# BEYOND DISTANCE: RELIABILITY-GATED FISHER–RAO ROUTING FOR HETEROPHILIC GRAPHS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Heterophilic graphs connect nodes with different labels, yet nodes of the same class can retain characteristic neighborhood label distributions (NLDs). Prior methods treat NLDs mainly as Euclidean statistics, auxiliary targets, or pairwise similarity signals. We instead place them on the probability simplex and separate the radial magnitude of a Fisher–Rao displacement from its tangent direction. When distributions become close along different local rays, distance can shrink while normalized direction remains distinguishable. This observation motivates the Fisher–Rao neighborhood-label-distribution network (FR-NLD). For every neighbor, target, and candidate class, FR-NLD compares the target and class prototype in the Fisher tangent space anchored at that neighbor. Reliability gates use angular evidence only when both displacements are long enough and otherwise fall back to radial evidence. In dynamic-routing configurations, degree-aware NLD estimation regularizes finite neighborhoods, while residual updates and belief refresh support repeated geometric inference. Across five modern heterophily benchmarks and their ten public splits, FR-NLD achieves $99.116\%$ ROC-AUC on Minesweeper, while its validation-selected deployment variants lead the ranked comparators on Roman-empire, Tolokers, and Questions and rank third on Amazon-ratings. On the same hardware, the Minesweeper model improves reproduced Polynormer by $1.835$ points while using only $21\%$ of its parameters and $23\%$ of its peak memory. Matched compact-model ablations across all five datasets show that viewer shuffling and inference-time angle removal lower fitted-model performance; the full geometric route outperforms its retrained nongeometric scaffold on four datasets.

## 1 INTRODUCTION

Graph neural networks are often introduced through a local consensus principle: connected nodes exchange features, and repeated aggregation produces representations that are easier to classify (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018). This picture is incomplete on heterophilic graphs. A product can be co-purchased with products from another category, a word can depend syntactically on a word with a different semantic role, and users with different behaviours can interact systematically. In these graphs, a neighbor's label need not match the target label. What matters is the *pattern* of labels around a node.

Neighborhood label distributions (NLDs) make this pattern explicit. Let $C \geq 2$ be the number of classes. Each node $v$ carries a soft class belief $p_v = (p_{v,1}, \ldots, p_{v,C}) \in \mathbb{R}^C$ satisfying $p_{v,k} \geq 0$ and $\sum_{k=1}^{C} p_{v,k} = 1$. For a center node $u$, let $\mathcal{N}(u)$ denote the nodes that interact with $u$. Its NLD is the average belief of these neighbors:

$$r_u = \frac{1}{|\mathcal{N}(u)|} \sum_{v \in \mathcal{N}(u)} p_v.$$

This introductory average assumes $|\mathcal{N}(u)| > 0$; the dynamic-routing estimator in Section C gives an explicit prior-only value for isolated nodes. Nodes can therefore be compared through distributions even when their individual neighbors are heterophilic. This idea underlies recent theory and label-guided graph networks (Wang et al., 2024; Yang et al., 2025). Yet an NLD is not an unconstrained Euclidean vector. It is a probability distribution, and its coordinates are coupled by positivity and unit



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.

mass. Ambient distance and cosine can provide useful scalar comparisons, but they do not define a direction intrinsic to the simplex or specify the point from which that direction is observed. We therefore ask not only *how far apart* two neighborhood distributions are, but also *toward which class-relative direction* one lies when viewed from a particular neighbor.

The distinction becomes important in deep models. If class-conditional neighborhood distributions approach a common interior point along different local rays, their radial separation becomes small while the normalized directions of their nonzero tangent displacements can remain different. Distance and direction therefore describe complementary aspects of neighborhood-distribution geometry. This local picture motivates a router that measures both quantities rather than treating a small distance as evidence that all class structure has disappeared.

We introduce FR-NLD, a standalone Fisher–Rao neighborhood-distribution network. A feature encoder first produces soft labels. The model then estimates smoothed NLDs, constructs class prototypes from training nodes only, and lets every neighbor act as a geometric viewer of the target. For interacting nodes $v \in \mathcal{N}(u)$ and candidate class $k$, the model compares $\mathrm{Log}_{r_v} r_u$ with $\mathrm{Log}_{r_v} \mu_k$ in the tangent space at the viewer distribution $r_v$. This neighbor-anchored construction preserves direction without assuming that $u$ and $v$ share a class. A distance-dependent mixture suppresses the angle when either log vector becomes too short, allowing same-point or prototype-distance evidence to take over. In the dynamic-routing path, residual blocks update hidden states, periodically refresh soft labels and NLDs, and combine information across depth.

In the dynamic-routing configurations, finite neighborhoods introduce a second problem: an empirical distribution based on two neighbors should not be trusted as much as one based on two hundred. We address this with a train-only empirical-Bayes prior. Its total strength depends on routing depth, its class composition interpolates between the global class prior and a soft-label-conditioned mixture of class prototypes, and the same degree-to-prior ratio controls how much Fisher–Rao message reaches the node. Isolated nodes receive no geometric message. These choices turn uncertainty in the estimated distribution into an explicit routing reliability rather than leaving the network to infer it from noisy counts.

Our experiments use the five datasets introduced specifically to correct weaknesses in earlier heterophily evaluation: Roman-empire, Amazon-ratings, Minesweeper, Tolokers, and Questions (Platonov et al., 2023). We use all ten public splits and screen candidate configurations on validation splits. The evaluation asks three questions. First, is geometric routing competitive? The single-router model has the highest mean on Minesweeper; its auxiliary-readout variant leads our ranked comparators on Roman-empire, Tolokers, and Questions and ranks third on Amazon-ratings. Second, what does it cost? The selected Minesweeper and Questions models use $21\%$ and $17\%$ as many parameters as reproduced Polynormer, respectively, while the Tolokers model completes where the official Polynormer configuration is out of memory. Third, what does geometric routing contribute? Matched compact-model ablations across all five datasets compare the full route with a retrained nongeometric scaffold and test viewer identity and angular evidence in fitted models. The Amazon-ratings and Questions results also show how feature-path choices and training-only distillation complement the geometric evidence.

Our contributions are:

- **Viewer-conditioned Fisher–Rao evidence.** We represent an edge–class relation by a neighbor viewer, a target NLD, and a class prototype. Comparing the target and prototype in the viewer's Fisher tangent space makes direction class-specific and observer-dependent, while radial magnitude identifies when that direction is reliable.
- **A reliability-gated geometric network.** We turn this principle into standalone models that mix angular and radial experts and produce signed class-conditioned messages. Dynamic-routing configurations additionally regularize finite-degree NLDs and support repeated inference through residual routing and belief refresh.
- **Benchmark, efficiency, and mechanistic evidence.** Ten-split evaluation covers all five modern heterophily benchmarks. Same-hardware measurements expose the parameter and memory profile, while matched compact-model ablations test the contribution of geometric routing across five datasets. The single-model FR-NLD leads the ranked comparators on Minesweeper, and FR-NLD-Pro leads on Roman-empire, Tolokers, and Questions. On

Minesweeper, the single model uses 79% fewer parameters and 77% less peak memory than reproduced Polynormer.

## 2 RELATED WORK

**Local architectures for heterophily.** Classical graph convolutional network (GCN), GraphSAGE and graph attention network (GAT) layers aggregate neighboring features and therefore inherit a local smoothing bias (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018). Recent surveys organize the resulting heterophilic graph-learning literature and clarify how architectural, propagation, and data-centric remedies relate to one another (Zheng et al., 2026). Heterophily-oriented architectures modify either the propagated signal or the channels through which it is mixed. $H_2$GCN separates ego and neighbor representations, exposes higher-order neighborhoods, and combines multiple depths (Zhu et al., 2020); FAGCN and GPR-GNN preserve high-frequency information through signed or learned propagation responses (Bo et al., 2021; Chien et al., 2021); and ACM-GCN adaptively mixes low-pass, high-pass, and identity channels (Luan et al., 2022). LINKX decouples adjacency and feature processing, whereas GloGNN learns a global aggregation mechanism (Lim et al., 2021; Li et al., 2022). These approaches mainly redesign *what* is aggregated or *which frequency* is retained. FR-NLD instead asks how a target neighborhood distribution moves, relative to a class prototype, from the viewpoint of the sending neighbor.

**Global, multiscale, and directed propagation.** Several recent models move beyond a fixed local receptive field. Ordered GNN assigns successive hops to ordered representation blocks so that depth need not indiscriminately overwrite earlier information (Song et al., 2023). Dir-GNN treats incoming and outgoing neighborhoods separately and shows that edge direction can increase effective homophily (Rossi et al., 2024). Polynormer combines local propagation with linear global attention and polynomial expressivity (Deng et al., 2024), while Wideformer partitions global aggregation and directs attention toward informative source clusters (Sun et al., 2025). The Bundle Neural Network (BuNN) takes a different route by diffusing messages over flat vector bundles (Bamberger et al., 2025). These are important strong references, but graph direction, hop selection, and global token mixing are distinct from our notion of direction: ours is an intrinsic tangent direction between probability distributions and is conditioned jointly on viewer, target, and candidate class.

Graph framelets provide a complementary multiscale route. PEGFAN constructs sparse, permutation-equivariant Haar-type framelets to extract heterophilic patterns beyond one hop (Li et al., 2024), while high-variance graph framelets explicitly seek representations compatible with nonsmooth label signals (Zheng et al., 2025). These methods decompose graph signals across spectral or multiscale channels; FR-NLD instead performs class-conditioned routing on the probability simplex.

**Neighborhood distributions and compatibility.** In a heterophilic graph, the class composition of a neighborhood may be more stable than any individual edge. Compatibility-based reasoning summarizes class-to-class transition patterns, and SoftGNN predicts soft labels and uses them to learn aggregation weights (Yang et al., 2025). Under the heterophilic stochastic block model, Wang et al. (2024) relate class separability to the distance between neighborhood label distributions (NLDs), together with node degree, and show that normalized NLD separation can remain informative during deep propagation. Our work starts from the same distributional object but changes the representation of evidence: NLDs are points on a statistical manifold, training labels form leakage-free class prototypes, and finite degree controls the reliability of the resulting geometric prior. Consequently, two targets at comparable distance from a viewer can still support different classes because their normalized tangent directions differ.

**Graph geometry and rewiring.** Geom-GCN constructs latent geometric neighborhoods before aggregation (Pei et al., 2020), while graph rewiring changes which neighbors interact. DHGR uses angular comparisons of multi-hop neighborhood distributions to form a two-point similarity for one-time edge rewiring (Bi et al., 2024). FR-NLD instead retains the original edges and evaluates a viewer–target–class relation within each routing layer: it compares $\mathrm{Log}_{r_v} r_u$ and $\mathrm{Log}_{r_v} \mu_k$ in the Fisher tangent space at the neighbor viewer $r_v$. This yields a class-conditioned direction rather than a pairwise similarity score.

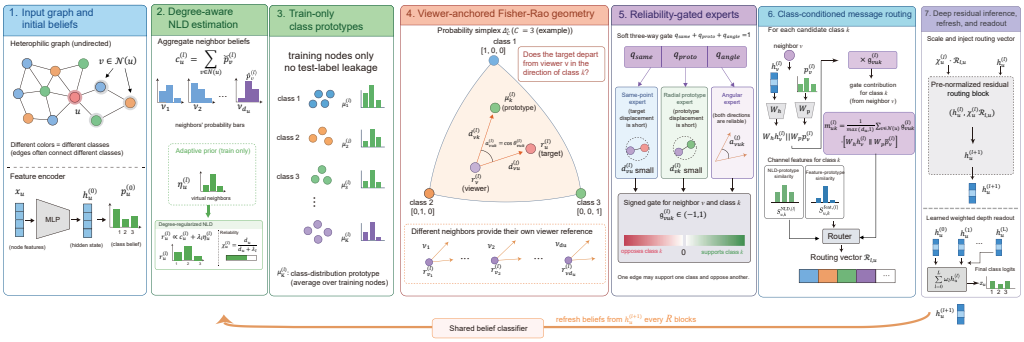



Figure 1: **Overview of FR-NLD.** In the dynamic-routing path shown, soft node beliefs form degree-regularized neighborhood label distributions (NLDs) and train-only class prototypes. Each neighbor then supplies a viewer-specific Fisher–Rao relation between the target NLD and every prototype. Smooth gates turn this evidence into class-conditioned messages for residual routing; the beliefs and NLDs are periodically refreshed across depth. The diagram depicts the dynamic-routing path; Amazon-ratings uses feature–geometry fusion instead. One prototype per class is shown for clarity; the integer hyperparameter $M$ may select a small train-only prototype bank. The figure shows uniform NLD aggregation; an optional feature-conditioned weighting is defined in Appendix C.2.

**Information geometry.** Information geometry equips a statistical model with the Fisher metric, whose local quadratic form captures distributional distinguishability and is invariant to smooth reparameterization (Rao, 1945; Amari, 1998; 2016). For categorical distributions, the square-root map identifies the simplex with the positive orthant of a sphere and provides closed-form distances, exponential maps, and logarithmic maps. Most heterophily methods use Euclidean feature distances, ambient cosine similarities, or scalar divergences. We instead transport each viewer–target and viewer–prototype relation to the same viewer-specific tangent space. This common base point is what makes the angle observer-conditioned and class-specific, and it is the central geometric distinction of FR-NLD.

**Depth and over-smoothing.** Repeated propagation can contract node representations toward a low-dimensional or indistinguishable limit (Oono & Suzuki, 2020). Existing remedies preserve pairwise spread, thin the propagation graph, inject initial representations, or read out several depths (Zhao & Akoglu, 2020; Rong et al., 2020; Chen et al., 2020; Xu et al., 2018). Ordered GNN additionally assigns different hops to different feature blocks (Song et al., 2023), while stability analyses relate deep-GCN behavior to both depth and graph-filter spectra (Yang et al., 2026). In FR-NLD, residual states and a learned cross-depth readout support repeated geometric inference; our depth ablation treats this as an architectural property rather than a separate geometric claim.

## 3 Geometry and Model

Let $G = (V, E)$ be a finite undirected graph, where $V$ is its node set and $E$ contains unordered node pairs. For node $u \in V$, let $\mathcal{N}(u) := \{v \in V : \{u, v\} \in E\}$ and $d_u := |\mathcal{N}(u)|$. Let $L \in \mathbb{N}$ be the number of routing blocks, and let $l \in \{0, \dots, L\}$ index routing depth. Node $u$ has features $x_u \in \mathbb{R}^F$, label $y_u \in \{1, \dots, C\}$, hidden state $h_u^{(l)} \in \mathbb{R}^D$, and soft class belief $p_u^{(l)}$ in the interior probability simplex $\Delta_C^\circ := \{p \in \mathbb{R}_{>0}^C : \sum_{i=1}^C p_i = 1\}$. Appendix A gives the full notation and the connection between NLDs and the class-compatibility matrix.

### 3.1 FISHER–RAO NEIGHBORHOOD EVIDENCE

For $p, q \in \Delta_C^\circ$, let $T_p\Delta_C^\circ := \{\xi \in \mathbb{R}^C : \sum_{i=1}^C \xi_i = 0\}$ denote the tangent space at $p$. For $\xi, \xi' \in T_p\Delta_C^\circ$, the Fisher–Rao metric, induced norm, distance, and tangent cosine are

$$\begin{aligned} g_p(\xi, \xi') &:= \sum_{i=1}^{C} \frac{\xi_i \xi_i'}{p_i}, \qquad \|\xi\|_p := \sqrt{g_p(\xi, \xi)}, \\ d_{\mathrm{FR}}(p, q) &:= 2 \arccos\left( \sum_{i=1}^{C} \sqrt{p_i q_i} \right), \qquad \cos_p^{\mathrm{FR}}(\xi, \xi') := \frac{g_p(\xi, \xi')}{\|\xi\|_p \|\xi'\|_p}. \end{aligned} \tag{1}$$

The last expression applies to nonzero tangent vectors. The Riemannian log map $\mathrm{Log}_p(q) \in T_p\Delta_C^\circ$ points from $p$ toward $q$ and satisfies $\|\mathrm{Log}_p(q)\|_p = d_{\mathrm{FR}}(p, q)$; Appendix A.1 gives its closed form and the zero-displacement convention.

Radial magnitude and tangent direction need not disappear together: normalized log-map directions can remain distinguishable even when distributions become close. This motivates retaining angular evidence while suppressing it whenever a displacement is too short to normalize reliably.

An encoder produces $h_u^{(0)}$ and an initial belief $p_u^{(0)}$. Its optional input channels are controlled by validation-selected switches specified in Appendix C.2. At depth $l$, let $\breve{p}_u^{(l)} \in \Delta_C^\circ$ be the belief supplied to the NLD estimator; $\breve{p}_u^{(l)} = p_u^{(l)}$ when the geometry-balancing exponent is zero, while a positive exponent activates the imbalance correction in Appendix C. For the dynamic-routing configurations, we estimate degree-regularized NLDs and their routing reliability as follows. Let $V_{\mathrm{tr},k}$ be the training nodes of class $k$ and $n_k := |V_{\mathrm{tr},k}|$. With uniform distribution $\mathbf{u}_C := (1/C, \ldots, 1/C)$, smoothing $\varepsilon_{\mathrm{sm}} \in (0, 1)$, train-only prior centre $\eta_u^{(l)} \in \Delta_C^\circ$, and prior pseudocount $\lambda_l > 0$, define

$$\begin{aligned} c_u^{(l)} &:= \sum_{v \in \mathcal{N}(u)} \breve{p}_v^{(l)}, \quad r_u^{(l)} := (1 - \varepsilon_{\mathrm{sm}}) \frac{c_u^{(l)} + \lambda_l \eta_u^{(l)}}{d_u + \lambda_l} + \varepsilon_{\mathrm{sm}} \mathbf{u}_C, \\ \chi_u^{(l)} &:= \frac{d_u}{d_u + \lambda_l}, \qquad \mu_k^{(l)} := \frac{1}{n_k} \sum_{u \in V_{\mathrm{tr},k}} r_u^{(l)}. \end{aligned} \tag{2}$$

Thus $r_u^{(l)}$ is a degree-regularized NLD, $\chi_u^{(l)} \in [0, 1)$ measures the fraction of its effective count supplied by observed neighbors, and $\mu_k^{(l)}$ is the train-only NLD prototype of class $k$. Appendix C specifies the adaptive prior centre, depth-dependent pseudocount, and imbalance correction.

For $v \in \mathcal{N}(u)$ and candidate class $k$, define

$$\mathfrak{a}_{vuk}^{(l)} := \cos_{r_v^{(l)}}^{\mathrm{FR}} \left( \mathrm{Log}_{r_v^{(l)}} r_u^{(l)}, \mathrm{Log}_{r_v^{(l)}} \mu_k^{(l)} \right), \quad d_{vu}^{(l)} := d_{\mathrm{FR}}(r_v^{(l)}, r_u^{(l)}), \quad d_{vk}^{(l)} := d_{\mathrm{FR}}(r_v^{(l)}, \mu_k^{(l)}). \tag{3}$$

The angle cosine $\mathfrak{a}_{vuk}^{(l)}$ measures whether the target NLD and class-$k$ prototype depart from the neighbor's NLD in similar Fisher–Rao directions; the two distances provide the corresponding radial evidence.

### 3.2 RELIABILITY-GATED GEOMETRIC ROUTING

The model uses angle only when both displacements are long enough. Let $s_\tau(d) := d^2/(d^2 + \tau^2)$, and abbreviate $s_{\mathrm{tgt}} := s_{\tau_{\mathrm{tgt}}^{(l)}}(d_{vu}^{(l)})$ and $s_{\mathrm{cls}} := s_{\tau_{\mathrm{cls}}^{(l)}}(d_{vk}^{(l)})$ for learned positive thresholds. The regime weights and signed class gate are

$$\begin{aligned} \left(q_{\mathrm{same},vuk}^{(l)}, q_{\mathrm{proto},vuk}^{(l)}, q_{\mathrm{angle},vuk}^{(l)}\right) &:= \left(1 - s_{\mathrm{tgt}},\ s_{\mathrm{tgt}}(1 - s_{\mathrm{cls}}),\ s_{\mathrm{tgt}} s_{\mathrm{cls}}\right), \\ \mathfrak{g}_{vuk}^{(l)} &:= \tanh\left( \sum_{e \in \{\mathrm{same},\mathrm{proto},\mathrm{angle}\}} q_{e,vuk}^{(l)} e_{e,vuk}^{(l)} \right) \in (-1, 1). \end{aligned} \tag{4}$$

Here $e \in \{\text{same}, \text{proto}, \text{angle}\}$ indexes one of the three learned experts, and $e^{(l)}_{e,vuk} \in \mathbb{R}$ is that expert's scalar output for neighbor $v$, target $u$, and candidate class $k$ at depth $l$. Appendix C gives the exact expert inputs and dimensions. The three weights are nonnegative and sum to one. They select same-point evidence when the target displacement is short, radial prototype evidence when only the prototype displacement is short, and angular evidence when both directions are reliable. The resulting class-conditioned message $m^{(l)}_{u,k} \in \mathbb{R}^{2D}$ is

$$m^{(l)}_{u,k} := \frac{1}{\max(d_u, 1)} \sum_{v \in \mathcal{N}(u)} \mathfrak{g}^{(l)}_{vuk} \big[ W^{(l)}_h \overline{h}^{(l)}_v \| W^{(l)}_p \breve{p}^{(l)}_v \big], \tag{5}$$

Here $\overline{h}^{(l)}_v := \mathrm{Norm}^{\mathrm{route}}_l(h^{(l)}_v)$ is the routing-normalized hidden state, $W^{(l)}_h \in \mathbb{R}^{D \times D}$ and $W^{(l)}_p \in \mathbb{R}^{D \times C}$ are learned projections, and $\|$ denotes concatenation. A learned channel mixer combines all $\{m^{(l)}_{u,k}\}_{k=1}^{C}$ with train-only NLD- and feature-prototype scores to obtain $\mathcal{R}_{l,u} \in \mathbb{R}^D$. Appendix C specifies these scores, their training-only standardization, and the mixer dimensions.

### 3.3 Deep residual inference

The dynamic-routing configurations use the following repeated update. For each $l \in \{0, \ldots, L-1\}$, routing block $l$ preserves the incoming hidden state while injecting the reliability-scaled routing vector,

$$h^{(l+1)}_u := \mathrm{ResBlock}_l\Big( h^{(l)}_u, \chi^{(l)}_u \mathcal{R}_{l,u} \Big). \tag{6}$$

Here $\mathrm{ResBlock}_l : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^D$ is a pre-normalized residual router; Appendix C gives its LayerScale, dropout, normalization, and feed-forward operations. Every $R \in \{1, \ldots, L\}$ completed blocks, a shared classifier refreshes the beliefs $p^{(l)}$, after which the NLDs and prototypes are recomputed. Deep configurations may reuse routing parameters across consecutive blocks without reducing the executed depth.

Finally, learned nonnegative weights $\omega_0, \ldots, \omega_L$ summing to one combine the hidden depths:

$$z_u := W_{\mathrm{out}} J_\psi \left( \sum_{l=0}^{L} \omega_l h^{(l)}_u \right) + b_{\mathrm{out}} \in \mathbb{R}^C. \tag{7}$$

Here $J_\psi : \mathbb{R}^D \to \mathbb{R}^D$ is the readout map, $W_{\mathrm{out}} \in \mathbb{R}^{C \times D}$ and $b_{\mathrm{out}} \in \mathbb{R}^C$ form the final affine classifier, and $z_u$ contains its class logits. Appendix C gives the refresh rule, depth-weight parameterization, parameter-sharing schedule, and complete training objective.

## 4 Experiments

All five configurations use train-only class prototypes and neighbor-anchored Fisher–Rao geometric evidence. Roman-empire, Minesweeper, Tolokers, and Questions use dynamic residual routing with periodic belief and NLD refresh. Amazon-ratings instead uses a smoothed neighbor mean and injects this evidence once through feature–geometry fusion, without degree-to-prior shrinkage or periodic refresh. Its richer feature interactions motivate the fusion path, while the class imbalance in Questions motivates geometry-side balancing and training-only distillation; the selected settings are detailed in the appendix. Input encoders, routing settings, and training objectives are selected on validation data. We evaluate on all ten public splits of these datasets (Platonov et al., 2023). We report test mean $\pm$ sample standard deviation. Accuracy is used for the multiclass graphs and area under the receiver operating characteristic curve (ROC-AUC) for the binary graphs. Dataset statistics, principal dataset-varying configurations, the validation-selection protocol, and paired uncertainty summaries appear in Appendix E.

**Model variants.** We use FR-NLD to denote a single geometric model. FR-NLD-Plus combines the class probabilities of three independently trained FR-NLD members to reduce seed sensitivity without changing any member architecture. FR-NLD-Pro can further combine the Plus prediction with an auxiliary GCN readout, exploiting the complementarity between distributional geometry and

conventional feature smoothing when the readout is enabled. Ensemble weights and implementation details are given in Appendix C.2.

Table 1: **Results on the five modern heterophily benchmarks.** Entries are mean ± standard deviation: our rows use ten public splits, while published rows follow their source protocols (Platonov et al., 2023; Luo et al., 2024; Deng et al., 2024; Sun et al., 2025; Bamberger et al., 2025). The superscript $*$ marks five-run tuned results. The published-only † row lacks an executable implementation and is shown for reference, not ranking. Best means are **bold**, second-best means are blue, and third-best means are orange. "-sep" denotes separate center and neighbor feature transforms; "Acc." denotes accuracy. The FR-NLD row reports one deployed geometric router; its configuration is listed in Appendix C.2. FR-NLD-Plus combines three active independently trained members in probability space, and FR-NLD-Pro optionally adds a validation-selected auxiliary GCN readout. Member and readout settings are given in the appendix. "Avg." is the unweighted macro average of the five dataset means and is reported only for complete rows; the adjacent column marks its top four ranked executable methods. Higher is better.

| Method | Roman-empire Acc. | Amazon-ratings Acc. | Minesweeper ROC-AUC | Tolokers ROC-AUC | Questions ROC-AUC | Avg. | Rank |
|---|---|---|---|---|---|---|---|
| ResNet | 65.88 ± 0.38 | 45.90 ± 0.52 | 50.89 ± 1.39 | 72.95 ± 1.06 | 70.34 ± 0.76 | 61.192 | |
| GCN | 73.69 ± 0.74 | 48.70 ± 0.63 | 89.75 ± 0.52 | 83.64 ± 0.67 | 76.09 ± 1.27 | 74.374 | |
| GraphSAGE | 85.74 ± 0.67 | 53.63 ± 0.39 | 93.51 ± 0.57 | 82.43 ± 0.44 | 76.44 ± 0.62 | 78.350 | |
| GAT-sep | 88.75 ± 0.41 | 52.70 ± 0.62 | 93.91 ± 0.35 | 83.78 ± 0.43 | 76.79 ± 0.71 | 79.186 | |
| Graph Transformer | 86.51 ± 0.73 | 51.17 ± 0.66 | 91.85 ± 0.76 | 83.23 ± 0.64 | 77.95 ± 0.68 | 78.142 | |
| Graph Transformer-sep | 87.32 ± 0.39 | 52.18 ± 0.80 | 92.29 ± 0.47 | 82.52 ± 0.92 | 78.05 ± 0.93 | 78.472 | |
| $H_2$GCN | 60.11 ± 0.52 | 36.47 ± 0.23 | 89.71 ± 0.31 | 73.35 ± 1.01 | 63.59 ± 1.46 | 64.646 | |
| GPR-GNN | 64.85 ± 0.27 | 44.88 ± 0.34 | 86.24 ± 0.61 | 72.94 ± 0.97 | 55.48 ± 0.91 | 64.878 | |
| FSGNN | 79.92 ± 0.56 | 52.74 ± 0.83 | 90.08 ± 0.70 | 82.76 ± 0.61 | 78.86 ± 0.92 | 76.872 | |
| Tuned GCN* | 91.27 ± 0.20 | 53.80 ± 0.60 | **97.86 ± 0.24** | – | **79.02 ± 0.60** | – | |
| Tuned GraphSAGE* | 91.06 ± 0.27 | **55.40 ± 0.21** | 97.77 ± 0.62 | – | 77.21 ± 1.28 | – | |
| Tuned GAT* | 90.63 ± 0.14 | **55.54 ± 0.51** | 97.73 ± 0.73 | – | 77.95 ± 0.51 | – | |
| Polynormer-r | **92.55 ± 0.37** | 54.81 ± 0.49 | 97.46 ± 0.36 | 85.91 ± 0.74 | 78.92 ± 0.89 | **81.930** | 3rd |
| Polynormer + Wideformer | 92.16 ± 0.24 | 55.05 ± 0.08 | 97.26 ± 0.01 | 85.33 ± 0.23 | 79.00 ± 0.20 | 81.760 | |
| BuNN (published only)† | 91.75 ± 0.39 | 53.74 ± 0.51 | 98.99 ± 0.16 | 84.78 ± 0.80 | 78.75 ± 1.09 | 81.602 | |
| (Ours) FR-NLD | 91.677 ± 0.416 | 53.975 ± 0.724 | **99.116 ± 0.170** | **85.974 ± 0.739** | 78.405 ± 1.467 | 81.829 | 4th |
| (Ours) FR-NLD-Plus | **92.326 ± 0.354** | 54.155 ± 0.668 | **99.112 ± 0.250** | **86.180 ± 0.801** | **79.058 ± 1.334** | **82.166** | 2nd |
| (Ours) FR-NLD-Pro | **92.916 ± 0.386** | **55.358 ± 0.590** | **99.112 ± 0.250** | **86.269 ± 0.740** | **79.379 ± 1.317** | **82.607** | **1st** |

The single FR-NLD model has the **highest mean** on Minesweeper. FR-NLD-Pro has the highest means on Roman-empire, Tolokers, and Questions among the ranked comparators in Table 1, and ranks third on Amazon-ratings. The Plus readout improves Amazon-ratings from 53.975% to 54.155%, Tolokers from 85.974% to 86.180%, and Questions from 78.405% to 79.058%. On Minesweeper it remains close to the single-model result (99.112% versus 99.116%). Adding the auxiliary readout raises Roman-empire to 92.916%, Amazon-ratings to 55.358%, Tolokers to 86.269%, and Questions to 79.379%; validation leaves it off on Minesweeper. Deployment details appear in Appendix C.2.

Table 2: **Same-hardware resource comparison with reproduced Polynormer.** $\Delta$ is FR-NLD minus Polynormer in percentage points. M is millions of trainable parameters and GB is peak allocated GPU memory. Ratios are FR-NLD/Polynormer. These measurements use the single-model configurations reported in Table 1. OOM means out of memory on a 24GB GPU.

| | Test score (%) | | | Parameters (M) | | | Peak memory (GB) | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | Ours | Poly. | $\Delta$ | Ours | Poly. | Ratio | Ours | Poly. | Ratio |
| Roman-empire | 91.677 | 92.483 | −0.806 | 11.33 | 9.92 | 114% | 6.32 | 6.36 | 99% |
| Amazon-ratings | 53.975 | 54.392 | −0.417 | 14.76 | 9.11 | 162% | 18.15 | 9.13 | 199% |
| Minesweeper | **99.116** | 97.281 | **+1.835** | **2.20** | 10.54 | **21%** | **0.97** | 4.15 | **23%** |
| Tolokers | **85.974** | OOM | – | **0.055** | – | – | **2.56** | > 24 | < 11% |
| Questions | **78.405** | 78.299 | **+0.106** | **1.18** | 6.74 | **17%** | **3.79** | 10.99 | **34%** |

The Minesweeper model improves reproduced Polynormer by 1.835 points while using 79% fewer parameters and 77% less peak memory. Questions uses 83% fewer parameters, and the 0.055M-

parameter Tolokers model completes where the official Polynormer configuration exceeds 24GB. Roman-empire uses a sharing span of four.

Table 3: **Small-model geometric ablations across five datasets.** Each cell reports the paired mean test-score gap over ten official splits for a predeclared eight-step model. A positive value means the baseline scores higher than the intervention by that many percentage points. The first column retrains the matched nongeometric scaffold from the same initial predictor; the remaining interventions use the trained full checkpoint, except for the final angle-zero retraining column. Amazon-ratings uses feature–geometry fusion; the other datasets use dynamic routing.

| | $\Delta$ = baseline − intervention (pp) | | | |
|---|---|---|---|---|
| Dataset | Geometry off, retrained | Shuffle viewer | Zero angle at inference | Zero angle, retrained |
| Roman-empire | +16.832 | +3.060 | +1.509 | +0.236 |
| Amazon-ratings | -0.657 | +0.910 | +0.124 | -0.261 |
| Minesweeper | +40.822 | +1.113 | +0.154 | +0.118 |
| Tolokers | +6.938 | +3.096 | +0.039 | +0.012 |
| Questions | +3.938 | +0.257 | +0.166 | +0.313 |

Removing the complete geometric route produces the main overall comparison; viewer shuffling and angle zeroing then isolate specific geometric signals. Viewer shuffling reduces the fitted models' scores on all five datasets. Zeroing the angle at inference also lowers the score on all five datasets, with a smaller effect. Retraining without angle produces mixed outcomes across datasets. Appendix E.3 reports the complete component path, dynamic-refresh and prior ablations, and the intervention protocol.

As a supporting architecture ablation, the matched depth–residual factorial in Appendix E.3 shows that residual state preservation enables beneficial deep routing on Roman-empire, Minesweeper, Tolokers, and Questions.

## 5 Conclusion

Neighborhood-distribution distance and direction provide complementary evidence on heterophilic graphs. FR-NLD makes that direction observer-conditioned by comparing a target and class prototype in each neighbor's Fisher tangent space, and uses radial magnitude to decide when the angle is reliable. Across five modern heterophily benchmarks, the single-router model attains the highest mean on Minesweeper among the ranked comparators in our table, while its optional auxiliary-readout variant leads on Roman-empire, Tolokers, and Questions. The Minesweeper configuration uses about one fifth of reproduced Polynormer's parameters and one quarter of its peak memory. Compact-model ablations across five datasets show consistent sensitivity to viewer identity and inference-time angular evidence, while the geometric route outperforms a retrained nongeometric scaffold on four datasets. Together, these results support a focused claim: reliability-gated Fisher–Rao direction is a useful signal for heterophilic graph learning.

**Reproducibility and ethics.** All prototypes and priors use training labels only. The anonymous supplementary material provides selected evaluation utilities, final deployment manifests, public split identifiers, and code to recompute the macro averages and ranks in Table 1. The complete training and evaluation implementation will be released upon acceptance. The work uses public benchmarks and introduces no new personal data; relational deployment should nevertheless include subgroup, privacy, and human-oversight review.

## AI Use Statement

Generative AI tools assisted with refining hypotheses and experimental designs, developing and debugging code, orchestrating experiments, interpreting results, drafting mathematical exposition, reviewing relevant literature and checking citations, and editing prose. The authors reviewed the AI-assisted work and take responsibility for the paper's claims, results, citations, and code.

## REFERENCES


Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer, 2016. doi: 10.1007/978-4-431-55978-8.

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

Wendong Bi, Lun Du, Qiang Fu, Yanlin Wang, Shi Han, and Dongmei Zhang. Make heterophily graphs better fit GNN: A graph rewiring approach. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8744–8757, 2024. doi: 10.1109/TKDE.2024.3441766.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021. doi: 10.1609/aaai.v35i5.16514.

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

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hmv1LpNfXa.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Jianfei Li, Ruigang Zheng, Han Feng, Ming Li, and Xiaosheng Zhuang. Permutation equivariant graph framelets for heterophilous graph learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11634–11648, 2024. doi: 10.1109/TNNLS.2024.3370918.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13242–13256, 2022.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20887–20902, 2021.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://openreview.net/forum?id=NjeEfP7e3KZ.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic GNNs are strong baselines: Reassessing GNNs for node classification. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b10ed15ff1aa864f1be3a75f1ffc021b-Abstract-Datasets_and_Benchmarks_Track.html.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

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

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

Calyampudi Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3):81–91, 1945.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M. Bronstein. Edge directionality improves learning on heterophilic graphs. In *Proceedings of the Second Learning on Graphs Conference*, volume 231 of *Proceedings of Machine Learning Research*, pp. 25:1–25:27, 2024. URL https://proceedings.mlr.press/v231/rossi24a.html.

Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Ordered GNN: Ordering message passing to deal with heterophily and over-smoothing. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=wKPmPBHSnT6.

Junshu Sun, Wanxing Chang, Chenxue Yang, Qingming Huang, and Shuhui Wang. Relieving the over-aggregating effect in graph transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0a0e436e8830f746300f592377730fca-Abstract-Conference.html.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Junfu Wang, Yuanfang Guo, Liang Yang, and Yunhong Wang. Understanding heterophily for graph neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50489–50529, 2024. URL https://proceedings.mlr.press/v235/wang24u.html.

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 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462, 2018. URL https://proceedings.mlr.press/v80/xu18c.html.

Guangrui Yang, Ming Li, Han Feng, and Xiaosheng Zhuang. Deeper insights into deep graph convolutional networks: Stability and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1707–1719, 2026. doi: 10.1109/TPAMI.2025.3616350.

Han Yang, Junyuan Fang, Jiajing Wu, Dan Li, Yaonan Wang, and Zibin Zheng. Soft label enhanced graph neural network under heterophily. *Knowledge-Based Systems*, 309:112861, 2025. doi: 10.1016/j.knosys.2024.112861.

Lingxiao Zhao and Leman Akoglu. PairNorm: Tackling oversmoothing in GNNs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkecl1rtwB.

Ruigang Zheng, Ming Li, and Xiaosheng Zhuang. High-variance graph framelets for heterophilous graph learning. In *2025 IEEE International Conference on Knowledge Graph*, pp. 397–403, 2025. doi: 10.1109/ICKG66886.2025.00058.

Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 38(7):4385–4404, 2026. doi: 10.1109/TKDE.2026.3680353.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7793–7804, 2020.

# A FULL FISHER–RAO NEIGHBORHOOD GEOMETRY

Let $G = (V, E)$ be a graph, where $V$ is its node set and $E$ is its edge set. Throughout, $\mathbb{N} := \{1, 2, \ldots\}$, $\mathbb{R}$ is the real line, $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ are its positive and nonnegative parts, and $|A|$ denotes the cardinality of a finite set $A$. The functions $\log$ and $\exp$ are the natural logarithm and exponential. Write $N := |V|$, $F \in \mathbb{N}$ for the input-feature dimension, $C \geq 2$ for the number of classes, and $D \in \mathbb{N}$ for the hidden dimension. We write $u, v \in V$ for nodes and assume that $G$ is undirected, so $E$ consists of unordered pairs $\{u, v\}$. Define $\mathcal{N}(u) := \{v \in V : \{u, v\} \in E\}$ and $d_u := |\mathcal{N}(u)|$ as the neighborhood and degree of $u$. Each node $u$ has a feature vector $x_u \in \mathbb{R}^F$ and a label $y_u \in \{1, \ldots, C\}$; equivalently, $y : V \to \{1, \ldots, C\}$ is the node-label function. A feature encoder $f_\theta : \mathbb{R}^F \to \mathbb{R}^D$, with trainable parameters $\theta$, followed by a trainable affine classifier $(W_0, b_0) \in \mathbb{R}^{C\times D} \times \mathbb{R}^C$, produces

$$h_u^{(0)} = f_\theta(x_u), \qquad z_u^{(0)} = W_0 h_u^{(0)} + b_0, \qquad p_u^{(0)} = \mathrm{softmax}(z_u^{(0)}). \tag{8}$$

Here $h_u^{(0)} \in \mathbb{R}^D$, $z_u^{(0)} \in \mathbb{R}^C$, and $p_u^{(0)} \in \Delta_C^\circ$, with $\mathrm{softmax}(z)_k := \exp(z_k)/\sum_{j=1}^C \exp(z_j)$. The feature-only setting uses the two-layer MLP

$$f_\theta(x) := \mathrm{ReLU}(E_2\,\mathrm{Drop}^{\mathrm{enc}}(\mathrm{ReLU}(E_1 x + c_1)) + c_2)\,, \tag{9}$$

where $E_1 \in \mathbb{R}^{D\times F}$, $c_1, c_2 \in \mathbb{R}^D$, $E_2 \in \mathbb{R}^{D\times D}$, $\theta := (E_1, c_1, E_2, c_2)$, and $\mathrm{ReLU}(t) := \max\{t, 0\}$ is applied coordinatewise. The map $\mathrm{Drop}^{\mathrm{enc}}$ follows the dropout definition given in Section C. The initial NLD is formed from neighboring beliefs; Appendix C.2 specifies the binary encoder switches that may augment this base setting without changing the classifier or router. Section C gives the exact finite-sample estimator.

## A.1 THE CATEGORICAL SIMPLEX

We work on the interior of the $C$-class probability simplex,

$$\Delta_C^\circ = \left\{ p \in \mathbb{R}^C_{>0} : \sum_{i=1}^C p_i = 1 \right\}. \tag{10}$$

The strict-positivity condition excludes the boundary, where the coordinate expression of the Fisher–Rao metric below becomes singular. For $p \in \Delta_C^\circ$, its tangent space is $T_p\Delta_C^\circ = \{\xi \in \mathbb{R}^C : \sum_{i=1}^C \xi_i = 0\}$. For $\xi, \xi' \in T_p\Delta_C^\circ$, the Fisher–Rao metric is

$$g_p(\xi, \xi') = \sum_{i=1}^C \frac{\xi_i \xi_i'}{p_i}. \tag{11}$$

We write $\langle \xi, \xi' \rangle_p := g_p(\xi, \xi')$ and define the induced norm by $\|\xi\|_p := \sqrt{g_p(\xi, \xi)}$. Define the positive orthant of the radius-two sphere by $\mathbb{S}^{C-1}_{2,+} := \{s \in \mathbb{R}^C_{>0} : \|s\|_2 = 2\}$, where $\|x\|_2 := (\sum_{i=1}^C x_i^2)^{1/2}$ is the Euclidean norm on $\mathbb{R}^C$. The square-root map $\Phi : \Delta_C^\circ \to \mathbb{S}^{C-1}_{2,+}$, $\Phi(p) = 2(\sqrt{p_1}, \ldots, \sqrt{p_C})$, is an isometry. Consequently,

$$d_{\mathrm{FR}}(p, q) = 2 \arccos\left( \sum_{i=1}^C \sqrt{p_i q_i} \right). \tag{12}$$

Let $U_p \subset \Delta_C^\circ$ be a normal neighborhood of $p$ and let $\mathcal{U}_p \subset T_p\Delta_C^\circ$ be its inverse image in normal coordinates. The Riemannian exponential $\mathrm{Exp}_p : \mathcal{U}_p \to U_p$ and logarithm $\mathrm{Log}_p : U_p \to \mathcal{U}_p$ associated with $g$ are inverse maps. For $p, q \in \Delta_C^\circ$, write $\vartheta(p, q) := \arccos(\sum_{i=1}^C \sqrt{p_i q_i})$, where $\arccos$ takes values in $[0, \pi]$ and $\pi$ is the circle constant. With $\odot$ denoting coordinatewise multiplication, the logarithm used in our implementation is

$$\mathrm{Log}_p(q) = \frac{2\vartheta(p, q)}{\sin \vartheta(p, q)} \left( \sqrt{p \odot q} - \cos\big(\vartheta(p, q)\big) p \right), \tag{13}$$

with the continuous value $0 \in T_p\Delta_C^\circ$ when $p = q$. In particular, $\|\mathrm{Log}_p(q)\|_p = d_{\mathrm{FR}}(p, q)$. Two nonzero tangent vectors are compared using

$$\cos_p^{\mathrm{FR}}(\xi, \xi') = \frac{g_p(\xi, \xi')}{\|\xi\|_p \|\xi'\|_p}. \tag{14}$$

For numerical evaluation, we extend Equation 14 by setting the cosine to zero whenever either tangent vector is zero; the regime weights below simultaneously send the corresponding angular mass to zero.

### A.2 NEIGHBOR-ANCHORED CLASS EVIDENCE

Let $V_{\mathrm{tr}} \subseteq V$ be the training-node set and $V_{\mathrm{tr},k} := \{u \in V_{\mathrm{tr}} : y_u = k\}$. Given any current NLDs $\{r_u \in \Delta_C^\circ : u \in V\}$, the routing prototype of class $k$ is the arithmetic simplex mean of its training-node NLDs:

$$\mu_k = \frac{1}{|V_{\mathrm{tr},k}|} \sum_{u \in V_{\mathrm{tr},k}} r_u. \tag{15}$$

Thus $\mu_k \in \Delta_C^\circ$. We assume $V_{\mathrm{tr},k} \neq \varnothing$ for every class, as holds for the evaluated public splits; if a training split lacks a class, the implementation uses the global training-node NLD mean as an explicit fallback. For interacting nodes $v \in \mathcal{N}(u)$, neighbor $v$ is the viewer and $u$ the target. Candidate class $k$ receives the angular evidence

$$\mathfrak{a}_{vuk} = \cos_{r_v}^{\mathrm{FR}}\big(\mathrm{Log}_{r_v} r_u, \mathrm{Log}_{r_v} \mu_k\big), \tag{16}$$

together with radial lengths $d_{vu} = d_{\mathrm{FR}}(r_v, r_u)$ and $d_{vk} = d_{\mathrm{FR}}(r_v, \mu_k)$. This construction asks a concrete question: viewed from the neighborhood distribution of $v$, does the target $u$ move in the same simplex direction as class prototype $k$? It does not require $u$ and $v$ to share a label.

Angles are unreliable when either displacement is nearly zero. For a distance $d \geq 0$ and scale $\tau > 0$, define $s_\tau(d) := d^2/(d^2 + \tau^2)$. Let $\tau_{\mathrm{tgt}}, \tau_{\mathrm{cls}} > 0$ be learned target- and prototype-distance thresholds, respectively. Let $\varepsilon_{\mathrm{num}} > 0$ be a fixed numerical-stability constant. In the implementation, $\tau_e = \mathrm{softplus}(\widetilde{\tau}_e) + \varepsilon_{\mathrm{num}}$ for $e \in \{\mathrm{tgt}, \mathrm{cls}\}$, where $\widetilde{\tau}_e \in \mathbb{R}$ is unconstrained and $\mathrm{softplus}(x) := \log(1 + \exp x)$. We define three smooth regime weights

$$q_{\mathrm{same},vuk} = 1 - s_{\tau_{\mathrm{tgt}}}(d_{vu}), \tag{17}$$

$$q_{\mathrm{proto},vuk} = s_{\tau_{\mathrm{tgt}}}(d_{vu})[1 - s_{\tau_{\mathrm{cls}}}(d_{vk})], \tag{18}$$

$$q_{\mathrm{angle},vuk} = s_{\tau_{\mathrm{tgt}}}(d_{vu})\, s_{\tau_{\mathrm{cls}}}(d_{vk}). \tag{19}$$

They are nonnegative and sum to one. If $d_{vu}$ is small relative to $\tau_{\mathrm{tgt}}$, the same-point expert takes over. If $d_{vk}$ is small relative to $\tau_{\mathrm{cls}}$, the prototype-distance expert receives the non-same-point mass. The angle is used only when both tangent lengths are sufficiently large relative to their thresholds.

## B GEOMETRIC MOTIVATION: COMPATIBILITY AND CONTINUITY

### B.1 CONNECTION TO A CLASS-COMPATIBILITY MATRIX

We use $H \in [0,1]^{C \times C}$ to describe empirical class mixing. Select a node–neighbor incidence uniformly and denote its center and neighbor by $\mathsf{U}$ and $\mathsf{V}$. For $a, b \in [C]$ with $\mathbf{P}(y_{\mathsf{U}} = a) > 0$, define

$$H_{ab} := \mathbf{P}(y_{\mathsf{V}} = b \mid y_{\mathsf{U}} = a). \tag{20}$$

Thus $H_{ab}$ is the probability that a neighbor of a class-$a$ node has class $b$. Every defined row is a categorical distribution: $H_{ab} \geq 0$ and $\sum_{b=1}^{C} H_{ab} = 1$. Large diagonal entries indicate homophilic transitions, whereas large off-diagonal entries indicate heterophilic transitions. The matrix characterizes the graph's label-mixing structure; it is not a learned parameter of FR-NLD.

Let $\varpi = (\varpi_1, \ldots, \varpi_C) \in \Delta_C^\circ$ be a class prior. Assume node classes are drawn independently from $\varpi$, and let $B \in \mathbb{R}_{\geq 0}^{C \times C}$ be a symmetric class-connectivity matrix, where $Y_u$ denotes the random class of node $u$, $\mathbf{P}(Y_u = a) = \varpi_a$, and the conditional edge probability is $\mathbf{P}(\{u, v\} \in E \mid Y_u = a, Y_v = b) = \kappa_B B_{ab}$ for a common $\kappa_B > 0$ chosen so that all probabilities lie in $[0, 1]$. Assume $\sum_{j=1}^{C} \varpi_j B_{aj} > 0$ for every class $a$. Under this generative model, the population compatibility matrix is

$$H_{ab} = \mathbf{P}(Y_v = b \mid v \in \mathcal{N}(u), Y_u = a) = \frac{\varpi_b B_{ab}}{\sum_{j=1}^{C} \varpi_j B_{aj}}. \tag{21}$$

The numerator is the joint mass of choosing class $b$ and forming an $a$–$b$ edge; the denominator sums the same joint mass over every possible neighbor class and therefore normalizes each row. Recall that $H$ is primitive if $H^m$ is strictly positive entrywise for some $m \in \mathbb{N}$. In that case, $H^l$ converges to a rank-one matrix with unique stationary distribution $p_\star \in \Delta_C^\circ$, characterized by $p_\star^\top H = p_\star^\top$. Equation 21 connects the population class-mixing process to the empirical compatibility matrix defined from node–neighbor incidences above.

### B.2 CONTINUITY OF THE REGIME MIXTURE

For $d \geq 0$ and $\tau > 0$, $s_\tau(d) = d^2/(d^2 + \tau^2)$ is continuous, lies in $[0, 1)$, and satisfies $s_\tau(0) = 0$. The three weights in Equation 19 sum to

$$1 - s_{\tau_{\mathrm{tgt}}}(d_{vu}) + s_{\tau_{\mathrm{tgt}}}(d_{vu})\,[1 - s_{\tau_{\mathrm{cls}}}(d_{vk}) + s_{\tau_{\mathrm{cls}}}(d_{vk})] = 1. \tag{22}$$

If $d_{vu} \to 0$, $q_{\mathrm{same},vuk} \to 1$ and $q_{\mathrm{angle},vuk} \to 0$. If $d_{vk} \to 0$ while $d_{vu}$ remains fixed, then $q_{\mathrm{angle},vuk} \to 0$ and $q_{\mathrm{proto},vuk} \to s_{\tau_{\mathrm{tgt}}}(d_{vu})$; the remaining mass is $q_{\mathrm{same},vuk}$. Thus continuous bounded experts combined with these weights remain continuous across the two angular singularities.

## C THE FR-NLD NETWORK

### C.1 ARCHITECTURE OVERVIEW

**Finite-sample NLD posterior.** The following prior construction and degree-reliability gate apply to the dynamic-routing configurations; the Amazon-ratings fusion path uses a smoothed neighbor mean instead. Let $\mathbf{u}_C := (1/C, \ldots, 1/C) \in \Delta_C^\circ$ be the uniform distribution and let $\varepsilon_{\mathrm{sm}} \in (0, 1)$ be a fixed smoothing constant. Define the training-class counts $n_k := |V_{\mathrm{tr},k}|$ and the smoothed train-only class prior

$$\pi_{\mathrm{tr},k} := \frac{n_k + \varepsilon_{\mathrm{sm}}/C}{|V_{\mathrm{tr}}| + \varepsilon_{\mathrm{sm}}}, \qquad k \in \{1, \ldots, C\}. \tag{23}$$

**Optional geometry-side class balancing.** For the optional imbalance correction used on Questions, $p_{u,k}^{(l)} \in [0, 1]$ denotes the model probability that node $u$ belongs to class $k$ at routing depth $l$, and $\pi_{\mathrm{tr},k}$ is the smoothed training-set proportion of class $k$ from Equation 23. Let $\zeta \in [0, 1]$ be the geometry-balancing exponent. Define the geometry-facing belief and prior by

$$\breve{p}_{u,k}^{(l)} := \frac{p_{u,k}^{(l)} \pi_{\mathrm{tr},k}^{-\zeta}}{\sum_{j=1}^{C} p_{u,j}^{(l)} \pi_{\mathrm{tr},j}^{-\zeta}}, \qquad \breve{\pi}_{\mathrm{tr},k} := \frac{\pi_{\mathrm{tr},k}^{1-\zeta}}{\sum_{j=1}^{C} \pi_{\mathrm{tr},j}^{1-\zeta}}. \tag{24}$$

Here the vector $\breve{p}_u^{(l)} := (\breve{p}_{u,1}^{(l)}, \ldots, \breve{p}_{u,C}^{(l)})$ is the class-frequency-corrected node belief, and $\breve{\pi}_{\mathrm{tr}} := (\breve{\pi}_{\mathrm{tr},1}, \ldots, \breve{\pi}_{\mathrm{tr},C})$ is the correspondingly corrected training prior. The exponent $\zeta$ controls the correction strength. Setting $\zeta = 0$ leaves both distributions unchanged, whereas $\zeta = 1$ makes $\breve{\pi}_{\mathrm{tr}}$ uniform and applies full inverse-prior correction to the node belief. Intermediate values provide partial correction. This transformation changes only the distributions supplied to the geometric estimator; it does not change the feature state $h_u^{(l)}$.

**Adaptive train-only prior.** The soft neighborhood count and its smoothed raw NLD are

$$c_u^{(l)} := \sum_{v \in \mathcal{N}(u)} \breve{p}_v^{(l)}, \qquad \widehat{r}_u^{(l)} := (1 - \varepsilon_{\mathrm{sm}}) \begin{cases} c_u^{(l)}/d_u, & d_u > 0, \\ \mathbf{u}_C, & d_u = 0, \end{cases} + \varepsilon_{\mathrm{sm}} \mathbf{u}_C. \tag{25}$$

The vector $c_u^{(l)} \in \mathbb{R}_{\geq 0}^C$ is the soft neighborhood class-count vector used to construct the NLD of node $u$: its $k$th component is $c_{u,k}^{(l)} = \sum_{v \in \mathcal{N}(u)} \breve{p}_{v,k}^{(l)}$. It has total mass $d_u$, and $\widehat{r}_u^{(l)} \in \Delta_C^\circ$. The prototypes used to construct the prior centre are $\widehat{\mu}_k^{(l)} := n_k^{-1} \sum_{u \in V_{\mathrm{tr},k}} \widehat{r}_u^{(l)} \in \Delta_C^\circ$. They are distinguished from the routing prototypes $\mu_k^{(l)}$, which are recomputed from the posterior NLDs below. For any $p \in \Delta_C^\circ$, define its Shannon entropy and normalized confidence by

$$\mathcal{H}(p) := -\sum_{i=1}^{C} p_i \log p_i, \qquad \mathrm{conf}(p) := 1 - \frac{\mathcal{H}(p)}{\log C} \in [0, 1]. \tag{26}$$

The confidence is zero for the uniform distribution and approaches one as $p$ approaches a simplex vertex. Let $\alpha_{\max} \in [0, 1]$ be the maximum permitted class-conditional mixing weight and let $\gamma > 0$

be its confidence-response exponent. Define the class-conditional prior centre by

$$\widetilde{\eta}_u^{(l)} := \sum_{k=1}^{C} \breve{p}_{u,k}^{(l)} \widehat{\mu}_k^{(l)}, \tag{27}$$

$$\alpha_u^{(l)} := \alpha_{\max} \operatorname{conf}(\breve{p}_u^{(l)})^{\gamma}, \tag{28}$$

$$\eta_u^{(l)} := (1 - \alpha_u^{(l)}) \breve{\pi}_{\mathrm{tr}} + \alpha_u^{(l)} \widetilde{\eta}_u^{(l)}, \tag{29}$$

Both $\widetilde{\eta}_u^{(l)}$ and $\eta_u^{(l)}$ lie in $\Delta_C^\circ$. The exponent $\gamma$ controls how sharply the conditional mixing weight responds to confidence: $\gamma > 1$ is more conservative for intermediate-confidence predictions, whereas $0 < \gamma < 1$ is less conservative. Consequently $0 \le \alpha_u^{(l)} \le \alpha_{\max}$ and $\eta_u^{(l)}$ is a convex combination of the geometry-balanced training prior and the predicted class-conditional prototype mixture. The prior strength changes with depth according to

$$\lambda_l = \lambda_\infty + (\lambda_0 - \lambda_\infty) \exp(-l/T_\lambda), \tag{30}$$

where $\lambda_l > 0$ is the prior's effective sample size, or pseudocount, at routing depth $l$. The parameter $\lambda_0 > 0$ is the initial prior strength, $\lambda_\infty > 0$ is its limiting strength as $l \to \infty$, and $T_\lambda > 0$ is the e-folding decay depth: at $l = T_\lambda$, the difference $\lambda_l - \lambda_\infty$ is $e^{-1}$ times its initial value. Our decaying-prior configurations use $\lambda_0 \ge \lambda_\infty$, so early noisy NLDs are shrunk more strongly than mature ones; $\lambda_0 = \lambda_\infty$ recovers a constant prior strength. Instead of normalizing $c_u^{(l)}$ alone, the model forms the posterior centre

$$\bar{r}_u^{(l)} := \frac{c_u^{(l)} + \lambda_l \eta_u^{(l)}}{d_u + \lambda_l}, \qquad r_u^{(l)} := (1 - \varepsilon_{\mathrm{sm}}) \bar{r}_u^{(l)} + \varepsilon_{\mathrm{sm}} \mathbf{u}_C. \tag{31}$$

The positive prior mixture places $\bar{r}_u^{(l)}$ in $\Delta_C^\circ$, and the second smoothing operation applies an additional numerical floor to $r_u^{(l)}$. Finally, the routing prototype used in Equations 16–19 is

$$\mu_k^{(l)} := \frac{1}{n_k} \sum_{u \in V_{\mathrm{tr},k}} r_u^{(l)} \in \Delta_C^\circ. \tag{32}$$

Both $\widehat{\mu}_k^{(l)}$ and $\mu_k^{(l)}$ are treated as constants during back-propagation. More generally, we write $\mathrm{sg}[x]$ for the stop-gradient operator: it equals $x$ in the forward pass and has derivative zero. Equation 31 also gives the empirical-data reliability

$$\chi_u^{(l)} = \frac{d_u}{d_u + \lambda_l}. \tag{33}$$

Thus low-degree nodes are shrunk toward a train-only prior and receive a smaller geometric update. For $d_u = 0$, $\chi_u^{(l)} = 0$ and the geometric message vanishes.

**Class-conditioned routing.** At routing depth $l$, $h_v^{(l)} \in \mathbb{R}^D$ is node $v$'s hidden state and $\overline{h}_v^{(l)} := \mathrm{Norm}_l^{\mathrm{route}}(h_v^{(l)})$ is the normalized state supplied to the router. Here $\mathrm{Norm}_l^{\mathrm{route}} : \mathbb{R}^D \to \mathbb{R}^D$ is the configured LayerNorm, BatchNorm, or identity map at depth $l$. The vector $p_v^{(l)} \in \Delta_C^\circ$ is the current soft class belief and $\breve{p}_v^{(l)}$ is its geometry-facing version from Equation 24. The learned bias-free maps $W_h^{(l)} \in \mathbb{R}^{D \times D}$ and $W_p^{(l)} \in \mathbb{R}^{D \times C}$ project the hidden state and soft belief, respectively, into $D$-dimensional message channels. The symbol $\|$ denotes vector concatenation, so $[W_h^{(l)} \overline{h}_v^{(l)} \| W_p^{(l)} \breve{p}_v^{(l)}] \in \mathbb{R}^{2D}$. The superscript $(l)$ identifies the routing operator used at depth $l$; under parameter sharing, several depths reuse the same $W_h$ and $W_p$.

We now define the gate exactly. The symbols $\mathfrak{a}_{vuk}^{(l)}$, $d_{vu}^{(l)}$, $d_{vk}^{(l)}$, and $q_{\mathrm{same},vuk}^{(l)}$, $q_{\mathrm{proto},vuk}^{(l)}$, and $q_{\mathrm{angle},vuk}^{(l)}$ denote the quantities in Equations 16–19 after replacing $(r, \mu, \tau_{\mathrm{tgt}}, \tau_{\mathrm{cls}})$ by $(r^{(l)}, \mu^{(l)}, \tau_{\mathrm{tgt}}^{(l)}, \tau_{\mathrm{cls}}^{(l)})$. The positive thresholds are trainable parameters of the routing block. For $x, y \in \mathbb{R}^D$, write $\langle x, y\rangle_2 := x^\top y$ for the Euclidean inner product. Define the Euclidean feature cosine between the normalized states of $v$ and $u$ by

$$c_{vu}^{(l)} := \frac{\langle \overline{h}_v^{(l)}, \overline{h}_u^{(l)} \rangle_2}{\max(\|\overline{h}_v^{(l)}\|_2, \varepsilon_{\mathrm{num}}) \max(\|\overline{h}_u^{(l)}\|_2, \varepsilon_{\mathrm{num}})} \tag{34}$$

using the numerical-stability constant introduced above. Define the relative radial mismatch

$$\Delta_{vuk}^{(l)} := \frac{|d_{vu}^{(l)} - d_{vk}^{(l)}|}{d_{vu}^{(l)} + d_{vk}^{(l)} + \varepsilon_{\mathrm{num}}}. \tag{35}$$

The three experts are two-layer multilayer perceptrons (MLPs) $\psi_{\mathrm{same}}^{(l)}, \psi_{\mathrm{proto}}^{(l)} : \mathbb{R}^2 \to \mathbb{R}$ and $\psi_{\mathrm{angle}}^{(l)} : \mathbb{R}^5 \to \mathbb{R}$ shared across all edges and classes within one parameter group. Each has the form $\psi(x) = w_2^\top \mathrm{ReLU}(W_1 x + b_1) + b_2$, with separately learned parameters $W_1 \in \mathbb{R}^{16 \times d_e}$, $b_1, w_2 \in \mathbb{R}^{16}$, and $b_2 \in \mathbb{R}$, where $d_e = 2$ for the first two experts and $d_e = 5$ for the angle expert. Their scalar outputs are

$$e_{\mathrm{same},vuk}^{(l)} := \psi_{\mathrm{same}}^{(l)}\Big(\big(-[d_{vk}^{(l)}]^2, c_{vu}^{(l)}\big)^\top\Big), \tag{36}$$

$$e_{\mathrm{proto},vuk}^{(l)} := \psi_{\mathrm{proto}}^{(l)}\Big(\big(-[d_{vu}^{(l)}]^2, c_{vu}^{(l)}\big)^\top\Big), \tag{37}$$

$$e_{\mathrm{angle},vuk}^{(l)} := \psi_{\mathrm{angle}}^{(l)}\left(\left(\mathfrak{a}_{vuk}^{(l)}, \Delta_{vuk}^{(l)}, \frac{d_{vu}^{(l)}}{\pi}, \frac{d_{vk}^{(l)}}{\pi}, c_{vu}^{(l)}\right)^\top\right). \tag{38}$$

The Fisher–Rao diameter (the supremum of pairwise distances) of the open categorical simplex is $\pi$. With the regime weights from Equation 19, evaluated at depth $l$, set

$$\mathfrak{g}_{vuk}^{(l)} := \tanh\Big(q_{\mathrm{same},vuk}^{(l)} e_{\mathrm{same},vuk}^{(l)} + q_{\mathrm{proto},vuk}^{(l)} e_{\mathrm{proto},vuk}^{(l)} + q_{\mathrm{angle},vuk}^{(l)} e_{\mathrm{angle},vuk}^{(l)}\Big) \in (-1, 1). \tag{39}$$

Here $\tanh(t) := (\exp(t) - \exp(-t))/(\exp(t) + \exp(-t))$ is the hyperbolic tangent. Thus the indices denote viewer $v$, target $u$, and candidate class $k$. The class-specific message $m_{u,k}^{(l)} \in \mathbb{R}^{2D}$ is

$$m_{u,k}^{(l)} = \frac{1}{\max(d_u, 1)} \sum_{v \in \mathcal{N}(u)} \mathfrak{g}_{vuk}^{(l)} \left[W_h^{(l)} \overline{h}_v^{(l)} \| W_p^{(l)} \breve{p}_v^{(l)}\right]. \tag{40}$$

For $d_u = 0$, the empty sum is the zero vector. Concatenating $\{m_{u,k}^{(l)}\}_{k=1}^C$ preserves which candidate class a message supports. Unlike a scalar edge attention score, the same edge may support one class and oppose another.

**Deep routing and dynamic distributions.** For a network with $L \in \mathbb{N}$ routing blocks, let $l \in \{0, \ldots, L\}$ index the hidden state after $l$ completed blocks; block $l$ maps state $l$ to state $l+1$ for $l \in \{0, \ldots, L-1\}$. Stack nodewise quantities as $h^{(l)} \in \mathbb{R}^{N \times D}$, $p^{(l)} = (p_u^{(l)})_{u \in V} \in (\Delta_C^\circ)^N$, and $r^{(l)} = (r_u^{(l)})_{u \in V} \in (\Delta_C^\circ)^N$. Also write $\breve{p}^{(l)} = (\breve{p}_u^{(l)})_{u \in V} \in (\Delta_C^\circ)^N$ and $\mu^{(l)} = (\mu_k^{(l)})_{k=1}^C \in (\Delta_C^\circ)^C$ for the train-only routing prototypes at depth $l$. To define the remaining router inputs, let

$$\nu_k^{(l)} := \mathrm{sg}\left[\frac{1}{n_k} \sum_{u \in V_{\mathrm{tr},k}} \overline{h}_u^{(l)}\right] \in \mathbb{R}^D \tag{41}$$

be the train-only feature prototype. For any score matrix $S = (S_{u,k}) \in \mathbb{R}^{N \times C}$, define its train-only scalar standardization by

$$\mathrm{Std}_{\mathrm{tr}}(S)_{u,k} := \frac{S_{u,k} - \bar{S}_{\mathrm{tr}}}{\max(\sigma_{\mathrm{tr}}(S), \varepsilon_{\mathrm{num}})}, \quad \bar{S}_{\mathrm{tr}} := \frac{1}{C|V_{\mathrm{tr}}|} \sum_{i \in V_{\mathrm{tr}}} \sum_{j=1}^{C} S_{i,j}, \tag{42}$$

where $\sigma_{\mathrm{tr}}(S)^2 := (C|V_{\mathrm{tr}}|)^{-1}\sum_{i\in V_{\mathrm{tr}}}\sum_{j=1}^{C}(S_{i,j}-\bar{S}_{\mathrm{tr}})^2$ is the corresponding population variance. The implementation applies stop-gradient to $\bar{S}_{\mathrm{tr}}$ and $\sigma_{\mathrm{tr}}(S)$ after computing their forward values. The absolute NLD and feature-prototype score matrices are

$$S^{\mathrm{NLD},(l)}_{u,k} := -d_{\mathrm{FR}}(r^{(l)}_u, \mu^{(l)}_k)^2, \qquad S^{\mathrm{feat},(l)}_{u,k} := -\frac{1}{D}\|\overline{h}^{(l)}_u - \nu^{(l)}_k\|_2^2. \tag{43}$$

Let $m^{(l)}_u := [m^{(l)}_{u,1}\| \cdots \| m^{(l)}_{u,C}] \in \mathbb{R}^{2CD}$ and let $\mathcal{M}_l : \mathbb{R}^{2CD+2C} \to \mathbb{R}^D$ be the learned affine channel mixer

$$\mathcal{M}_l(x) := W_{M,l}x + b_{M,l}, \qquad W_{M,l} \in \mathbb{R}^{D\times(2CD+2C)}, \quad b_{M,l} \in \mathbb{R}^D. \tag{44}$$

The geometric routing operator is therefore

$$\mathcal{R}_l(\overline{h}^{(l)}, \breve{p}^{(l)}, r^{(l)}, \mu^{(l)}; G)_u := \mathcal{M}_l\Big(m^{(l)}_u \| \operatorname{Std}_{\mathrm{tr}}(S^{\mathrm{NLD},(l)})_{u,:} \| \operatorname{Std}_{\mathrm{tr}}(S^{\mathrm{feat},(l)})_{u,:}\Big) \in \mathbb{R}^D. \tag{45}$$

Here $S_{u,:}$ denotes row $u$ of a score matrix. This equation fully specifies the quantities that are mapped back to the hidden dimension. The fixed context suppressed from the notation of $\mathcal{R}_l$ consists of the training set $V_{\mathrm{tr}}$, its labels, and the learned parameters of block $l$.

Each routing block uses two pre-normalization modules, $\mathrm{Norm}^{\mathrm{route}}_l$ before geometric routing and $\mathrm{Norm}^{\mathrm{ffn}}_l$ before the nodewise two-layer feed-forward network $\mathrm{FFN}_l : \mathbb{R}^D \to \mathbb{R}^D$. Each normalization is a configured LayerNorm, BatchNorm, or identity map. To make this notation precise, for $x \in \mathbb{R}^D$ define

$$\mathrm{LN}_{\gamma^{\mathrm{norm}},\beta^{\mathrm{norm}}}(x) := \gamma^{\mathrm{norm}} \odot \frac{x - \bar{x}\mathbf{1}_D}{\sqrt{D^{-1}\|x-\bar{x}\mathbf{1}_D\|_2^2 + \varepsilon_{\mathrm{norm}}}} + \beta^{\mathrm{norm}}, \qquad \bar{x} := D^{-1}\sum_{j=1}^{D} x_j, \tag{46}$$

where $\gamma^{\mathrm{norm}}, \beta^{\mathrm{norm}} \in \mathbb{R}^D$ are learned scale and shift vectors, $\mathbf{1}_D \in \mathbb{R}^D$ is the all-ones vector, and $\varepsilon_{\mathrm{norm}} > 0$ is fixed. For a node-state matrix $X \in \mathbb{R}^{N\times D}$, training-mode BatchNorm is

$$\begin{aligned} \mathrm{BN}_{\gamma^{\mathrm{norm}},\beta^{\mathrm{norm}}}(X)_{u,j} &:= \gamma^{\mathrm{norm}}_j \frac{X_{u,j} - \bar{X}_j}{\sqrt{v_j + \varepsilon_{\mathrm{norm}}}} + \beta^{\mathrm{norm}}_j, \\ \bar{X}_j &:= \frac{1}{N}\sum_{i=1}^{N} X_{i,j}, \qquad v_j := \frac{1}{N}\sum_{i=1}^{N}(X_{i,j} - \bar{X}_j)^2. \end{aligned} \tag{47}$$

At evaluation it substitutes the stored running mean and variance for $\bar{X}_j$ and $v_j$; the identity option returns its input unchanged. Each of $\mathrm{Norm}^{\mathrm{route}}_l$ and $\mathrm{Norm}^{\mathrm{ffn}}_l$ has its own learned $\gamma^{\mathrm{norm}}, \beta^{\mathrm{norm}}$. All selected configurations in Table 8 use LayerNorm. Let $\mathrm{Drop}_l : \mathbb{R}^D \to \mathbb{R}^D$ denote dropout during training and the identity map at evaluation. Precisely, for dropout probability $\varrho_{\mathrm{drop}} \in [0,1)$, its training-mode coordinates are $[\mathrm{Drop}_l(x)]_j = \mathfrak{m}_j x_j/(1-\varrho_{\mathrm{drop}})$ with independent $\mathfrak{m}_j \sim \mathrm{Bernoulli}(1-\varrho_{\mathrm{drop}})$; masks are sampled independently across coordinates, nodes, and dropout calls. Let $\beta_l, \delta_l \in \mathbb{R}^D$ be learned LayerScale vectors for the routing and feed-forward residual branches, respectively. They are initialized as small positive multiples of the all-ones vector. With $\odot$ denoting elementwise multiplication, the update is

$$\widetilde{h}^{(l+1)}_u = h^{(l)}_u + \beta_l \odot \mathrm{Drop}_l\Big(\chi^{(l)}_u \mathcal{R}_l(\overline{h}^{(l)}, \breve{p}^{(l)}, r^{(l)}, \mu^{(l)}; G)_u\Big), \tag{48}$$

$$h^{(l+1)}_u = \widetilde{h}^{(l+1)}_u + \delta_l \odot \mathrm{Drop}_l\Big(\mathrm{FFN}_l\Big(\mathrm{Norm}^{\mathrm{ffn}}_l(\widetilde{h}^{(l+1)}_u)\Big)\Big). \tag{49}$$

More explicitly, let $m_{\mathrm{ffn}} \geq 1$ be the FFN expansion multiplier and $D_{\mathrm{ffn}} := \max\{D, \mathrm{round}(m_{\mathrm{ffn}}D)\}$, where round is Python's nearest-integer operation, with exact half-integer ties sent to the nearest even integer. The implemented nodewise map is

$$\mathrm{FFN}_l(x) := A_{2,l}\,\mathrm{Drop}^{\mathrm{in}}_l(\mathrm{GELU}(A_{1,l}x + a_{1,l})) + a_{2,l}, \tag{50}$$

where $A_{1,l} \in \mathbb{R}^{D_{\mathrm{ffn}}\times D}$, $a_{1,l} \in \mathbb{R}^{D_{\mathrm{ffn}}}$, $A_{2,l} \in \mathbb{R}^{D\times D_{\mathrm{ffn}}}$, and $a_{2,l} \in \mathbb{R}^D$. The inner dropout $\mathrm{Drop}^{\mathrm{in}}_l : \mathbb{R}^{D_{\mathrm{ffn}}} \to \mathbb{R}^{D_{\mathrm{ffn}}}$ has the same coordinatewise convention and dropout probability as $\mathrm{Drop}_l$ but an independently sampled mask. The GELU is $\mathrm{GELU}(t) := t\Phi_{\mathrm{N}}(t)$ coordinatewise, where $\Phi_{\mathrm{N}}$ is the standard-normal cumulative distribution function. Let $R \in \{1,\dots,L\}$ be the refresh interval, measured in completed routing blocks, and define $\mathcal{S}_R := \{s \in \{1,\dots,L-1\} : s \equiv 0 \pmod R\}$.

At every refresh depth $s \in \mathcal{S}_R$, a shared bias-free classifier $W_{\mathrm{dyn}} \in \mathbb{R}^{C\times D}$ forms candidate logits and beliefs

$$z^{(s)}_{\mathrm{aux},u} := \frac{W_{\mathrm{dyn}} h^{(s)}_u}{T_{\mathrm{dyn}}}, \qquad \widehat{p}^{(s)}_u := \mathrm{softmax}(z^{(s)}_{\mathrm{aux},u}), \tag{51}$$

where $T_{\mathrm{dyn}} > 0$ is a temperature. Let $\widehat{p}^{(s)}_{u,(1)} \geq \widehat{p}^{(s)}_{u,(2)}$ be the two largest coordinates of $\widehat{p}^{(s)}_u$ and define

$$M^{(s)}_u := \widehat{p}^{(s)}_{u,(1)} - \widehat{p}^{(s)}_{u,(2)}, \qquad \kappa^{(s)}_u := \kappa_{\max}\,\mathrm{sigmoid}\left(\frac{M^{(s)}_u - M_0}{T_{\mathrm{gate}}}\right), \tag{52}$$

where $\kappa_{\max} \in [0,1]$ is the largest refresh weight, $M_0 \in \mathbb{R}$ is the confidence-margin threshold, and $T_{\mathrm{gate}} > 0$ is the gate temperature; here $\mathrm{sigmoid}(t) := (1+\exp(-t))^{-1}$. The refreshed belief is

$$p^{(s)}_u := (1-\kappa^{(s)}_u)p^{(s-R)}_u + \kappa^{(s)}_u \widehat{p}^{(s)}_u. \tag{53}$$

The NLDs $r^{(s)}$ and routing prototypes $\mu^{(s)}$ are then recomputed from $p^{(s)}$ using Equations 24– 32. Between refresh depths, $p$, $r$, and $\mu$ retain their most recently computed values. Thus $R = 4$, for example, refreshes after blocks 4, 8, 12, and so on, but not after the final block. All prototypes use training labels only; validation and test labels never enter their construction. For parameter efficiency, deep configurations may reuse a residual routing block across consecutive depths; Appendix C.3 specifies the exact sharing rule.

The implemented depth readout uses one learned scalar logit $\upsilon^{\mathrm{read}}_l \in \mathbb{R}$ for every hidden depth $l \in \{0,\dots,L\}$. Define

$$\omega_l := \frac{\exp(\upsilon^{\mathrm{read}}_l)}{\sum_{j=0}^{L}\exp(\upsilon^{\mathrm{read}}_j)}, \qquad h^{\mathrm{read}}_u := J_\psi\left(\sum_{l=0}^{L}\omega_l h^{(l)}_u\right), \qquad z_u := W_{\mathrm{out}} h^{\mathrm{read}}_u + b_{\mathrm{out}}, \tag{54}$$

where $J_\psi(x) := \mathrm{Drop}^{\mathrm{read}}(\mathrm{GELU}(A_{\mathrm{read}}x + a_{\mathrm{read}}))$, $A_{\mathrm{read}} \in \mathbb{R}^{D\times D}$, $a_{\mathrm{read}} \in \mathbb{R}^D$, and $\psi$ denotes the affine parameters together with the readout-dropout setting. The readout dropout obeys the same training/evaluation convention as $\mathrm{Drop}_l$. Furthermore, $(W_{\mathrm{out}}, b_{\mathrm{out}}) \in \mathbb{R}^{C\times D} \times \mathbb{R}^C$ is the final classifier. The weights $\omega_l > 0$ sum to one and are shared by all nodes; they are not node-specific, and $z_u \in \mathbb{R}^C$ is the final logit vector of node $u$. We write $z := (z_u)_{u\in V} \in \mathbb{R}^{N\times C}$ for the stacked final logits.

**Training objective.** For logits $\ell = (\ell_u)_{u\in V} \in \mathbb{R}^{N\times C}$ and the label vector $y := (y_u)_{u\in V}$, let $\mathrm{softmax}(\ell_u) \in \Delta^\circ_C$ denote the softmax distribution, $\mathrm{softmax}(\ell_u)_k := \exp(\ell_{u,k})/\sum_{j=1}^{C}\exp(\ell_{u,j})$. For positive class weights $w_1,\dots,w_C$, the weighted cross-entropy loss is

$$\mathcal{L}_{\mathrm{CE}}(\ell, y) := -\frac{1}{\sum_{u\in V_{\mathrm{tr}}} w_{y_u}} \sum_{u\in V_{\mathrm{tr}}} w_{y_u} \log \mathrm{softmax}(\ell_u)_{y_u}. \tag{55}$$

The ordinary unweighted loss is recovered with $w_k = 1$ for every class. For two collections of nodewise beliefs $p, q \in (\Delta^\circ_C)^N$, we use the node-averaged Kullback–Leibler divergence

$$D_{\mathrm{KL}}(p\|q) := \frac{1}{N}\sum_{u\in V}\sum_{k=1}^{C} p_{u,k}\log\frac{p_{u,k}}{q_{u,k}}. \tag{56}$$

Write $z^{(s)}_{\mathrm{aux}} := (z^{(s)}_{\mathrm{aux},u})_{u\in V} \in \mathbb{R}^{N\times C}$. Using the stop-gradient operator defined after Equation 32, the refresh regularizer for $\mathcal{S}_R \neq \varnothing$ is

$$\begin{aligned}\mathcal{L}_{\mathrm{refresh}} := {} & \frac{\lambda_{\mathrm{aux}}}{|\mathcal{S}_R|}\sum_{s\in\mathcal{S}_R}\mathcal{L}_{\mathrm{CE}}(z^{(s)}_{\mathrm{aux}}, y) \\ & + \frac{\lambda_{\mathrm{cons}}}{|\mathcal{S}_R|}\sum_{s\in\mathcal{S}_R} D_{\mathrm{KL}}\left(\mathrm{sg}[p^{(s-R)}]\|p^{(s)}\right).\end{aligned} \tag{57}$$

Here $\lambda_{\mathrm{aux}}, \lambda_{\mathrm{cons}} \geq 0$ weight refresh supervision and consecutive-belief consistency. If $\mathcal{S}_R$ is empty, we set $\mathcal{L}_{\mathrm{refresh}} := 0$. After an initial-encoder warm-up, the feature encoder and initial classifier have

been trained by minimizing $\mathcal{L}_{\mathrm{CE}}(z^{(0)}, y)$, where $z^{(0)} := (z_u^{(0)})_{u \in V} \in \mathbb{R}^{N \times C}$. They are then frozen while the geometry phase minimizes

$$\mathcal{L}_{\mathrm{geom}} := \mathcal{L}_{\mathrm{CE}}(z, y) + \mathcal{L}_{\mathrm{refresh}}. \tag{58}$$

During optional joint fine-tuning, the encoder and initial classifier are unfrozen and the objective is

$$\mathcal{L}_{\mathrm{joint}} := \mathcal{L}_{\mathrm{geom}} + \lambda_{\mathrm{base}} \mathcal{L}_{\mathrm{CE}}(z^{(0)}, y), \tag{59}$$

where $\lambda_{\mathrm{base}} \geq 0$ preserves supervision of the initial logits. For highly imbalanced binary graphs we optionally set $w_k \propto n_k^{-\tau}$, where $\tau \in [0, 1]$ controls the strength of inverse-frequency weighting and the counts $n_k$ were defined above Equation 23. Because Equation 55 normalizes by the sum of sample weights, the unspecified positive proportionality constant cancels. The loss exponent $\tau$ and geometry-balancing exponent $\zeta$ are distinct hyperparameters. Neither should be confused with the learned geometric distance thresholds $\tau_{\mathrm{tgt}}^{(l)}$ and $\tau_{\mathrm{cls}}^{(l)}$.

### C.2 Validation-selected configuration switches

**Optional deployment variants.** Let $z_u^{(m)}$ denote the logits of independently trained FR-NLD member $m$, and set $p_u^{(m)} = \mathrm{softmax}(z_u^{(m)}/T_+)$. The Plus and Pro predictions are

$$p_u^{+} = \sum_{m=1}^{3} \alpha_m p_u^{(m)}, \qquad p_u^{\mathrm{Pro}} = (1 - \rho_{\mathrm{pro}}) p_u^{+} + \rho_{\mathrm{pro}} p_u^{\mathrm{GCN}}, \tag{60}$$

where $\boldsymbol{\alpha} \in \Delta_3$ gives the Plus member weights: equal except on Questions, where validation selects them. Validation also selects $\rho_{\mathrm{pro}} \in [0, 1]$. The probability temperature is $T_+ = 1.5$ for Questions and $T_+ = 1$ otherwise. Plus retains only three FR-NLD members at inference; Pro retains an auxiliary GCN only when $\rho_{\mathrm{pro}} > 0$. GCNs used solely to construct distillation targets are discarded at inference.

Table 4 records the validation-selected deployment weights. Here $\boldsymbol{\beta}$ mixes the three auxiliary GCNs before the outer Pro mixture. Questions selects three active Plus members from four candidates, with full candidate weights $(0, 0.25, 0.50, 0.25)$ and probability temperature $1.5$.

Table 4: Validation-selected FR-NLD-Pro readout settings. A zero outer weight recovers FR-NLD-Plus exactly.

| Dataset | Auxiliary weights $\boldsymbol{\beta}$ | $\rho_{\mathrm{pro}}$ |
|---|---|---|
| Roman-empire | $(0.50, 0.25, 0.25)$ | 0.40 |
| Amazon-ratings | $(0.50, 0.25, 0.25)$ | 0.50 |
| Minesweeper | off | 0.00 |
| Tolokers | $(0.50, 0.25, 0.25)$ | 0.40 |
| Questions | $(0.25, 0.25, 0.50)$ | 0.25 |

Viewer-anchored Fisher–Rao evidence is shared across datasets; its injection path, input channels, and optional training-only distillation term are selected separately. The input and NLD estimator expose three binary switches $s_{\mathrm{sep}}, s_{\mathrm{graph}}, s_{\mathrm{att}} \in \{0, 1\}$, together with the integer prototype count $M$ and bounded readout weight $\omega_{\mathrm{out}}$.

The separated-encoder switch preserves distinct transformations of the center feature and mean neighbor feature when active. With

$$\begin{aligned} \bar{x}_u &:= \frac{1}{\max(d_u, 1)} \sum_{v \in \mathcal{N}(u)} x_v, \\ h_u^{(0)} &:= (1 - s_{\mathrm{sep}}) E_{\mathrm{feat}}(x_u) + s_{\mathrm{sep}} [E_{\mathrm{ego}}(x_u) + \sigma(\gamma) \odot E_{\mathrm{nbr}}(\bar{x}_u)]. \end{aligned} \tag{61}$$

$E_{\mathrm{feat}}$, $E_{\mathrm{ego}}$, and $E_{\mathrm{nbr}}$ are two-layer MLPs, $\gamma \in \mathbb{R}^D$ is a learned channelwise gate logit, $\sigma(t) := (1 + \exp(-t))^{-1}$ is the logistic sigmoid, and $\odot$ denotes elementwise multiplication. Thus $s_{\mathrm{sep}} = 0$ is the feature-only encoder and $s_{\mathrm{sep}} = 1$ activates separated center/neighbor encoding. The gate is initialized to $0.1$ after the sigmoid.

The graph-channel switch can augment the feature encoder without changing any Fisher–Rao routing operation. Let $E_{\mathrm{feat}}$ be the feature-only MLP and $E_{\mathrm{gcn}}$ a residual GCN encoder. We use

$$\begin{aligned} h_u^{(0)} &= (1 - s_{\mathrm{graph}}) E_{\mathrm{feat}}(x_u) \\ &\quad + s_{\mathrm{graph}}[(1-\rho) \odot E_{\mathrm{feat}}(x_u) + \rho \odot E_{\mathrm{gcn}}(X, A)_u], \qquad \rho := \sigma(a_{\mathrm{graph}}). \end{aligned} \tag{62}$$

where $s_{\mathrm{graph}} \in \{0, 1\}$ is the configuration switch and $a_{\mathrm{graph}} \in \mathbb{R}^D$ is learned. Thus $s_{\mathrm{graph}} = 0$ recovers the feature-only encoder exactly, whereas $s_{\mathrm{graph}} = 1$ supplies a gated graph channel to the same router.

The integer $M \geq 1$ controls how many prototypes represent each class. For $M > 1$, one class is represented by a small bank $\{\mu_{k,j}^{(l)}\}_{j=1}^{M}$ rather than a single NLD prototype. The bank is computed from training nodes only by deterministic spherical clustering after the square-root simplex embedding. For each viewer and candidate class, modes are softly weighted according to their Fisher–Rao distances, with temperature $T_{\mathrm{mix}} > 0$:

$$\rho_{v,k,j}^{(l)} := \frac{\exp\Big(-d_{\mathrm{FR}}(r_v^{(l)}, \mu_{k,j}^{(l)})^2 / T_{\mathrm{mix}}\Big)}{\sum_{t=1}^{M} \exp\Big(-d_{\mathrm{FR}}(r_v^{(l)}, \mu_{k,t}^{(l)})^2 / T_{\mathrm{mix}}\Big)}. \tag{63}$$

The class-conditioned edge gate is the corresponding weighted mixture of the $M$ mode-specific gates. Setting $M = 1$ exactly recovers the original prototype.

The attention switch controls whether the NLD uses uniform or feature-compatible neighbor weights. For $d_u > 0$, define

$$e_{vu} := \frac{\langle \mathrm{norm}(W_Q x_u), \mathrm{norm}(W_K x_v) \rangle}{T_{\mathrm{att}}}, \qquad \alpha_{vu} := \frac{\exp(e_{vu})}{\sum_{w \in \mathcal{N}(u)} \exp(e_{wu})}, \tag{64}$$

where $W_Q$ and $W_K$ are learned projections, $\mathrm{norm}(a) := a / \max(\|a\|_2, \varepsilon_{\mathrm{num}})$, and $T_{\mathrm{att}} > 0$ is a temperature. A learned scalar $\kappa_{\mathrm{att}} = \sigma(a_{\mathrm{att}}) \in (0, 1)$ interpolates between uniform and learned weights:

$$\begin{aligned} w_{vu} &:= \frac{1 - s_{\mathrm{att}} \kappa_{\mathrm{att}}}{d_u} + s_{\mathrm{att}} \kappa_{\mathrm{att}} \alpha_{vu}, \\ d_{u,\mathrm{eff}} &:= \left( \sum_{v \in \mathcal{N}(u)} w_{vu}^2 \right)^{-1}, \qquad c_{u,\mathrm{att}}^{(l)} := d_{u,\mathrm{eff}} \sum_{v \in \mathcal{N}(u)} w_{vu} \breve{p}_v^{(l)}. \end{aligned} \tag{65}$$

The finite-sample posterior then replaces $(c_u^{(l)}, d_u)$ by $(c_{u,\mathrm{att}}^{(l)}, d_{u,\mathrm{eff}})$. Setting $s_{\mathrm{att}} = 0$ recovers the ordinary uniform NLD estimator exactly; $s_{\mathrm{att}} = 1$ activates the learned interpolation. The attention-weighted ablation uses a 16-dimensional query/key space, $T_{\mathrm{att}} = 1$, and initializes $\kappa_{\mathrm{att}} = 0.1$.

An optional training-only distillation coefficient $\lambda_{\mathrm{KD}} \geq 0$ augments Equation 58 by

$$\mathcal{L}_{\mathrm{geom}}^{+} := \mathcal{L}_{\mathrm{geom}} + \lambda_{\mathrm{KD}} T_{\mathrm{KD}}^2 \frac{1}{|V|} \sum_{u \in V} \mathrm{KL}\Big( q_u^{(T_{\mathrm{KD}})} \,\Big\|\, p_u^{(T_{\mathrm{KD}})} \Big), \tag{66}$$

where $p_u^{(T)} := \mathrm{softmax}(z_u / T)$ and, for a frozen probability target $q_u$, $q_u^{(T)} := \mathrm{softmax}(\log q_u / T)$. Setting $\lambda_{\mathrm{KD}} = 0$ recovers ordinary training exactly. For the Amazon-ratings FR-NLD result in Table 1, the target is the equal three-seed teacher ensemble protected by a $0.5$ probability mixture with the undistilled student, and validation selection chooses $\lambda_{\mathrm{KD}} = 0.3$ and $T_{\mathrm{KD}} = 1$. The teacher is used only to construct training targets and is absent from inference; consequently the distilled and undistilled students have the same inference parameter count.

For Questions, let $t_{u,j}$ denote the logit vector of frozen GCN teacher $j \in \{0, 1, 2\}$, and let $z_u^{\mathrm{base}}$ denote the undistilled FR-NLD logit vector. The validation-selected soft target is

$$\bar{q}_u := \sum_{j=0}^{2} \alpha_j^{\mathrm{teach}} \, \mathrm{softmax}(t_{u,j}), \qquad q_u^{\mathrm{Q}} := (1 - \rho) \, \mathrm{softmax}(z_u^{\mathrm{base}}) + \rho \bar{q}_u, \tag{67}$$

where $\alpha_j^{\text{teach}} \geq 0$, $\sum_j \alpha_j^{\text{teach}} = 1$, and $\rho \in [0, 1]$. We screen a quarter-step simplex for $\boldsymbol{\alpha}^{\text{teach}}$ and $\rho \in \{0.25, 0.40, 0.50, 0.60, 0.75\}$ using mean validation ROC-AUC over the ten public splits. Validation selects $\boldsymbol{\alpha}^{\text{teach}} = (0.25, 0.25, 0.50)$ and $\rho = 0.40$. Substituting $q_u^{\text{Q}}$ into Equation 66, validation selects $\lambda_{\text{KD}} = 0.4$ and $T_{\text{KD}} = 2$. The resulting single student reaches $78.405 \pm 1.467$ test ROC-AUC with $1.18$M inference parameters, compared with $77.442 \pm 2.744$ without distillation. It wins nine of ten paired splits and improves split 9 by $4.892$ points. The GCN teachers are used only to precompute training targets and are discarded at inference.

The Questions FR-NLD-Plus readout selects three active students from four validation candidates. Three candidates use $\lambda_{\text{KD}} = 0.3$, $T_{\text{KD}} = 1$, and a protected target mixing the equal-weight three-GCN ensemble with the undistilled FR-NLD prediction at weight $0.5$; the fourth uses the best-target setting $\lambda_{\text{KD}} = 0.4$, $T_{\text{KD}} = 2$. Validation selects probability temperature $1.5$ and weights $(0, 0.25, 0.50, 0.25)$, leaving three active students. This yields $79.058 \pm 1.334$ test ROC-AUC with $3.54$M inference parameters. The equal-weight three-student reference reaches $79.014 \pm 1.447$. GCN teachers used to construct soft targets are discarded before inference.

Finally, the bounded hyperparameter $\omega_{\text{out}}$ interpolates in probability space between the base belief and routed belief:

$$p_u^{\text{out}} = (1 - \omega_{\text{out}}) p_u^{(0)} + \omega_{\text{out}} p_u^{\text{route}}, \qquad \omega_{\text{out}} \in [0, 1]. \tag{68}$$

In the matched Amazon/Tolokers transfer test, both $s_{\text{graph}}$ and $\omega_{\text{out}}$ are selected on validation splits 0–2 before the ten test splits are aggregated. Tolokers selects $s_{\text{graph}} = 1$ and $\omega_{\text{out}} = 0.5$, whereas Amazon-ratings selects the graph channel off. The remaining main-table configurations use $s_{\text{graph}} = 0$ and $\omega_{\text{out}} = 1$.

Table 5 records the single-model configurations reported in Table 1. Candidate values were screened on public validation splits, and the selected settings were evaluated on all ten public test splits. The input path and optional training objective vary by dataset; all configurations retain the viewer-anchored geometric router.

Table 5: **Selected single-model configurations in Table 1.** The input path names the initial predictor: A28 uses feature–geometry fusion with even two-hop graph-filtered features, H1 uses an MLP–GCN hybrid, and the dual stream separates center and neighbor features. $M$ is the number of NLD prototypes per class; $\lambda_{\text{KD}}$ and $T_{\text{KD}}$ specify the training-only distillation term in Equation 66; and $\omega_{\text{out}}$ is the protected readout weight in Equation 68. All rows use uniform NLD aggregation. Scores are official ten-split test means and sample standard deviations in percent.

| Dataset | Input path | $M$ | $\lambda_{\text{KD}}$ | $T_{\text{KD}}$ | $\omega_{\text{out}}$ | Test score |
|---|---|---|---|---|---|---|
| Roman-empire | feature MLP | 1 | 0 | – | 1.0 | $91.677 \pm 0.416$ |
| Amazon-ratings | A28 + even-2-hop | 1 | 0.3 | 1 | 1.0 | $53.975 \pm 0.724$ |
| Minesweeper | feature MLP | 1 | 0 | – | 1.0 | $99.116 \pm 0.170$ |
| Tolokers | H1 hybrid | 1 | 0.3 | 1 | 0.5 | $85.974 \pm 0.739$ |
| Questions | dual stream | 2 | 0.4 | 2 | 1.0 | $78.405 \pm 1.467$ |

For Tolokers, the fixed-hybrid input and protected readout are combined with training-only distillation at $\lambda_{\text{KD}} = 0.3$ and $T_{\text{KD}} = 1$. FR-NLD-Plus and FR-NLD-Pro add the inference readouts specified in Equation 60.

On Tolokers, the matched raw graph-channel model improves the control by $0.588$ points (95% paired interval $[0.280, 0.896]$) and wins all ten splits; the validation-selected protected readout improves it by $0.701$ points ($[0.396, 1.006]$), also on all ten splits. A compact transfer of the same channel to Amazon-ratings decreases the mean by $0.995$ points, and validation selection keeps that channel off there.

### C.3 Parameter-shared depth

Deep geometric iteration need not allocate a distinct parameter set at every executed step. Let $L \in \mathbb{N}$ be the number of executed routing updates and $P \in \{1, \ldots, L\}$ the number of distinct trainable

parameter groups. We write $\phi_j$ for the complete parameter tuple of group $j$, including the geometric router, normalization modules, feed-forward network, and LayerScale vectors. Let $\mathcal{B}_{\phi_j}$ denote the complete residual routing block and $\mathcal{R}_{\phi_j}$ its geometric routing component. The assignment is

$$\iota(l) = \min\left(P-1, \left\lfloor \frac{lP}{L} \right\rfloor\right), \qquad \mathcal{B}_l = \mathcal{B}_{\phi_{\iota(l)}}, \quad \text{and hence} \quad \mathcal{R}_l = \mathcal{R}_{\phi_{\iota(l)}}. \tag{69}$$

Here $\iota(l)$ is the parameter-group index assigned to executed depth $l$. All depths assigned to one group reuse the entire residual block, but the model still executes $L$ geometric updates; sharing changes the parameter count, not the propagation depth or NLD-refresh schedule. For Roman-empire, $L = 112$ and $P = 28$, so four consecutive updates share one parameter tuple. At the extremes, $P = L$ gives no sharing and $P = 1$ shares one tuple across all updates.

# D Full experimental specification and analysis

## D.1 Protocol

We evaluate on the five modern heterophily benchmarks of Platonov et al. (2023). Table 6 summarizes their scale. We use the ten released splits, screen hyperparameter configurations on validation splits, and report the mean and sample standard deviation over test splits. Accuracy is used for the multiclass datasets and the area under the receiver operating characteristic curve (ROC-AUC) for the binary datasets. All reported prototypes and priors are constructed without validation or test labels.

Table 6: **Datasets and evaluation metrics.** Edge counts follow the undirected benchmark representation. "Feat." is the input-feature dimension $F$.

| Dataset | Nodes | Edges | Feat. | Classes | Metric |
|---|---|---|---|---|---|
| Roman-empire | 22,662 | 32,927 | 300 | 18 | Accuracy |
| Amazon-ratings | 24,492 | 93,050 | 300 | 5 | Accuracy |
| Minesweeper | 10,000 | 39,402 | 7 | 2 | ROC-AUC |
| Tolokers | 11,758 | 519,000 | 10 | 2 | ROC-AUC |
| Questions | 48,921 | 153,540 | 301 | 2 | ROC-AUC |

Within the dynamic-routing configurations, depth, width, dropout, prior strength, parameter sharing, and optimization budgets are validation selected. Roman-empire executes 112 routing steps with 28 parameter groups; the remaining dynamic-routing datasets use shallower configurations. Amazon-ratings uses feature–geometry fusion without periodic refresh. Questions uses both the inverse-frequency loss weights in Equation 55 and the geometry-facing class-prior correction in Equation 24. These are imbalance hyperparameters; neither introduces a new routing block or message rule.

The same-hardware single-model resource comparison appears in Table 2 in the main paper.

## D.2 Main results

The complete baseline comparison is reported in Table 1 in the main paper.

Among the external ranked comparators in Table 1, the single FR-NLD model has the highest mean on Minesweeper and Tolokers, reaching $99.116$ and $85.974$ ROC-AUC, respectively. It exceeds the published tuned GCN by $1.256$ points on Minesweeper and $0.407$ points on Roman-empire. It trails Polynormer-r by $0.873$ points on Roman-empire and exceeds its Tolokers mean by $0.064$ points. The Plus and Pro readouts raise performance further as shown in Table 1.

The feature-path and distillation gains on Amazon-ratings and Questions are discussed in Appendix E.4.

## D.3 What produces the gain?

**Finite-degree and prior components.** Table 7 follows a matched Roman-empire path. Degree-aware shrinkage improves the standalone dynamic model by $0.309$ points. A stronger layerwise prior

Table 7: **Matched Roman-empire component results.** The first comparison uses a common experiment; the next four rows use the constant-prior matched control $\lambda_l \equiv 24$. The final row is the selected deep shared configuration.

| Model component | Test accuracy | Gain |
|---|---|---|
| Dynamic standalone router | $90.321 \pm 0.309$ | — |
| + degree-aware prior | $90.630 \pm 0.223$ | $+0.309$ |
| Degree-aware control, $\lambda = 24$ | $91.168 \pm 0.550$ | — |
| + class-conditional prior only | $91.098 \pm 0.394$ | $-0.071$ |
| + layerwise strength only | $91.218 \pm 0.561$ | $+0.049$ |
| + both (FR-NLD) | $91.329 \pm 0.413$ | $+0.161$ |
| + deep shared routing | $91.677 \pm 0.416$ | $+0.348$ |

improves the result further, while class-conditional and layerwise priors together reach $91.329$. A longer validation-selected schedule and depth refinement followed by span-four parameter sharing yields the final $91.677$ result in Table 1.

The combination gain over the $\lambda = 24$ control is paired across splits: $+0.161$ points with a 95% confidence interval of $[0.015, 0.306]$ and a win/tie/loss count of $8/0/2$. The class-conditional term is not independently beneficial in this search, but becomes useful when coupled to depth-dependent prior strength. This is evidence for an interaction, not for additive effects. The last row additionally changes depth and optimization budget, so its gain is a configuration result rather than an isolated causal effect.

**Dynamic refresh.** In an earlier matched ten-split experiment, dynamic NLD refresh improved the static router by $0.607$ points (95% CI $[0.388, 0.827]$; $10/0/0$ wins/ties/losses).

**Parameter sharing and resource use.** Roman-empire uses a sharing span of four; the selected configurations for the other datasets are listed in Appendix E.

Additional interpretation and limitations appear in Appendix E.4.

# E Additional experimental analyses

## E.1 Model-selection and reporting protocol

Configurations and nonfixed readout weights are selected using validation data; Plus member weights are fixed equal except on Questions. We report the settings on all ten public test splits. Complete candidate lists and validation rankings will accompany the full code release upon acceptance.

## E.2 Selected configurations

Table 8: **Principal dataset-varying FR-NLD configurations.** Hidden is the hidden dimension $D$. For dynamic-routing rows, Steps is the number $L$ of executed routing updates and Groups is the number $P$ of distinct routing parameter blocks; for Amazon-ratings, Steps counts feature-attention layers and Groups is not applicable. Dropout is $\varrho_{\mathrm{drop}}$. The prior columns are the initial strength $\lambda_0$, limiting strength $\lambda_\infty$, and decay depth $T_\lambda$ for dynamic routing and are not applicable to Amazon-ratings. The columns $\zeta$ and $\tau$ are the geometry- and loss-balancing exponents from Equations 24 and 55. All other shared and optimization settings will accompany the full code release.

| Dataset | Hidden | Steps | Groups | Dropout | $\lambda_0$ | $\lambda_\infty$ | $T_\lambda$ | $\zeta$ | $\tau$ |
|---|---|---|---|---|---|---|---|---|---|
| Roman-empire | 96 | 112 | 28 | 0.5 | 48 | 24 | 32 | 0 | 0 |
| Amazon-ratings | 384 | 8 | – | 0.7 | – | – | – | 0 | 0 |
| Minesweeper | 96 | 32 | 32 | 0.2 | 36 | 18 | 32 | 0 | 0 |
| Tolokers | 32 | 4 | 4 | 0.3 | 48 | 24 | 32 | 0 | 0 |
| Questions | 128 | 8 | 8 | 0.5 | 3.75 | 1.5 | 32 | 0.25 | 0.75 |

### E.3 EXPANDED INTERVENTION AND RESOURCE ANALYSIS

Table 9: **Cross-dataset depth and routing-residual ablation.** All entries are paired ten-split test-score changes in percentage points. $\Delta_{\mathrm{depth}}$ compares the deep and shallow direction-enabled residual models; $\Delta_{\mathrm{res}}$ compares residual on and off at the deep depth; and $\Delta_{\mathrm{int}}$ is their difference-in-differences.

| Dataset | Shallow $\rightarrow$ deep | $\Delta_{\mathrm{depth}}$ | $\Delta_{\mathrm{res}}$ | $\Delta_{\mathrm{int}}$ |
|---|---|---|---|---|
| Roman-empire | $16 \rightarrow 112$ | $+0.980$ | $+8.212$ | $+4.665$ |
| Amazon-ratings | $2 \rightarrow 8$ | $-0.077$ | $-0.861$ | $-0.042$ |
| Minesweeper | $8 \rightarrow 32$ | $+3.509$ | $+8.361$ | $+4.175$ |
| Tolokers | $1 \rightarrow 4$ | $+2.620$ | $+1.419$ | $+0.812$ |
| Questions | $2 \rightarrow 8$ | $+0.561$ | $+1.058$ | $+0.941$ |

On Roman-empire, Minesweeper, Tolokers, and Questions, all three contrasts have positive paired 95% intervals: deeper residual routing improves the test score, the residual is beneficial at the deep configuration, and its benefit increases with depth. For Amazon-ratings, the depth change is $-0.077$ points (interval $[-0.406, 0.252]$), the deep residual effect is $-0.861$ points (interval $[-1.157, -0.564]$), and the interaction is $-0.042$ points (interval $[-0.439, 0.355]$).

**Small-model geometric ablations.** Table 3 uses compact eight-step models on all five datasets. Amazon-ratings retains its feature–geometry fusion path, while the other datasets use dynamic routing. Configuration choices and stopping points are selected using validation data; the full and zero-angle models share the same initial predictor but train their geometric parameters separately. For checkpoint interventions, the trained full model is evaluated unchanged, with angle set to zero, and with its geometric viewers shuffled three times using fixed seeds; the shuffle result is averaged within each split. Reported deltas are normal minus altered test score in percentage points. The first column retains the nongeometric feature/FFN scaffold and retrains it from the same initial predictor; Table 3 reports all ten official splits.

**Earlier Roman checkpoint interventions.** The earlier intervention study uses independently trained 32-layer frozen- and dynamic-NLD predecessors. Viewer shuffling preserves the marginal NLD set but breaks the neighbor–tangent-base assignment; angle removal preserves radial distances. Zeroing distances disables the geometric route and is out of distribution, so its magnitude measures sensitivity rather than the standalone value of distance. Together these interventions separately implicate viewer identity, radial evidence, and tangent direction in those fitted checkpoints.

The selected Minesweeper model uses $2.20$M parameters versus Polynormer's $10.54$M in our official-implementation reproduction, combining a $1.835$-point ROC-AUC gain with approximately $79\%$ fewer parameters.

### E.4 ADDITIONAL DISCUSSION

**Operating regime.** Amazon-ratings benefits from richer feature interactions, and the class-imbalanced Questions graph benefits from training-only distillation. Their single-model scores reach $53.975$ accuracy and $78.405$ ROC-AUC, respectively. Three active FR-NLD students raise them to $54.155$ and $79.058$, while the auxiliary-readout variant reaches $55.358$ and $79.379$. The GCN teachers used for distillation are discarded at inference; only Pro retains an auxiliary GCN readout.

**Limitations.** FR-NLD depends on stable train-only class prototypes, so sparse classes, domain shift, or miscalibrated beliefs can corrupt its prior; recomputing NLDs and prototypes adds cost; and the five-dataset interventions use compact models rather than the full Table 1 configurations.