025): ML^2-GCL is a graph contrastive learning framework that leverages manifold learning principles to construct lightweight augmentations and objectives, enabling effective representation learning by preserving intrinsic geometric structures in graph data.
- **AMPs** Errica et al. (2025): AMPs is a graph neural network framework that adaptively controls message passing to balance information flow, enabling effective mitigation of oversmoothing, oversquashing, and underreaching in deep graph models.
- **WaveGC** Liu et al. (2025): WaveGC is a graph neural network that employs spectral wavelet convolutions via Chebyshev order decomposition, enabling multi-scale feature extraction and effective representation learning over graph structures.
- **SPARROW** Lin et al. (2025): SPARROW is a graph contrastive learning model that eliminates explicit data augmentation by leveraging intrinsic structural signals, enabling effective representation learning through a simplified and efficient contrastive objective.
- **G^{2}Former** Zhang et al. (2025): G^{2}Former is a graph neural network that integrates restricted global-aware graph filters to bridge GNNs and Transformers, enabling expressive node representation learning by capturing both local and global dependencies.

(2) **Manifold-based GNNs.** We compare GeoF with three Manifold-based GNNs:

[1] https://pyg.org/

- **HGCN** Chami et al. (2019): HGCN is a graph neural network that operates in hyperbolic space, enabling the learning of hierarchical node representations by performing message passing under non-Euclidean geometry.
- **D-GCN** Sun et al. (2024): D-GCN is a graph neural network that integrates motif-aware Riemannian representations with generative-contrastive learning, enabling expressive node embeddings by capturing higher-order structures and non-Euclidean geometry.
- **SPDGNN** Wang & Chang (2025): SPDGNN is a graph neural network that operates on symmetric positive definite (SPD) manifolds using Cholesky decomposition, enabling stable and expressive representation learning by preserving the geometric structure of covariance features.

(3) **Adaptive GNNs.** We compare GeoF with four adaptive GNNs:

- **ACE-HGNN** Fu et al. (2021):ACE-HGNN is a geometric graph neural network that adaptively explores manifold curvature, enabling flexible representation learning by capturing heterogeneous geometric structures across graph data.
- **BEC-GNN** Hevapathige et al. (2025): BEC-GNN is a graph neural network that leverages learnable Bakry–Émery curvature to adapt message passing depth, enabling flexible representation learning by dynamically controlling information propagation across graph structures.
- **GNRF** Chen et al. (2025): GNRF is a graph neural network that leverages Ricci flow to evolve node features from a curvature perspective, enabling adaptive information propagation by dynamically reshaping the underlying graph geometry.
- **ARGNN** Wang et al. (2026): ARGNN is a graph neural network that learns anisotropic node-wise Riemannian metric tensors from node features and neighborhood means, enabling geometry-adaptive message passing by tailoring the local metric to each node.

H.2 Implementation Details

We implement GeoF and all baselines in PyTorch[2] and conduct all experiments on NVIDIA A100 GPUs. For the baselines, we use the official implementations released by the authors when available, and otherwise implement them following the original papers; their hyperparameters follow the settings reported in the corresponding papers. GeoF is trained with Adam using a learning rate of 1×10^{-3} and a weight decay of 5×10^{-4}, with model selection on the validation set. We use $B = 8$ geometric blocks, $K = 4$ geometric prototypes, and $L = 2$ recurrent steps with shared parameters. Structural signatures u_i consist of the normalized degree and random-walk return probabilities. The controller Γ_θ has a single hidden layer and outputs factor matrices $U_{i,b}^l, V_{i,b}^l \in \mathbb{R}^{m\times r}$ of rank $r = 4$. The log-triangular coordinates are clipped to the admissible domain $\mathcal{Z}$ with $a_{\max} = 2.0$, $\ell_{\min} = -5.0$, and $\ell_{\max} = 5.0$. For the geometric update, we set the residual-target mixing weight $\lambda = 0.5$ and the correction strength $\gamma = 0.1$; the neighborhood weighting temperature is $\tau = 1.0$. We report accuracy (ACC) for node classification and ROC-AUC for link prediction; for graph classification, we report ACC on PROTEINS, Mutagenicity, NCI1, and FRANKENSTEIN, and ROC-AUC on BBBP and ogbg-molhiv. Results for node classification and link prediction are averaged over 10 random splits, and graph classification uses 10-fold cross-validation with the mean performance across folds.

I Algorithm

The overall training and inference process of the proposed GeoF is shown in Algorithm 1.

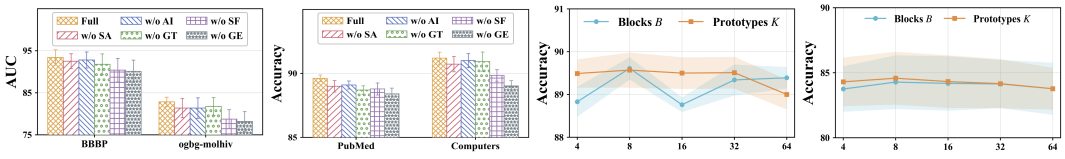
J Complexity Analysis

In this section, we analyze the computational complexity of the proposed GeoF. Let n and $|E|$ denote the numbers of nodes and edges, respectively, K the number of prototypes, L the number of recurrent steps, and d the hidden dimension. We partition the feature space into B blocks of size m such that

[2]https://pytorch.org/

Table 7: Cross-task aggregate comparison across all 18 reported dataset–task settings. **Bold** indicates the best result.

Type	Model	NC Avg.	LP Avg.	GC Avg.	Overall Avg.	Avg. Rank	Avg. Gain	p_{Holm}
General GNNs	GCN	87.7	90.6	76.3	84.9	13.2	+5.7	2.91×10^{-15}
	GIN	87.1	90.0	78.4	85.2	13.3	+5.4	1.82×10^{-15}
	ML^2-GCL	87.9	96.1	80.6	88.2	9.4	+2.4	1.83×10^{-7}
	AMPs	89.8	96.1	81.9	89.3	5.0	+1.3	2.07×10^{-2}
	WaveGC	89.7	96.6	81.7	89.4	4.4	+1.2	4.17×10^{-2}
	SPARROW	87.8	96.5	81.5	88.6	7.6	+1.9	6.45×10^{-5}
	G^{2}Former	89.2	96.5	81.6	89.1	5.3	+1.5	1.55×10^{-2}
Manifold GNNs	HGCN	86.5	92.2	78.4	85.7	13.3	+4.9	1.82×10^{-15}
	D-GCN	88.3	94.9	79.9	87.7	10.3	+2.8	6.01×10^{-9}
	SPDGNN	88.4	94.1	80.7	87.7	10.0	+2.8	1.76×10^{-8}
Adaptive GNNs	ACE-HGNN	88.5	96.4	80.4	88.4	8.5	+2.1	3.90×10^{-6}
	BEC-GNN	89.1	96.6	80.0	88.6	7.4	+2.0	1.09×10^{-4}
	GNRF	88.9	96.1	81.4	88.8	6.9	+1.7	3.90×10^{-4}
	ARGNN	90.1	96.6	81.5	89.4	4.4	+1.2	4.17×10^{-2}
	GeoF	**91.3**	**97.6**	**82.8**	**90.5**	**1.0**	–	–



(a) BBBP and ogbg-molhiv (b) PubMed and Computers (c) PubMed (d) Mutagenicity

Figure 2: Ablation studies on BBBP and ogbg-molhiv in (a), and PubMed and Computers in (b); sensitivity to geometric blocks B and atlas prototypes K on PubMed in (c) and Mutagenicity in (d).

$d = Bm$. The prototype atlas initialization incurs a complexity of $\mathcal{O}(n \cdot K \cdot d \cdot m)$. During the L recurrent steps, edge-level message passing performs batched triangular solves and residual outer products on block-diagonal frames with a cost of $\mathcal{O}(L \cdot |E| \cdot d \cdot m)$, while node-level geometric updates require stabilized Cholesky factorizations of $m \times m$ matrices, resulting in $\mathcal{O}(L \cdot n \cdot d \cdot m^2)$. Together with the $\mathcal{O}(L \cdot n \cdot d^2)$ cost of the shared feature transformations, the overall computational complexity simplifies to $\mathcal{O}(L \cdot d \cdot m \cdot |E| + L \cdot d \cdot (d + m^2) \cdot n)$. In particular, the block-diagonal parameterization reduces the node-wise geometric operations from $\mathcal{O}(d^3)$ to $\mathcal{O}(d \cdot m^2)$, keeping the cost of GeoF linear in the numbers of nodes and edges.

K More Experimental Results

K.1 Cross-Task Aggregate Analysis

We further summarize the cross-task aggregate results in Table 7. Each dataset–task pair is treated as one evaluation setting. NC Avg., LP Avg., and GC Avg. report the average performance within the corresponding task category, while Overall Avg. aggregates all settings. Avg. Rank is computed by ranking the methods within each setting and then averaging their ranks, whereas Avg. Gain measures the mean percentage-point improvement of GeoF over each baseline. The final column reports the Holm-adjusted post-hoc p-value derived from the average-rank differences following the global Friedman test, which yields $\chi^2 = 169.2$ with 14 degrees of freedom and rejects the hypothesis of equal performance across methods.

GeoF achieves the strongest average within every task category and overall, together with an average rank of 1.0, since it ranks first in all 18 settings. Its gains over the baselines are positive throughout and grow from the strongest general and adaptive GNNs to the manifold-based and classical GNNs, and all 14 pairwise comparisons are significant at the 0.05 level after Holm correction. The closest baselines, WaveGC and ARGNN, share the smallest rank difference, and their separation from GeoF remains significant. Among the baseline families, adaptive GNNs and the strongest general GNNs

Table 8: Time consumption of different methods in the training stage for each epoch (in seconds).

Methods	PubMed	CS	Computers	NCI1	Mutagenicity	ogbg-molhiv
GCN	0.0086	0.0167	0.0153	0.2094	0.2312	2.1633
AMPs	0.0393	0.0478	0.0761	0.3170	0.3173	2.8410
G^2Former	0.0637	0.0739	0.0640	0.2011	0.2173	2.2996
SPDGNN	0.0277	0.0321	0.0283	0.2073	0.2287	2.3973
ARGNN	0.0902	0.1573	0.4328	0.2437	0.3383	2.5373
GeoF	0.0683	0.1207	0.1810	0.4391	0.5357	6.0456

Table 9: GPU memory consumption of different methods in the training stage (in GB).

Methods	PubMed	CS	Computers	NCI1	Mutagenicity	ogbg-molhiv
GCN	0.7	1.3	1.3	0.8	0.8	0.9
AMPs	2.7	4.9	5.0	1.3	1.6	2.0
G^2Former	3.3	3.2	3.0	1.5	1.5	1.4
SPDGNN	1.3	2.9	1.9	0.9	1.0	0.9
ARGNN	4.8	8.3	21.3	2.7	2.6	3.8
GeoF	6.4	10.9	15.2	3.5	3.0	4.8

form the leading group, whereas manifold-based GNNs trail on graph classification in particular. Overall, the gains are distributed across tasks and datasets rather than concentrated in a few favorable settings.

K.2 More ablation study

We further extend the ablation study to PubMed and Computers for node classification and to BBBP and ogbg-molhiv for graph classification, covering different tasks and graph scales. As shown in Fig. 2(a) and (b), the full model achieves the best performance on all four datasets. Freezing the geometric state (w/o GE) causes the largest degradation on every dataset, with more pronounced drops on the molecular benchmarks. Removing second-order feedback (w/o SF) causes the next largest drop on BBBP, ogbg-molhiv, and Computers, whereas on PubMed it is comparable to disabling frame transport (w/o GT). This ordering matches the main ablation and highlights the benefit of adapting propagation geometry during message passing and the contribution of residual statistics beyond task-supervised corrections. Disabling frame transport, removing structural signatures (w/o SA), and removing atlas initialization (w/o AI) yield smaller, dataset-dependent decreases, indicating that these components provide complementary improvements.

K.3 More sensitivity study

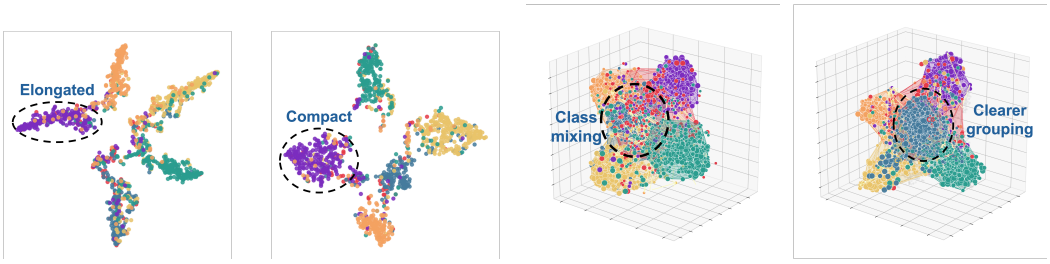
To assess whether GeoF relies on narrowly tuned geometric capacity, we further vary the number of geometric blocks B and atlas prototypes K on PubMed and Mutagenicity. As shown in Fig. 2(c) and (d), performance remains within a relatively narrow range across moderate values of both hyperparameters. On PubMed, varying B produces modest non-monotonic fluctuations, while performance remains stable over a broad range of K before decreasing at the largest setting. Mutagenicity exhibits an even flatter profile for both B and K, followed by a mild decline under excessive capacity. Overall, increasing geometric granularity or atlas size does not yield systematic improvements, indicating that moderate block partitioning and a compact prototype atlas are sufficient for stable performance across different task settings.

K.4 Efficiency and Resource Consumption Analysis

We further evaluate the training efficiency of different methods in terms of per-epoch training time and GPU memory consumption. As shown in Tables 8 and 9, on the three node-classification datasets, GeoF trains faster than ARGNN, and the advantage grows with graph size, reaching its largest margin on Computers, while lightweight GNNs such as GCN and SPDGNN remain the cheapest. On the

graph-classification datasets, GeoF requires more time than ARGNN and GCN, and the gap widens with the number of graphs, peaking on ogbg-molhiv. GPU memory consumption follows the same pattern: GeoF uses more memory than the compared baselines on most datasets, while remaining below ARGNN on Computers, and its memory on the graph-classification datasets stays close to that of ARGNN. The additional cost comes from the edge-level transport and residual outer products and the node-level factorizations, which are computed at every recurrent step. Across all datasets, the overhead relative to ARGNN remains a moderate constant factor rather than growing with graph size.

L VISUALIZATION



(a) ARGNN (b) GeoF (c) Correction Off (d) Correction On

Figure 3: (a), (b) show t-SNE visualizations of node representations learned by ARGNN and GeoF. (c), (d) compare representations with the learned geometric correction disabled and enabled.

Figure 3(a), (b) visualize node representations from ARGNN and GeoF on CiteSeer using 2D t-SNE. The highlighted class cluster is elongated under ARGNN and more compact under GeoF, as indicated by the dashed ellipses. To examine the effect of geometric correction, Fig. 3(c), (d) compare representations from the same checkpoint with the correction disabled ($\gamma = 0$) and enabled ($\gamma = 0.1$). Both sets are visualized in a shared 3D t-SNE projection, with marker size encoding $\log\det(L_i)$. Enabling the correction yields clearer class grouping in the highlighted region, providing qualitative support for its role in shaping task-relevant representations.

Algorithm 1 Training and inference of GeoF

Require: Training graph(s) with task-specific supervision, test inputs, hidden dimension $d = Bm$, number of prototypes K, recurrent steps L, temperature τ, update weights (λ, γ), spectral regularization ϵ_s, and admissible coordinate domain $\mathcal{Z}$.

Ensure: Test predictions $\{p_i\}$, $\{p_G\}$, or $\{p_{uv}\}$ for node classification, graph classification, or link prediction, respectively.

1: **Stage 1: Structural Encoding and Model Initialization**
2: Construct fixed structural signatures U from normalized degree and random-walk return probabilities for each input graph.
3: Initialize the prototype atlas, feature projection, structural gate, message transformation, feature gate, controller Γ_θ, and task-specific readout.
4: Share all propagation and controller parameters across the L recurrent steps.
5: **Stage 2: End-to-End Training**
6: **while** not converged **do**
7: Select training input(s) and supervision for the current task.
8: **for** each input graph $G = (V, E, X)$ **do**
9: Initialize ξ_i^0, α_i, and $z_{i,b}^0$ for all nodes i and blocks b (Eq. (3)).
10: Reconstruct $L_{i,b}^0$ and assemble L_i^0 (Eqs. (1) and (2)).
11: **for** $l = 0, \ldots, L-1$ **do**
12: *i. Triangular Frame Transport*
13: Compute h_i^l, e_{ij}^l, and ω_{ij}^l for $j \in \widetilde{\mathcal{N}}(i)$ (Eq. (4)).
14: Transform features with $\phi(\xi_i^l) = W_\phi \xi_i^l$ and obtain aligned messages $m_{j\to i,b}^l$ by triangular solves (Eq. (5)).
15: Concatenate blockwise messages and compute the aligned aggregate $\bar{\xi}_i^l$ (Eq. (6)).
16: Compute the feature gate r_i^l and update ξ_i^{l+1} (Eq. (7)).
17: *ii. Second-Order Residual Feedback*
18: Compute the residuals $\delta_{ij,b}^l = m_{j\to i,b}^l - (\phi(\xi_i^l))_b$ and accumulate the regularized second-moment matrix $C_{i,b}^l$ with the same weights ω_{ij}^l (Eq. (8)).
19: Compute $R_{i,b}^l = \mathrm{SChol}(C_{i,b}^l)$ and construct the log-triangular target $\widehat{z}_{i,b}^l$ (Eq. (10)).
20: *iii. Task-Guided Geometry Evolution*
21: Form $q_{i,b}^l = [(\xi_i^l)_b \| (\bar{\xi}_i^l)_b \| u_i]$, obtain $U_{i,b}^l$, $V_{i,b}^l$, and $d_{i,b}^l$ from Γ_θ, and construct the geometric correction $\Delta z_{i,b}^l$ (Eq. (11)).
22: Update $z_{i,b}^{l+1} = \Pi_{\mathcal{Z}}\big((1-\lambda) z_{i,b}^l + \lambda \widehat{z}_{i,b}^l + \gamma \Delta z_{i,b}^l\big)$ (Eq. (12)).
23: Reconstruct $L_{i,b}^{l+1}$ and assemble L_i^{l+1} (Eqs. (1) and (2)).
24: **end for**
25: Obtain the final environment representations $h_i^L = L_i^L \xi_i^L$.
26: Compute p_i, p_G, or p_{uv} using the corresponding task-specific readout (Eqs. (13), (14), and (15)).
27: **end for**
28: Evaluate $\mathcal{L}_{\mathrm{node}}$, $\mathcal{L}_{\mathrm{graph}}$, or $\mathcal{L}_{\mathrm{link}}$ for the selected task using its training supervision.
29: Backpropagate through the L recurrent steps and jointly update all trainable parameters.
30: **end while**
31: **Stage 3: Inference**
32: Fix the trained model parameters.
33: **for** each test input graph G **do**
34: Apply the initialization, recurrent evolution, and readout in lines 9–26 using the graph's structural signatures U.
35: Retain p_i for test nodes, p_G for test graphs, or p_{uv} for test candidate pairs, according to the task.
36: **end for**
37: **return** $\{p_i\}$, $\{p_G\}$, or $\{p_{uv}\}$ for the corresponding test inputs.